Computational and Applied Mathematics manuscript No.
(will be inserted by the editor)

An inexact restoration derivative-free filter method for nonlinear programming

N. Echebest · M.L. Schuverdt · R.P. Vignau

Received: date / Accepted: date

Abstract An inexact restoration derivative-free filter method for nonlinear programming is introduced in this paper. Each iteration is composed of a restoration phase, which reduces a measure of infeasibility, and an optimization phase, which reduces the objective function. The restoration phase is solved using a derivative-free method for solving underdetermined nonlinear systems with bound constraints, developed previously by the authors. An alternative for solving the optimization phase is considered. Theoretical convergence results and some preliminary numerical experiments are presented.

Keywords Derivative-free · Nonlinear programming · Filter methods · Inexact restoration methods

Mathematics Subject Classification (2010) 65K05 · 90C30 · 90C56

1 Introduction

In this paper we shall be concerned with the nonlinear programming problem

$$\begin{array}{c}\text{minimize } f(x)\\ \text{subject to } c(x)=0\end{array} \tag{1}$$

where the functions $f:\mathbb{R}^n\to\mathbb{R}$, $c:\mathbb{R}^n\to\mathbb{R}^m$ are continuously differentiable but their derivatives are not available. We denote by $J_c(.)$ the Jacobian matrix of c and we consider the function h that measures the constraint infeasibility in each point $x\in\mathbb{R}^n$, $h(x)=\|c(x)\|$ where $\|.\|$ denotes the Euclidean norm. Such a kind of optimization problems encompasses

N. Echebest
Department of Mathematics, FCE, University of La Plata, Argentina
E-mail: opti@mate.unlp.edu.ar

M.L. Schuverdt
CONICET, Department of Mathematics, FCE, University of La Plata, Argentina
E-mail: schuverd@mate.unlp.edu.ar

R.P. Vignau
Department of Mathematics, FCE, University of La Plata, Argentina
E-mail: vignau@mate.unlp.edu.ar

many real-world problems arising in different fields like e.g. computational mathematics, physics and engineering, in which it is necessary to minimize functions whose derivatives are not available (see e.g. [1,6,19]). Unconstrained techniques based on local explorations, line searches or quadratic models have been suitably adapted to box-constrained and linearly constrained derivative-free optimization [2,5,7,20–22,31,32]. Problems with more general constraints are more difficult because they need to obtain optimality and feasibility controlling the number of function evaluations of the objective function and the nonlinear constraints. Derivative-free methods for more general constraints were addressed by means of augmented Lagrangian approaches in [8,23,24].

Modern inexact restoration (IR) methods for smooth constrained optimization proceed in two phases [15,26–28]. In the restoration phase, feasibility is improved without evaluations of the objective function at all. In the optimization phase, the objective function or a Lagrangian function is minimized. One of the more attractive features of the IR method is that the theory allows us to use any efficient algorithm to perform each phase. Optimality and feasibility can be combined using penalty functions, Augmented Lagrangians or can be treated more independently. Inexact restoration algorithms described by Martínez [26] and by Martínez and Pilotta [27,28], measure the progress by a merit function. Gonzaga, Karas and Vanti [15] have proposed an inexact restoration algorithm which uses a filter strategy for evaluating candidate points. This idea was proposed by Fletcher and Leyffer in [13] in other context.

A recently article [4] uses the IR method for solving a nonlinear derivative-free optimization problem where the derivatives of the constraints are available but the derivatives of the objective function are not. In this case, the second phase must be solved using derivative-free methods. An algorithm introduced by Kolda, Lewis and Torczon [20] for linearly constrained derivative-free optimization is employed for that purpose.

In this paper we propose a derivative-free method, based on the inexact restoration approach introduced in [15]. There the authors define a globally convergent filter method for nonlinear programming considering available the derivatives of the objective function and the constraints. That filter method belongs to the class of methods that treat f and h as two independent objectives. Each iteration proceeds in two phases: the restoration or feasibility phase in which feasibility must be improved without using the objective function and the optimization phase in which the objective function on a tangent approximation to the constraints must be minimized. As mentioned in [15], the filter algorithms define a forbidden region by memorizing the pairs $(f(x^k), h(x^k))$ from well chosen former iterations, avoiding points dominated by those by using the usual Pareto domination rule: “x dominates y if and only if $f(y) \geq f(x)$ and $h(y) \geq h(x)$”. For bibliography on filter methods see for example [12,13,15] and the references therein.

The algorithm developed in this work is based on models built by multivariate interpolation of the objective and the constraint functions [7], which is one of the main differences with [15].

The restoration phase must solve an underdetermined nonlinear system with bound constraints. In our implementation we performed this phase using the derivative-free method developed in [11].

On the other hand, the optimization phase must solve a derivative-free optimization problem with linear constraints. We shall use a linear constrained trust-region algorithm in which the derivative of the objective function is approximated by a model obtained by linear interpolation.

This paper is organized as follows. In Section 2 we present the hypotheses, concepts and some results that are fundamental throughout the work. Also we define the Derivative-Free Filter algorithm (DFF) for solving (1). In Section 3 we present the internal algorithms used in DFF and we show that they satisfy certain conditions that will be used in the analysis of the convergence. In Section 4 we show the global convergence results. In Section 5 we describe implementation details and we show some numerical experiments. Finally, Section 6 is devoted to conclusions and lines for future research.

Notation.

- $\|.\|$ denotes the Euclidean norm.
- Given two non-negative functions $g_1, g_2 : X \to \mathbb{R}$, $X \subset \mathbb{R}^n$, we denote $g_1(x) = O(g_2(x))$ (or equivalently $g_2(x) = \Omega(g_1(x))$) in $\Gamma \subset X$ if there exists $M > 0$ such that $g_1(x) \leq M g_2(x)$ for all $x \in \Gamma$.

2 Derivative-Free Filter Algorithm

We shall develop an algorithm which generates sequences $\{x^k\}, \{z^k\}$ in $\mathbb{R}^n$ and in order to obtain our global convergence we shall assume the following hypotheses.

General hypotheses:

(H1) The iterates x^k and z^k remain in a convex compact domain $X \subset \mathbb{R}^n$.
(H2) The functions f, c_i for $i = 1, \ldots, m$ are continuously differentiable in an open set containing X.
(H3) The functions ∇f, ∇c_i for $i = 1, \ldots, m$ are Lipschitz continuous in an open set containing X with constants L_1, $L_2 > 0$, respectively:

$$\|\nabla f(x) - \nabla f(y)\| \leq L_1 \|x - y\|$$

$$\|\nabla c_i(x) - \nabla c_i(y)\| \leq L_2 \|x - y\|, \quad \text{for } i = 1, \ldots, m$$

for all x, y in the open set containing X.

Before going further into details of the algorithm, we first introduce some concepts and results of multivariate polynomial interpolation models of the objective function and constraints that we make use throughout and that can be found to a more extent in Conn, Scheinberg and Vicente [6].

Each interpolation set $Y = \{y^0, y^1, \ldots, y^n\} \subset \mathbb{R}^n$, which is contained in the ball $B(y^0, \Delta(Y))$ centered at y^0 and with radius $\Delta(Y) = \max_{1 \leq i \leq n} \|y^i - y^0\|$, is "poised" for linear interpolation, i.e., the matrix of directions $S = [y^1 - y^0 \;\; y^2 - y^0 \; \ldots \; y^n - y^0]^T$ is nonsingular. The definition of poisedness is independent of the basis for the space of linear polynomials of degree 1. Hence, if Y is poised for the natural basis then it is poised for any other basis chosen [6, Ch. 2].

The simplex gradient of f at y^0 is defined by $\nabla_s f(y^0) = S^{-1} \delta f(Y)$ where $\delta f(Y) = (f(y^1) - f(y^0), f(y^2) - f(y^0), \ldots, f(y^n) - f(y^0))^T$.

If we consider $m_f(x) = f(y^0) + g_f^T (x - y^0)$ as the linear interpolating model of $f(x)$ on Y then we have that $g_f = \nabla_s f(y^0)$ [6]. Therefore, the simplex gradient of f is closely related to linear multivariate polynomial interpolation.

The geometrical properties of Y determine the quality of the corresponding g_f as an approximation to the exact gradient of the objective function. We are interested in the quality of $m_f(x)$ and g_f in the ball $B(y^0,\Delta(Y))$.

The definition of poisedness gives a threshold to the difference between the functions and their interpolation models. Then, for all $x \in B(y^0,\Delta(Y))$, considering the scaled matrix $\bar{S} = \dfrac{S}{\Delta(Y)}$, we have that

$$|f(x) - m_f(x)| \leq \kappa_{ef}\Delta^2(Y), \tag{2}$$
$$\|\nabla f(x) - \nabla m_f(x)\| \leq \kappa_{eg}\Delta(Y), \tag{3}$$

where $\kappa_{eg} = L_1(1+\frac{\sqrt{n}}{2}\|\bar{S}^{-1}\|)$ and $\kappa_{ef} = \kappa_{eg} + \frac{L_1}{2}$, which are given in Theorem 2.11 and Theorem 2.12 in [6].

Similarly, under the previous hypotheses, if we consider for all $j = 1,\ldots,m$, $m_{c_j}(x) = c_j(y^0) + g_{c_j}^T(x-y^0)$ as the linear interpolating model of $c_j(x)$ on Y then we have that $g_{c_j} = \nabla_s c_j(y^0)$ and the following error bounds

$$|c_j(x) - m_{c_j}(x)| \leq \kappa_{ec}\Delta^2(Y), \tag{4}$$
$$\|\nabla c_j(x) - \nabla m_{c_j}(x)\| \leq \kappa_{eg_c}\Delta(Y), \tag{5}$$

where $\kappa_{eg_c} = L_2(1+\frac{\sqrt{n}}{2}\|\bar{S}^{-1}\|)$ and $\kappa_{ec} = \kappa_{eg_c} + \frac{L_2}{2}$.

If we consider as an approximation of $J_c(y)$ the matrix $A(y)$, whose j-th row is the transpose of $\nabla m_{c_j}(y)$ then we have that

$$\|J_c(y) - A(y)\| \leq \kappa_{eJ_c}\Delta(Y), \tag{6}$$

where $\kappa_{eJ_c} = \sqrt{m}\ \kappa_{eg_c}$.

We assume that it is possible to maintain the constants κ_{ef}, κ_{eg} and κ_{eJ_c} uniformly bounded along the iterative process of our algorithm [6, Ch. 3 and 6].

Given an iterate z^k we consider the following hypothesis

(H4) The simplex gradient used to approximate the objective function gradient satisfies the error bound : $\|\nabla f(z^k) - \nabla_s f(z^k)\| \leq k_{eg}\Delta_f^k$ where Δ_f^k is the radius of the ball that contains the interpolation points.
The simplex derivatives used to approximate the true Jacobian satisfy the error bound : $\|J_c(z^k) - A(z^k)\| \leq k_{eJ_c}\Delta_c^k$ where Δ_c^k is the radius of the ball that contains the interpolation points.

The global convergence result of the method in [15] is obtained without discussing details of the algorithms used in the internal phases. The authors proved that their algorithm produces feasible points $\bar{x}$ satisfying

$$\liminf_{x\to\bar{x}} \|P_{T(x)}(x - \nabla f(x)) - x\| = 0, \tag{7}$$

where $P_{T(z)}(w)$ is the orthogonal projection of $w \in \mathbb{R}^n$ onto the closed set

$$T(z) = \{x \in \mathbb{R}^n : J_c(z)(x-z) = 0\}$$

that is a linearization of the set $\{x \in \mathbb{R}^n : c(x) = c(z)\}$ at the point z.

The direction $P_{T(z)}(z-\nabla f(z))-z$ appears as a sequential optimality condition in the Approximate Gradient Projected condition defined by Martínez and Svaiter in [29].

In this paper we address nonlinear problems in which the derivatives of the involved functions are not available. When this is the case we can not compute in an exact form the set $T(z)$ and the gradient of the objective function.

Thus, in this context, we will be able to prove that our derivative-free filter algorithm generates a sequence $\{x^k\}$ which has a feasible limit point $\bar{x} \in \mathbb{R}^n$, $\bar{x} = \lim_{k \in \mathscr{K}} x^k$ for some infinite subset $\mathscr{K} \subset \mathbb{N}$, satisfying

$$\lim_{k \in \mathscr{K}} \|d_c(x^k)\| = 0 \tag{8}$$

where $d_c(z) = P_{L(z)}(z - \nabla_s f(z)) - z$ and $L(z) = \{x \in \mathbb{R}^n : A(z)(x-z) = 0\}$.
This feasible point $\bar{x}$ will be called *quasi-stationary* throughout this work.

Now, following the ideas in [15], we present the inexact restoration derivative-free filter algorithm with no specification of the internal algorithms.

This algorithm constructs a sequence $F_0 \subset F_1 \subset \ldots \subset F_k \subset \ldots$ of filter sets composed of pairs $(f_j, h_j) \in \mathbb{R}^2$. In the following, we also mention the sets of forbidden points, $\mathscr{F}_k \subset \mathbb{R}^n$, $\mathscr{F}_k = \{x \in \mathbb{R}^n : f(x) \geq f_j, h(x) \geq h_j, \text{ for some } (f_j, h_j) \in F_k\}$, which are formally defined in each step of algorithm for clarity, but are never actually constructed. Each iteration starts with a filter and the corresponding forbidden region.

Given an iterate x^k, the filter slack at x^k is defined by

$$H_k = \min\{1, \min\{h_j : (f_j, h_j) \in F_k, f_j \leq f(x^k)\}\}. \tag{9}$$

Algorithm 1. *Derivative-Free Filter Algorithm (DFF).*
Given $x^0 \in \mathbb{R}^n$, $F_0 = \emptyset$, $\mathscr{F}_0 = \emptyset$, $\alpha \in (0,1)$, $\beta > 0$, $\varepsilon_f > 0$, $\varepsilon_I > 0$, $\{\delta_k\}_{k \in \mathbb{N}}$, $\delta_k > 0$, $\delta_k \to 0$. *Set* $k \leftarrow 0$.

Step 1 : Define $(\widetilde{f}, \widetilde{h}) = (f(x^k) - \alpha h(x^k), (1-\alpha) h(x^k))$.
Construct the set $\overline{F}_k = F_k \cup \{(\widetilde{f}, \widetilde{h})\}$.
Define the set $\overline{\mathscr{F}}_k = \mathscr{F}_k \cup \{x \in \mathbb{R}^n : f(x) \geq \widetilde{f}, h(x) \geq \widetilde{h}\}$.

Step 2 : **Restoration Phase**
If $h(x^k) = 0$ *then set* $z^k = x^k$.
Otherwise, compute $z^k \notin \overline{\mathscr{F}}_k$ *such that* $h(z^k) < (1-\alpha) h(x^k)$ *and* $\|z^k - x^k\| \leq \beta h(x^k)$. *If it is impossible then stop without success.* **END**.

Step 3 : **Optimization Phase**
3.1 Construct or update $Y_c^k = \{z^k, y_c^1, \ldots, y_c^n\}$, *a set of interpolation points centered at* z^k, *such that* $\Delta_c^k = \max_{i=1,\ldots,n} \{\|y_c^i - z^k\|\}$ *verifies* $\Delta_c^k \leq \beta \min\{\max\{h(x^k), H_k\}, \delta_k\}$.
Compute $A(z^k) = A_k$ *using simplex derivatives, by interpolation on* Y_c^k.
Define $L(z^k) = \{x \in \mathbb{R}^n : A_k(x - z^k) = 0\}$.
Construct or update $Y_f^k = \{z^k, y_f^1, \ldots, y_f^n\}$, *a set of interpolation points centered at* z^k, *such that* $\Delta_f^k = \max_{i=1,\ldots,n} \{\|y_f^i - z^k\|\}$ *verifies* $\Delta_f^k \leq \delta_k$.
Compute $\nabla_s f(z^k)$ *by interpolation on* Y_f^k *and* $d_c(z^k) = P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k$.
3.2 If $h(x^k) = 0$, $\max\{\Delta_f^k, \Delta_c^k\} < \varepsilon_I$ *and* $\|d_c(z^k)\| < \varepsilon_f$ *then stop with finite convergence.*
3.3 Compute, by an algorithm without derivatives, $x_T \notin \overline{\mathscr{F}}_k$ *such that* $x_T \in L(z^k)$ *and* $f(x_T) \leq f(z^k)$.

If $z^k = x^k$ and there is not a x_T such that $f(x_T) < f(z^k)$ then set $\Delta_f^k = \alpha\Delta_f^k$, $\Delta_c^k = \alpha\Delta_c^k$ and go to step 3.1.

Otherwise, define $x^{k+1} = x_T$.

Step 4 : **Filter Update**

If $f(x^{k+1}) < f(x^k)$ then $F_{k+1} = F_k$, $\mathscr{F}_{k+1} = \mathscr{F}_k$ (f-iteration).

Else, $F_{k+1} = \overline{F}_k$, $\mathscr{F}_{k+1} = \overline{\mathscr{F}}_k$ (h-iteration).

Set $k \leftarrow k+1$, go to Step 1.

Observe that, as it was made in [15], at the beginning of each iteration, the pair $(f(x^k) - \alpha h(x^k), h(x^k) - \alpha h(x^k))$ is temporarily introduced in the filter. After the complete successful iteration this entry will become permanent in the filter only if the iteration does not produce a decrease in f.

In [26], under suitable assumptions, J.M. Martínez has shown that a point that satisfies the feasibility phase requirements exists. Considering this, if $h(x^k) \neq 0$, it is plausible to believe that a point z^k satisfying $h(z^k) < (1-\alpha)h(x^k)$ and $\|z^k - x^k\| \leq \beta h(x^k)$ could be found, for example, by a Broyden-like method to solve the nonlinear system defined by the constraints.

In order to accept z^k, it is necessary to check if $z^k \notin \overline{\mathscr{F}}_k$. Since the pair $(f(z^k), h(z^k))$ is not dominated by $(\widetilde{f}, \widetilde{h})$, it is only necessary to verify that $z^k \notin \mathscr{F}_k$. Since $x^k \notin \mathscr{F}_k$, $\mathscr{F}_k$ is closed and the restored point has bounded distance from x^k, it is reasonable to believe that the algorithm has possibilities to complete the restoration phase. However, we do not have guaranties that such point would be found, and so the stopping criterion in Step 2 is essential.

Furthermore, when $h(x^k) = 0$ it is necessary to find x^{k+1} satisfying $f(x^{k+1}) < f(x^k)$, to fulfill the condition that $x^{k+1} \notin \overline{\mathscr{F}}_k$. Since we are not working with the true derivatives, the computed direction $d_c(z^k)$ could not be a descent direction of f in z^k over $L(z^k)$, although it is not null. This can happen because the simplex gradients are not good approximations of the true gradients. Consequently the procedure used in the optimization phase may not be able to find a point x_T such that $f(x_T) < f(z^k)$. If $z^k \neq x^k$, as $z^k \notin \overline{\mathscr{F}}_k$, it is possible to accept $x_T = z^k$ and $x^{k+1} = z^k$. But when $z^k = x^k$ and the algorithm can not find a point x_T such that $f(x_T) < f(z^k)$, we propose to restart the optimization phase recomputing the simplex gradient of f and the matrix A_k with the new radiuses $\alpha\Delta_f^k$ and $\alpha\Delta_c^k$ of the interpolation points.

The following lemma gives conditions for which $d_c(z^k)$ is a descent direction of f in z^k over $L(z^k)$.

Lemma 1 *Given $\varepsilon > 0$, $z^k \in \mathbb{R}^n$, if $\|d_c(z^k)\| > \varepsilon$ and $\|\nabla f(z^k) - \nabla_s f(z^k)\| < \frac{\varepsilon}{4}$ then*

$$\|z^k - P_{L(z^k)}(z^k - \nabla f(z^k))\| > \frac{3}{4}\,\varepsilon, \tag{10}$$

$$\nabla^T f(z^k)\, d_c(z^k) < -\frac{1}{4}\|d_c(z^k)\|^2. \tag{11}$$

Proof. Since the projection $P_{L(z^k)}$ is non-expansive,

$$\|P_{L(z^k)}(z^k - \nabla f(z^k)) - P_{L(z^k)}(z^k - \nabla_s f(z^k))\| \leq \|\nabla f(z^k) - \nabla_s f(z^k)\|,$$

then it follows that

$$\|z^k - P_{L(z^k)}(z^k - \nabla_s f(z^k))\| \leq \|z^k - P_{L(z^k)}(z^k - \nabla f(z^k))\| + \|\nabla f(z^k) - \nabla_s f(z^k)\|. \qquad (12)$$

Then we have that

$$\|z^k - P_{L(z^k)}(z^k - \nabla f(z^k))\| \geq \|z^k - P_{L(z^k)}(z^k - \nabla_s f(z^k))\| - \|\nabla f(z^k) - \nabla_s f(z^k)\| > \frac{3}{4}\,\varepsilon > 0,$$

as we wanted to prove.
Since $\nabla^T f(z^k)\; d_c(z^k) = (\nabla f(z^k) - \nabla_s f(z^k))^T\; d_c(z^k) + \nabla_s^T f(z^k)\; d_c(z^k)$, then

$$\nabla^T f(z^k)\; d_c(z^k) \leq \|d_c(z^k)\|\; \|\nabla f(z^k) - \nabla_s f(z^k)\| + \nabla_s^T f(z^k)\; d_c(z^k).$$

Therefore, considering

$$\nabla_s^T f(z^k)\; d_c(z^k) \leq -\frac{\|d_c(z^k)\|^2}{2}, \qquad (13)$$

which is obtained by a similar form to one of [27, Sec. 2.6, page 140] replacing $\nabla f(z^k)$ by $\nabla_s f(z^k)$, we obtain that

$$\nabla^T f(z^k)\; d_c(z^k) \leq \|d_c(z^k)\|^2 \Big(\frac{\|\nabla f(z^k) - \nabla_s f(z^k)\|}{\|d_c(z^k)\|} - \frac{1}{2} \Big).$$

Hence, we get $\nabla^T f(z^k)\; d_c(z^k) < \|d_c(z^k)\|^2(\frac{1}{4} - \frac{1}{2}) = -\frac{1}{4}\|d_c(z^k)\|^2$. Therefore, under the hypotheses given, $d_c(z^k)$ is a descent direction of f in z^k. □

Remark 1 Under the hypotheses of the previous lemma, if z^k is not in $\overline{\mathscr{F}}_k$, which is a closed set, then there must exist $\Delta > 0$ and $t > 0$ such that if $t\|d_c(z^k)\| < \Delta$ then $z^k + td_c(z^k)$ does not fall into the region $\overline{\mathscr{F}}_k$ and $f(z^k + td_c(z^k)) < f(z^k)$. Similarly when $h(x^k) = 0$, by construction $z^k = x^k$ and $z^k \in \overline{\mathscr{F}}_k$. In this case, since $z^k \notin \mathscr{F}_k$, which is a closed set, there exist $\Delta > 0$ and $t > 0$ such that if $t\|d_c(z^k)\| < \Delta$ then $z^k + td_c(z^k)$ does not fall into the region $\mathscr{F}_k$. Furthermore, since $f(z^k + td_c(z^k)) < f(z^k)$ it obtains that $z^k + td_c(z^k) \notin \overline{\mathscr{F}}_k$.

Lemma 2 *Algorithm 1 is well defined.*

Proof. If the method used in the restoration phase is not able to find a point z^k satisfying the required conditions then the Algorithm 1 stops.

In the optimization phase, when $z^k \neq x^k$ there always exists $x_T \notin \overline{\mathscr{F}}_k$ such that $f(x_T) \leq f(z^k)$ since $z^k \notin \overline{\mathscr{F}}_k$ and then it is possible to accept $x_T = z^k$.

When x^k is feasible, $z^k = x^k$, if it is possible to find x_T with $f(x_T) < f(z^k)$ then x^{k+1} is defined. If that is not possible then the algorithm restarts the optimization phase with smaller Δ_f^k and Δ_c^k, with the aim of improving the approximation of the gradients of f and c_i, for $i = 1,\dots,m$. In this case, in a finite number of iterations the radiuses Δ_f^k and Δ_c^k will become sufficiently small and if $\|d_c(z^k)\|$ is large enough, by Lemma 1 and Remark 1, it is possible to obtain $x_T \notin \overline{\mathscr{F}}_k$ such that $f(x_T) < f(x^k)$ and then x^{k+1} is defined. Otherwise, if $\|d_c(z^k)\| < \varepsilon_f$ and $\max\{\Delta_f^k, \Delta_c^k\} < \varepsilon_I$ then the algorithm finishes satisfying the finite termination criterion. □

Remark 2 When $h(x^k) > 0$, in the previous lemma we have used the possibility to accept $x^{k+1} = z^k$. When this happens an infinite number of iterations a feasible limit point is obtained. Until this moment, the internal algorithms have not been given. In the following section we will study the characteristics of the limit points using the properties of the internal algorithms.

As it was mentioned in [15] there are some facts that follow directly from the construction of the algorithm:

Fact 1. Given $k \in \mathbb{N}, x^{k+p} \notin \mathscr{F}_{k+1}$ for all $p \geq 1$.
Fact 2. Given $k \in \mathbb{N}$, at least one of the following two situations must occur:
1. $h(x^{k+1}) < (1-\alpha)h(x^k)$.
2. $f(x^{k+1}) < f(x^k) - \alpha h(x^k)$.

Fact 3. Given $k \in \mathbb{N}, h_j > 0$ for all $j \in \mathbb{N}$ such that $(f_j, h_j) \in F_k$. Consequently $H_k > 0$ for all $k \in \mathbb{N}$.

Remark 3 By definition of H_k, $H_k \leq 1$. Therefore, when x^k is in a neighborhood of a feasible point, assuming $h(x^k) < 1$, if $H_k = 1$ then $h(x^k) \leq H_k$ holds. If $H_k < 1$ then there exists a $h_j < 1$ such that $(f_j, h_j) \in F_k$, $f_j \leq f(x^k)$, such that $H_k = h_j$. In this case, since $x^k \notin \mathscr{F}_k$ and $f(x^k) \geq f_j$, it must be $h(x^k) < h_j$. Hence, if x^k is in a neighborhood of a feasible point then $h(x^k) \leq H_k$ holds.

3 Internal algorithms

Inexact restoration methodology gives the possibility of using different methods to solve each phase. In this section we describe the algorithms that we use in each phase. We will also show that they verify the conditions required to obtain global convergence of DFF.

3.1 Restoration phase

We use the BCDF-QNB algorithm [11] in the restoration phase of the DFF algorithm. BCDF-QNB (Box-Constrained Derivative-Free Quasi Newton), based on the Broyden update formula, is a derivative-free method for solving underdetermined nonlinear systems with bound constraints.

Given an iterate x^k, in Step 2 of DFF we apply BCDF-QNB starting from the initial point $y^0 = x^k$, until it finds a new point $z^k \notin \overline{\mathscr{F}}_k$ satisfying the descent condition $h(z^k) < (1-\alpha)h(x^k)$ and $\|z^k - x^k\| \leq \beta h(x^k)$ for fixed parameters $0 < \alpha < 1$, $\beta > 0$.

BCDF-QNB generates a sequence $\{y^j\}$, for $j = 0, 1, 2, \ldots$, with $y^j \in \Omega_k$, being $\Omega_k = \{y \in \mathbb{R}^n : \|y - x^k\|_\infty \leq \frac{\beta}{\sqrt{n}} h(x^k)\}$. At each iterate y^j, this algorithm computes a direction d_j, considering two possibilities: in a first attempt, as the solution of the constrained linear system

$$B_j d + c(y^j) = 0 \quad \text{and} \quad y^j + d \in \Omega_k, \tag{14}$$

if this is possible. Otherwise, the direction is computed as an approximate solution of the problem

$$\min_{y^j + d \,\in\, \Omega_k} \|B_j d + c(y^j)\| \tag{15}$$

where B_j is the matrix defined as:

$$B_j = B_{j-1} + \frac{(w_j - B_{j-1}s^j)(s^j)^T}{\|s^j\|^2} \tag{16}$$

where $w_j = c(y^j) - c(y^{j-1})$, $s^j = y^j - y^{j-1}$.

Once the current direction d_j is computed, the line search algorithm looks for a step length $\lambda_j \leq 1$ such that

$$h(y^j + \lambda_j d_j)^2 \leq \max_{0\leq i\leq M-1} h(y^{j-i})^2 + \eta_j - \gamma\lambda_j^2 \|d_j\|^2 \tag{17}$$

where M is a positive integer, $0 < \gamma < 1$ and $\sum_{j=0}^{\infty} \eta_j = \eta < \infty$, $\eta_j > 0$. This procedure is a combination of the well known nonmonotone line search strategy for unconstrained optimization introduced by Grippo, Lampariello and Lucidi [17] with the Li-Fukushima derivative-free line search scheme in [25]. The combined strategy produces a robust nonmonotone derivative-free line search that takes into account the advantages of both schemes. Under suitable conditions we have established in [11] the global convergence results for the BCDF-QNB method.

We describe the application of BCDF-QNB for solving the Restoration Phase.

Algorithm 2. *BCDF-QNB*

```
Given $x^k \in \Omega_k$, $0 < \alpha < 1$, $\beta > 0$, $W_k$ an approximation of $J_c(x^k)$, $0 < \gamma < 1$, $M \in \mathbb{N}$, $M > 0$,
    $\eta = \sum_{j=0}^{\infty} \eta_j < \infty$, $\eta_j > 0$, $0 \leq \theta_0 < \bar{\theta} < 1$, $ind = 0$, $imax > 0$, $imax \in \mathbb{N}$, $MaxIter > 0$.
    Set $j \leftarrow 0$, $y^0 = x^k$, $B_0 = W_k$.
Step 1: If $h(y^j) < (1-\alpha)h(x^k)$ and $y^j \notin \overline{\mathscr{F}}_k$, define $z^k = y^j$ and return with success.
    If $j > Maxiter$ then return without success.
Step 2: Computing the matrix $B_j$
    If $j > 0$ and $ind < imax$ compute $B_j$ using the Broyden update (16). If $ind = imax$
    compute $B_j$ by finite differences as an approximation to the Jacobian matrix in $y^j$.
Step 3: Computing the direction $d_j$
    3.1: Find $d$ satisfying (14).
        If such direction $d$ is found, define $d_j = d$, $\theta_{j+1} = \theta_j$, $ind = 0$ and go to Step 4.
    3.2: Find an approximate solution $d$ of the problem (15).
        If $d$ satisfies $\|B_j d + c(y^j)\| \leq \theta_j \|c(y^j)\|$, define $d_j = d$, $\theta_{j+1} = \theta_j$, $ind = 0$ and go
        to Step 4.
    3.3: Set $d_j = 0$, $y^{j+1} = y^j$, $\theta_{j+1} = \frac{\theta_j + \bar{\theta}}{2}$.
        If $ind < imax$, set $ind \leftarrow ind + 1$ and go to Step 5.
        If $ind = imax$, define $\bar{\theta} = \frac{\bar{\theta}+1}{2}$. Set $ind \leftarrow 0$ and go to Step 5.
Step 4: Find $\lambda_j$ and $y^{j+1} = y^j + \lambda_j d_j$, $0 < \lambda_j \leq 1$, using the derivative-free nonmonotone
    line search algorithm (Algorithm 1 in [11]), satisfying (17).
Step 5: Set $j \leftarrow j + 1$ and go to Step 1.
```

The matrix W_0 is an approximation of $J_c(x^0)$, which is obtained by finite differences. The initial matrix W_k, $k > 0$, is the updated Broyden matrix of A_{k-1}, where A_{k-1} is the matrix defined at z^{k-1} in the optimization phase.

Remark 4 Since $\{y^j\} \subset \Omega_k$, the obtained z^k satisfies the condition $\|z^k - x^k\| \leq \beta h(x^k)$, $\beta > 0$.

As a result, more formally, the procedure generates iterates that verify the following condition.

(C1) Restoration step condition: At all iterations $k \in \mathbb{N}$, the restoration step satisfies

$$\|z^k - x^k\| = O(h(x^k)). \tag{18}$$

Using (C1) and that ∇f is bounded in X, it follows that

$$|f(z^k) - f(x^k)| = O(\|z^k - x^k\|) = O(h(x^k)). \tag{19}$$

3.2 Optimization phase

Given $z^k \in X$, generated in the restoration phase, Step 3.3 of DFF must find $x^{k+1} \in L(z^k)$ such that $f(x^{k+1}) \leq f(z^k)$ and $x^{k+1} \notin \overline{\mathscr{F}}_k$ employing a derivative-free method.

We shall describe a linear trust region method and then we show that the resulting step satisfies a special condition needed for obtaining convergence.

At each iterate z^k, the trust region algorithm associated to z^k uses the linear model

$$m_k(x) = f(z^k) + \nabla_s^T f(z^k)(x - z^k)$$

where the simplex gradient of the objective function is considered.

The trust region step uses a radius $\Delta > 0$ and solves the problem

$$\begin{array}{c} \text{minimize } m_k(x) \\ \text{subject to } x \in L(z^k) \\ \qquad \|x - z^k\| \leq \Delta. \end{array}$$

As the model is linear we know that the solution of this problem is a point $z^k + d(z^k, \Delta)$ such that

$$d(z^k, \Delta) = \Delta \frac{d_c(z^k)}{\|d_c(z^k)\|} \tag{20}$$

if $d_c(z^k) \neq 0$, where $d_c(z^k)$ is the projected gradient direction defined by $P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k$.

We define the *predicted reduction* produced by the step $d(z^k, \Delta)$ as

$$pred(z^k, \Delta) = m_k(z^k) - m_k(z^k + d(z^k, \Delta)) \tag{21}$$

and the *actual reduction* of f as

$$ared(z^k, \Delta) = f(z^k) - f(z^k + d(z^k, \Delta)). \tag{22}$$

The step $d(z^k, \Delta)$ is only accepted if the sufficient decrease condition is satisfied, i.e,

$$ared(z^k, \Delta) > \eta \; pred(z^k, \Delta), \tag{23}$$

for a given $\eta \in (0, 1)$.

Since $pred(z^k, \Delta) = -\nabla_s^T f(z^k) d(z^k, \Delta) = -\nabla_s^T f(z^k) \frac{d_c(z^k)}{\|d_c(z^k)\|} \Delta$, considering (13), we get

$$pred(z^k, \Delta) \geq \frac{\Delta}{2} \|d_c(z^k)\|. \tag{24}$$

We briefly describe the linear trust region method for solving the optimization phase.

Algorithm 3. *Minimization on $L(z^k)$*

```
Given $\eta \in (0,1)$, $\Delta_{min} > 0$, $x^k$, $z^k \notin \mathscr{F}_k$, $d_c(z^k)$, $\Delta \geq \Delta_{min} > 0$, $tol > 0$.
    Set $x^+ = z^k$.
```

```
While ($\|d_c(z^k)\|\Delta > tol$ and $f(x^+) \geq f(z^k)$ ) do
    Compute $d = d(z^k,\Delta)$, $pred(z^k,\Delta)$ and $ared(z^k,\Delta)$ as in (20), (21) and (22) respectively.
      If $ared(z^k,\Delta) > \eta \; pred(z^k,\Delta)$ and $z^k + d \notin \overline{\mathscr{F}}_k$, define $x^+ = z^k + d$.
      Else, set $\Delta = \frac{\Delta}{2}$.
End While.
If $f(x^+) < f(z^k)$ or $z^k \neq x^k$, define $x_T = x^+$, $\Delta_k = \Delta$.
    Otherwise, return without success.
```

The procedure terminates in a finite number of steps with $f(x^+) < f(z^k)$ or with $x^+ = z^k$. In particular, it finishes in the first iteration when $\|d_c(z^k)\| = 0$. If it finishes with $x^+ = z^k$ and $z^k = x^k$, when x^k is feasible, then it is not possible to define $x_T \notin \overline{\mathscr{F}}_k$. Hence it returns without success and so Δ_f^k and Δ_c^k are reduced in Algorithm 1, which means that better models are built. In other cases successfully returns with $x_T = x^+$.

Now we study the optimality step near a feasible non quasi-stationary limit point $\bar{x} \in X$.

Lemma 3 *Let $\bar{x} \in X$ be a feasible non quasi-stationary limit point. Then there exists a neighborhood $\widetilde{V}$ of $\bar{x}$, $\widetilde{\Delta} > 0$ and a constant $\widetilde{c} > 0$ such that for any $z^k \in \widetilde{V}$ and for any $\Delta \in (0, \widetilde{\Delta})$,*

$$ared(z^k,\Delta) > \eta \; pred(z^k,\Delta) \geq \eta \widetilde{c} \Delta.$$

Proof. As $\bar{x}$ is a non quasi-stationary limit point, there exists a neighborhood $\widetilde{V}$ such that for $z^k \in \widetilde{V}$, $\|d_c(z^k)\| \geq \tilde{\varepsilon} > 0$ for all $k \geq k_0$.

Since f is continuously differentiable and ∇f is Lipschitz continuous, we know that

$$ared(z^k,\Delta) = f(z^k) - f(z^k + d(z^k,\Delta)) \geq (-\nabla f(z^k))^T d(z^k,\Delta) - L_1 \Delta^2 =$$

$$(-\nabla f(z^k) + \nabla_s f(z^k))^T d(z^k,\Delta) - (\nabla_s f(z^k))^T d(z^k,\Delta) - L_1 \Delta^2.$$

In particular, if $\|d_c(z^k)\| \geq \tilde{\varepsilon}$, using (24) we have that $-\nabla_s^T f(z^k) d(z^k,\Delta) = pred(z^k,\Delta) \geq \frac{\Delta}{2} \|d_c(z^k)\| \geq \frac{\Delta}{2} \tilde{\varepsilon}$. Then, considering

$$pred(z^k,\Delta) = \eta(-\nabla_s^T f(z^k) d(z^k,\Delta)) + (1-\eta)(-\nabla_s^T f(z^k) d(z^k,\Delta)),$$

it obtains $pred(z^k,\Delta) \geq \eta(-\nabla_s^T f(z^k) d(z^k,\Delta)) + (1-\eta)\frac{\Delta}{2}\tilde{\varepsilon}$.

Hence

$$ared(z^k,\Delta) \geq \eta pred(z^k,\Delta) + (1-\eta)\frac{\Delta}{2}\tilde{\varepsilon} + (-\nabla^T f(z^k) + \nabla_s^T f(z^k)) d(z^k,\Delta) - L_1 \Delta^2.$$

By (H4), we have $\| -\nabla^T f(z^k) + \nabla_s^T f(z^k)\| \leq k_{eg} \Delta_f^k$. Since $\Delta_f^k \leq \delta_k$ and $\delta_k \to 0$, when k goes to infinity, there exists $k_1 \geq k_0$ such that for $k \geq k_1$, $k_{eg}\Delta_f^k < \frac{(1-\eta)}{4}\tilde{\varepsilon}$. Then,

$$ared(z^k,\Delta) > \eta pred(z^k,\Delta) - \frac{(1-\eta)}{4}\tilde{\varepsilon}\|d(z^k,\Delta)\| + (1-\eta)\frac{\Delta}{2}\tilde{\varepsilon} - L_1\Delta^2 \geq$$

$$\eta pred(z^k,\Delta) - \frac{(1-\eta)}{4}\tilde{\varepsilon}\Delta + (1-\eta)\frac{\Delta}{2}\tilde{\varepsilon} - L_1\Delta^2.$$

Hence, $ared(z^k,\Delta) > \eta \; pred(z^k,\Delta) + (1-\eta)\frac{\Delta}{4}\tilde{\varepsilon} - L_1\Delta^2$. Therefore if $\Delta < \widetilde{\Delta} = \frac{(1-\eta)}{4L_1}\tilde{\varepsilon}$ we obtain that $ared(z^k,\Delta) > \eta \; pred(z^k,\Delta)$ and $pred(z^k,\Delta) \geq \frac{\Delta}{2}\|d_c(z^k)\| \geq \widetilde{c}\Delta$ where $\widetilde{c} = \frac{\tilde{\varepsilon}}{2}$, as we wanted to prove. □

Remark 5 In the previous lemma we have seen that if z^k, the point found in restoration phase, is in the neighborhood of a non quasi-stationary feasible point, then it is possible to find a step $d(z^k,\Delta)$ by (20) such that $f(z^k+d(z^k,\Delta))<f(z^k)$. Furthermore, when z^k is not in $\overline{\mathscr{F}}_k$, which is a closed set, then there must be a $\Delta \le \tilde{\Delta}$ for which $z^k+d(z^k,\Delta)$ does not fall into the forbidden region $\overline{\mathscr{F}}_k$. Similarly when $h(x^k)=0$, by construction $z^k=x^k$ and $z^k \in \overline{\mathscr{F}}_k$. By Lemma 3 as $f(z^k+d(z^k,\Delta))<f(z^k)$ for all $\Delta \in (0,\tilde{\Delta})$, $z^k+d(z^k,\Delta) \notin \{x \in \mathbb{R}^n : f(x) \ge f(z^k),\ h(x)>0\}$. Then considering that $z^k \notin \mathscr{F}_k$, which is a closed set, we get a similar result to the case when z^k is not in $\overline{\mathscr{F}}_k$. Hence, under the hypothesis of Lemma 3, Algorithm 3 finds a point $x^+ \notin \overline{\mathscr{F}}_k$ and then defines $x_T = x^+$.

Lemma 4 *Suppose that the matrix A_k is computed as an approximation of $J_c(z^k)$ by simplex derivatives using an interpolation radius Δ_c^k. Then if $z^k+d \in L(z^k)$,*

$$|h(z^k+d)-h(z^k)| \le \kappa_{eJ_c}\Delta_c^k\|d\|+O(\|d\|^2). \tag{25}$$

Proof. Since $z^k+d \in L(z^k)$, $A_k d=0$, considering the general hypotheses we have that $\|c(z^k+d)-c(z^k)-J_c(z^k)d\| \le \sqrt{m}\,L_2\|d\|^2$. Then $\|c(z^k+d)-c(z^k)\| \le \|(J_c(z^k)-A_k)d\| + \sqrt{m}\,L_2\|d\|^2$.

Hence, $|\|c(z^k+d)\|-\|c(z^k)\|| \le \|c(z^k+d)-c(z^k)\| \le \|(J_c(z^k)-A_k)\|\|d\|+\sqrt{m}\,L_2\|d\|^2$. Therefore, considering (6), $|h(z^k+d)-h(z^k)| \le \kappa_{eJ_c}\Delta_c^k\|d\|+\sqrt{m}\,L_2\|d\|^2$, as we wanted to prove. □

The bound in (25) is $O(\|d\|)$ because we are not using true derivatives. A similar bound appears in [15], section 4.3, where the authors proposed a simplified tangential step.

Under the hypotheses of Lemmas 3 and 4 and the condition (C1) it can be established that the proposed procedure generates iterates that verify the following condition.

(C2) Optimality step condition: Given a feasible non quasi-stationary point $\bar{x} \in X$, there exists a neighborhood V of $\bar{x}$ such that for any iterate $x^k \in V$,

$$f(z^k)-f(x^{k+1})=\Omega(\sqrt{H_k}). \tag{26}$$

Lemma 5 *Let $\bar{x} \in X$ be a feasible non quasi-stationary limit point. Let assume that (C1) and the hypothesis of Lemma 4 hold. Then there exists a neighborhood V of $\bar{x}$ such that if $x^k \in V$ then*

$$f(z^k)-f(x^{k+1})=\Omega(\sqrt{H_k}),$$

where $x^{k+1}=x_T$, x_T is computed by Algorithm 3.

Proof. Let $\{x^k\}_{k\in\mathscr{K}}$ a subsequence such that $\lim\limits_{k\in\mathscr{K}} x^k=\bar{x}$.

By (C1) $\|x^k-z^k\|=O(h(x^k))$, as $h(x^k)$ tends to zero, it follows that $\lim\limits_{k\in\mathscr{K}} z^k=\bar{x}$.

Let $\widetilde{V} \subset X$ and $\widetilde{\Delta}>0$ be the neighborhood of $\bar{x}$ and the radius given by Lemma 3, such that for any $z^k \in \widetilde{V}$, $k \in \mathscr{K}$ and for any $\Delta \in (0,\widetilde{\Delta})$, $ared(z^k,\Delta) > \eta\ pred(z^k,\Delta) \ge \eta\tilde{c}\Delta$.

Algorithm 3 starts with a radius $\Delta \ge \Delta_{min}$ and computes $d(z^k,\Delta_j)$, $\Delta_j=2^{-j}\Delta$ for $j=0,1,\ldots$, until $z^k+d(z^k,\Delta_j) \notin \overline{\mathscr{F}}_k$ and $ared(z^k,\Delta_j) > \eta\ pred(z^k,\Delta_j)$. Then, define $\Delta_k=\Delta_j$.

Let us define $\widehat{\Delta}$ as the first Δ_j such that

$$ared(z^k,\Delta_j) > \eta\ pred(z^k,\Delta_j), \text{and} \tag{27}$$

$$z^k + d(z^k, \Delta_j) \notin \overline{\mathscr{F}}_k \quad \text{or} \quad f(z^k + d(z^k, \Delta_j)) \geq \widetilde{f}, \tag{28}$$

where $(\widetilde{f}, \widetilde{h}) = (f(x^k) - \alpha h(x^k), (1-\alpha)h(x^k))$ is the temporary entry in the filter.

Let us denote $\widehat{d} = d(z^k, \widehat{\Delta})$ and $\widehat{x} = z^k + \widehat{d}$. Note that $\widehat{\Delta} \geq \Delta_k$, and $\widehat{\Delta} > \Delta_k$ happens only when $f(\widehat{x}) \geq \tilde{f}$.

Observe that, from Lemma 4, for a fixed Δ we have that there is a constant $\kappa_{eJ_c}\Delta_c^k > 0$ such that

$$|h(z^k + d(z^k, \Delta)) - h(z^k)| \leq \kappa_{eJ_c}\Delta_c^k \|d(z^k, \Delta)\| + \sqrt{m}\, L_2 \|d(z^k, \Delta)\|^2.$$

By Remark 3 we know that if x^k is in a neighborhood of a feasible point then $h(x^k) \leq H_k$. So, considering that $\|d(z^k, \Delta)\| \leq \Delta$ and $\Delta_c^k \leq \beta \min\{\max\{h(x^k), H_k\}, \delta_k\}$ we have that

$$|h(z^k + d(z^k, \Delta)) - h(z^k)| \leq \kappa_{eJ_c}\beta H_k \Delta + \sqrt{m}\, L_2 \Delta^2. \tag{29}$$

Let us consider $\bar{\Delta}$ such that $\bar{\Delta} \leq \dfrac{\alpha}{4\beta\kappa_{eJ_c}}$ and $\bar{\Delta} < \dfrac{\tilde{\Delta}}{2}$.

(i) Assume that $\widehat{\Delta} \geq \bar{\Delta}$. Then, by (24),

$$pred(z^k, \widehat{\Delta}) \geq \frac{\widehat{\Delta}}{2} \|d_c(z^k)\| \geq \frac{\tilde{\varepsilon}}{2}\widehat{\Delta}.$$

By considering $\widetilde{c} = \frac{\tilde{\varepsilon}}{2}$ as in the proof of Lemma 3 we have that

$$pred(z^k, \widehat{\Delta}) \geq \widetilde{c}\widehat{\Delta} \geq \widetilde{c}\bar{\Delta}.$$

By definition of $\widehat{\Delta}$, (27) holds, then

$$f(z^k) - f(\widehat{x}) > \eta\, pred(z^k, \widehat{\Delta}) \geq \eta \widetilde{c} \bar{\Delta} = \Omega(1).$$

Hence, since $H_k \leq 1$, it follows

$$f(z^k) - f(\widehat{x}) = \Omega(\sqrt{H_k}).$$

(ii) Assume that $\widehat{\Delta} < \bar{\Delta}$. Then $2\widehat{\Delta} < 2\bar{\Delta} < \widetilde{\Delta}$ and $2\widehat{\Delta}$ does not verify (28). By Lemma 3,

$$ared(z^k, d(z^k, 2\widehat{\Delta})) > \eta\, pred(z^k, d(z^k, 2\widehat{\Delta}))$$

and, by (28) it follows that $z^k + d(z^k, 2\widehat{\Delta}) \in \overline{\mathscr{F}}_k$ and $f(z^k + d(z^k, 2\widehat{\Delta})) < \widetilde{f}$. Consequently by definition of H_k, we must have $h(z^k + d(z^k, 2\widehat{\Delta})) \geq H_k$.

By construction, $h(z^k) < (1-\alpha)h(x^k) \leq (1-\alpha)H_k$. Therefore,

$$h(z^k + d(z^k, 2\widehat{\Delta})) - h(z^k) \geq \alpha H_k.$$

Then, using (29)

$$\alpha H_k \leq h(z^k + d(z^k, 2\widehat{\Delta})) - h(z^k) \leq \kappa_{eJ_c}\beta H_k 2\widehat{\Delta} + 4\sqrt{m}\, L_2 \widehat{\Delta}^2,$$

we obtain

$$H_k \leq \frac{2\beta}{\alpha}\kappa_{eJ_c} H_k \widehat{\Delta} + O(\widehat{\Delta}^2) \leq \frac{1}{2}H_k + O(\widehat{\Delta}^2).$$

Hence

$$\frac{1}{2}H_k = O(\widehat{\Delta}^2) \quad \text{or} \quad \widehat{\Delta} = \Omega(\sqrt{H_k}).$$

Using Lemma 3 with $\widehat{\Delta} < \bar{\Delta} < \widetilde{\Delta}$,

$$f(z^k) - f(\widehat{x}) = ared(z^k, \widehat{\Delta}) \geq \eta \widetilde{c} \widehat{\Delta} = \eta \widetilde{c} \Omega(\sqrt{H_k}). \tag{30}$$

Thus, for both cases, we have that $f(z^k) - f(\widehat{x}) = \Omega(\sqrt{H_k})$. Then the step $\widehat{d}$ satisfies the conditions in the Lemma.

To finish the proof, we must show that for large $k \in \mathscr{K}$, $f(\widehat{x}) < \tilde{f}$ which implies $\widehat{x} \notin \overline{\mathscr{F}}_k$ and thus $\widehat{x} = x^{k+1}$. From (30) there is a positive constant M such that

$$f(z^k) - f(\widehat{x}) \geq M\sqrt{H_k}$$

and

$$f(\widehat{x}) \leq f(z^k) - M\sqrt{H_k}.$$

From (19) there is a positive constant N such that

$$f(z^k) \leq f(x^k) + Nh(x^k).$$

Then, combining the last two inequalities we have that

$$f(\widehat{x}) \leq f(x^k) + Nh(x^k) - M\sqrt{H_k} \leq f(x^k) + Nh(x^k) - M\sqrt{h(x^k)} =$$

$$f(x^k) - \sqrt{h(x^k)}\left(M - N\sqrt{h(x^k)}\right)$$

and, for large $k \in \mathscr{K}$ such that $M - N\sqrt{h(x^k)} > \alpha\sqrt{h(x^k)}$, which means that $\sqrt{h(x^k)} < \frac{M}{N+\alpha}$, we have that $f(\widehat{x}) < f(x^k) - \alpha h(x^k) = \widetilde{f}$, completing the proof. □

4 Convergence Results

In this Section, based on conditions (C1), (C2) and considering the general hypotheses we will show the global convergence of DFF to a quasi-stationary point.

As it was done in [15], it can be shown that (C1) and (C2) imply the following condition.

(C3) Given a feasible non quasi-stationary point $\bar{x} \in X$, there exists a neighborhood V of $\bar{x}$ such that for any iterate $x^k \in V$,

$$f(x^k) - f(x^{k+1}) = \Omega(\sqrt{H_k}) \tag{31}$$

where H_k is the filter slack at x^k defined in (9).

The difference between the conditions (C2)-(C3) and the analogous in [15] is that here they are defined in neighborhood of a non quasi-stationary point while the other are in a neighborhood of a non stationary point.

Lemma 6 *(C1) and (C2) imply (C3).*

Proof. Let $\bar{x}$ be a feasible non quasi-stationary point and let V_1 be the neighborhood defined by (C2). Since $\|z^k - x^k\| = O(h(x^k))$ and $\bar{x}$ is a feasible point there exists a neighborhood $V_2 \subset V_1$ of $\bar{x}$ such that for $x^k \in V_2$, $z^k \in V_1$. Consider an iterate $x^k \in V_2$. By (19) there is a positive constant N such that $|f(z^k) - f(x^k)| \leq Nh(x^k)$ and $f(x^k) - f(z^k) \geq -Nh(x^k)$. By (C2) there is a positive constant M such that $f(z^k) - f(x^{k+1}) \geq M\sqrt{H_k}$. Then, considering that $h(x^k) \leq H_k$, we obtain

$$f(x^k) - f(x^{k+1}) = f(x^k) - f(z^k) + f(z^k) - f(x^{k+1}) \geq M\sqrt{H_k} - Nh(x^k) =$$

$$M\sqrt{H_k} - N\sqrt{h(x^k)}\sqrt{h(x^k)} \geq M\sqrt{H_k} - N\sqrt{H_k}\sqrt{h(x^k)}\ .$$

Thus,

$$f(x^k) - f(x^{k+1}) \geq \left(M - N\sqrt{h(x^k)}\right)\sqrt{H_k}.$$

By continuity of h at the feasible point $\bar{x}$, there exists a neighborhood $V \subset V_2$ such that, for any $x \in V$, $\sqrt{h(x)} \leq 0.5\frac{M}{N}$. Therefore, for any iterate $x^k \in V$, $f(x^k) - f(x^{k+1}) \geq 0.5M\sqrt{H_k}$, completing the proof. □

The following lemmas are adaptations of Lemma 2.5 and Lemma 2.6 in [15] for the definition of quasi-stationary point for the derivative-free case. Such results are obtained considering the validity of the (C3) condition. We state them here for completeness.

Lemma 7 *Let $\bar{x} \in X$ be a non quasi-stationary limit point. Then there exist $\bar{k} \in \mathbb{N}$ and a neighborhood V of $\bar{x}$ such that whenever $k > \bar{k}$ and $x^k \in V$, the iteration k is an f-iteration.*

Lemma 8 *Suppose that $\{x^k\}_{k\in\mathbb{N}}$ has no quasi-stationary accumulation point. Then for k sufficiently large, all iterations are f-iterations.*

Finally we can obtain the following main theorem. The proof of this theorem follows straightforward from [15].

Theorem 1 *The sequence $\{x^k\}_{k\in\mathbb{N}}$ has a quasi-stationary accumulation point.*

4.1 Convergence to a Karush-Khun-Tucker point

From the previous section we know that the sequence $\{x^k\}_{k\in\mathbb{N}}$ generated by the DFF algorithm has a quasi-stationary limit point $\bar{x}$. Then there exists $\mathscr{K} \subset \mathbb{N}$ such that $\lim\limits_{k\in\mathscr{K}} x^k = \bar{x}$. Furthermore, by (C1), we have that $\lim\limits_{k\in\mathscr{K}} z^k = \bar{x}$ and consequently

$$\lim_{k\in\mathscr{K}} \|P_{L(z^k)}(z^k - \nabla_s f(z^k)) - z^k\| = 0. \tag{32}$$

In this section we will prove that, using the Linear Independence constraint qualification (LICQ) [3], $\bar{x}$ is a Karush-Kuhn-Tucker (KKT) point of (1).

The following Lemma shows that (32) still holds when we replace $\nabla_s f(z^k)$ by $\nabla f(z^k)$ but maintaining the projection onto $L(z^k)$.

Lemma 9 *Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by the DFF algorithm. Then there exists $\mathscr{K} \subset \mathbb{N}$ such that*

$$\lim_{k\in\mathscr{K}} \|P_{L(z^k)}(z^k - \nabla f(z^k)) - z^k\| = 0. \tag{33}$$

Proof. From condition (H4),

$$\|\nabla f(z^k) - \nabla_s f(z^k)\| \le k_{eg}\Delta_f^k \le k_{eg}\delta_k, \tag{34}$$

where the sequence $\{\delta_k\}$ tends to zero. Then considering

$$\|z^k - P_{L(z^k)}(z^k - \nabla f(z^k))\| = \|z^k - P_{L(z^k)}(z^k - \nabla f(z^k) - \nabla_s f(z^k) + \nabla_s f(z^k))\| \tag{35}$$

and using (12) we have that

$$\|z^k - P_{L(z^k)}(z^k - \nabla f(z^k))\| \le \|z^k - P_{L(z^k)}(z^k - \nabla_s f(z^k))\| + \|\nabla f(z^k) - \nabla_s f(z^k)\|.$$

Therefore, using (32) and (34) and taking limit when k goes to infinite, $k \in \mathscr{K}$, we have (33) as we wanted to prove. □

The main difference between the condition (7) and the condition (32) is that in the last one just estimations of the true derivatives are used.

In [15, Lemma 1.1] the authors prove that condition (7), together with the Mangasarian-Fromovitz constraint qualification [3], is equivalent to the KKT conditions.

We are able to prove that if a quasi-stationary point of the sequence generated by the algorithm verifies the Linear Independence constraint qualification then this point is a KKT point of the problem (1).

Theorem 2 *Let $\{x^k\}_{k\in\mathbb{N}}$ be a sequence generated by the DFF algorithm and $\bar{x}$ a quasi-stationary accumulation point of $\{x^k\}$ that satisfies the Linear Independence constraint qualification. Then $\bar{x}$ is a KKT point of (1).*

Proof. Since $\bar{x}$ is a quasi-stationary accumulation point of $\{x^k\}$, then there exists $\mathscr{K} \subset \mathbb{N}$ such that $\lim\limits_{k\in\mathscr{K}} x^k = \bar{x}$.

Let $\tilde{z}^k = P_{L(z^k)}(z^k - \nabla f(z^k))$, then by definition $\tilde{z}^k$ is the solution of the problem

$$\begin{aligned} &min\ \|z - (z^k - \nabla f(z^k))\|^2 \\ &\text{subject to}\quad A_k(z - z^k) = 0. \end{aligned} \tag{36}$$

Since $\bar{x}$ is a quasi-stationary accumulation point and using the previous lemma we have that

$$\lim_{k\in\mathscr{K}} (\tilde{z}^k - z^k) = 0.$$

Since the feasible set of (36) is defined by linear constraints we know that there exists $\overline{\mu}^k \in \mathbb{R}^m$ such that

$$\begin{aligned} -(\tilde{z}^k - (z^k - \nabla f(z^k))) &= A_k^T \overline{\mu}^k \\ A_k(\tilde{z}^k - z^k) &= 0. \end{aligned}$$

Then

$$z^k - \tilde{z}^k = \nabla f(z^k) + \sum_{i=1}^{m} \overline{\mu}_i^k a_i^k$$

where a_i^k denotes the $i-th$ column of A_k^T. By Carathéodory's theorem (see for example [3], page 689), for each $k \in \mathscr{K}$ there exist $I_k \subset \{1,\dots,m\}$ and $\{\mu^k\} \subset \mathbb{R}^m$ such that

$$z^k - \tilde{z}^k = \nabla f(z^k) + \sum_{i\in I_k} \mu_i^k a_i^k$$

where the set $\{a_i^k\}_{i\in I_k}$ is linearly independent.

Since the number of possible sets I_k is finite, then there exists $\mathscr{K}_1 \subset \mathscr{K}$ such that for all $k \in \mathscr{K}_1$,

$$I_k = I \subset \{1,\ldots,m\}$$

and

$$z^k - \tilde{z}^k = \nabla f(z^k) + \sum_{i\in I} \mu_i^k a_i^k \tag{37}$$

where the set $\{a_i^k\}_{i\in I}$ is linearly independent.

If $\{\mu^k\}$ is not bounded, let $M_k = \|\mu^k\|_\infty$. Then $\lim\limits_{k\in\mathscr{K}_1} M_k = \infty$ and we may take an appropriate subsequence such that $\lim\limits_{k\in\mathscr{K}_2} \dfrac{\mu^k}{M_k} = \mu \neq 0$, where $\mathscr{K}_2 \subset \mathscr{K}_1$. Then

$$\frac{z^k - \tilde{z}^k}{M_k} = \frac{\nabla f(z^k)}{M_k} + \sum_{i\in I} \frac{\mu_i^k}{M_k} a_i^k. \tag{38}$$

Thus using (H4) and taking limit in (38) when k goes to infinite, $k \in \mathscr{K}_2$, we obtain that

$$\sum_{i\in I} \mu_i \nabla c_i(\bar{x}) = 0$$

which contradicts the Linear Independence constraint qualification. So $\{\mu^k\}$ is bounded and there exists $\mathscr{K}_3 \subset \mathscr{K}_1$ such that $\lim\limits_{k\in\mathscr{K}_3} \mu^k = \mu$. Then using (H4) and taking limit in (37) when k goes to infinite, $k \in \mathscr{K}_3$, we obtain that

$$\nabla f(\bar{x}) + \sum_{i\in I} \mu_i \nabla c_i(\bar{x}) = 0.$$

Hence, $\bar{x}$ is a KKT point of (1). □

5 Numerical Experiments

In this section we present some preliminary computational results obtained with a Fortran 77 implementation of the DFF algorithm. These experiments were run on a personal computer with INTEL(R) Core (TM) 2 Duo CPU E8400 at 3.00 GHz and 3.23 GB of RAM.

As it is usual in derivative-free optimization articles we are interested in the number of function evaluations needed for satisfying the stopping criteria.

5.1 Details on the implementation of the DFF algorithm

We have considered two versions of DFF: DFF1 and DFF2. The only difference between them is the form to compute the matrix A_k. In DFF1 it is computed by simplex derivatives as was described in Algorithm 1 and used in the theoretical results. In DFF2, once z^k is computed in the restoration phase, we consider a new Broyden matrix by updating the last one computed in that process, which is used as the matrix A_k.

In our experiments the parameters used in DFF1 and DFF2 are $\alpha = 0.1$, $\beta = 100$, $\varepsilon_f = 10^{-6}$ and $\varepsilon_I = 10^{-6}$.

In this implementation we declare convergence, if breakdown does not occur at the restoration phase, when $h(x^k) \leq \varepsilon_f$, $\max\{\Delta_f^k, \Delta_c^k\} \leq \varepsilon_I$ and $\|d_c(z^k)\| \leq \varepsilon_f$.

In the implementation of the optimization phase we use the subroutine DLSVRR of the IMSL Fortran Numerical Libraries, which is based on the LINPACK routine SSVDC [10], for computing the singular value decomposition (USV) of the matrix A_k to obtain the projection of $z^k - \nabla_s f(z^k)$ onto $L(z^k)$.

Step 3 of DFF requires the calculation of the simplex gradients of c_j, for $j = 1, \ldots m$, which requires to select a set of interpolation points. In the first iteration we construct the set $Y_c^0 = \{z^0, y_c^1, \ldots, y_c^n\}$ for obtaining the models $m_{c_j}(x) = c_j(z^0) + \nabla_s c_j(z^0)^T(x - z^0)$, $j = 1, \ldots, m$, generating the matrix A_0, as an approximation of $J_c(z^0)$. We consider $y_c^i - z^0 = \rho_0\, e_i$ and the corresponding values $c_j(y_c^i)$, for $i = 1, \ldots, n$ and $j = 1, \ldots, m$, $\rho_0 < \beta \max\{\delta_0, h(x^0)\}$.

Also, it requires to compute the model $m_f(x) = f(z^k) + \nabla_s f(z^k)^T(x - z^k)$. In the first iteration, we used the vectors of the matrix V of the decomposition USV of A_0 to obtain the model $m_f(x) = f(z^0) + \nabla_s f(z^0)^T(x - z^0)$, considering the set $Y_f^0 = \{z^0, y_f^1, \ldots, y_f^n\}$, where $y_f^i = z^0 + \rho_0\, v_i$ and $f(y_f^i)$, for $i = 1, \ldots, n$.

In the following iterations Y_c^k and Y_f^k are updated, adding the new z^k as the center of them and eliminating a point y_t, the farthest from the center, trying to maintain the independence of directions. In this preliminary implementation, in some iterations the interpolation sets are newly constructed, while in others they are updated from the previous ones. The construction takes place in the first iteration and whenever it is not possible to preserve the independence of the directions easily. To check the independence of the directions we use a similar algorithm to the one proposed in [16].

The parameters used in BCDF-QNB are the same used in [11].

Finally, the parameters used in Algorithm 3 are the following: $\eta = 0.1$, $\Delta_{min} = 0.5$ and $tol = 10^{-16}$.

5.2 Test Problems

We have used a set of nonlinear programming problems defined in Hock and Schittkowski [18]. Also, we have considered one problem which was used firstly in [15] and in our previous paper [11] where we introduced the basic ideas of the actual algorithm. The selected problems from [18] are those that have equality constraints. Also, we have considered some problems from [18] with inequality constraints. In these problems the inequality constraints have been replaced by equality constraints since they are active at the solution.

In Table 1 we show the data of the problems. The number of variables ranges from 2 to 10 and the number of equality constraints from 1 to 4. Initial points were the same as in the cited references.

5.3 Numerical Results

In Table 2 we show the results obtained taking into account the number of iterations (Iter), the number of objective function evaluations (ObjEval), the number of constraints evaluations (ConstEval), the final value $f(x^{end})$ and the final value of the infeasibility $h(x^{end})$.

We can notice that the DFF1 version has done fewer iterations than the DFF2 version in 70 % of the problems. We believe that this behavior is due to the fact that DFF1 uses a better approximation of $J_c(z^k)$ in many iterations, and as consequence the initial updated matrix in

Table 1 Data of the problems

Problem	n	m	Problem	n	m	Problem	n	m
HS 6 of [18]	2	1	HS 39 of [18]	4	2	HS 60 of [18]	3	1
HS 7 of [18]	2	1	HS 40 of [18]	4	3	HS 61 of [18]	3	2
HS 8 of [18]	2	2	HS 42 of [18]	4	2	HS 63 of [18]	3	2
HS 9 of [18]	2	1	HS 43 of [18]	4	3	HS 77 of [18]	5	2
HS 14 of [18]	2	2	HS 46 of [18]	5	2	HS 78 of [18]	5	3
HS 22 of [18]	2	2	HS 47 of [18]	5	3	HS 79 of [18]	5	3
HS 26 of [18]	3	1	HS 48 of [18]	5	2	HS 80 of [18]	5	3
HS 27 of [18]	3	1	HS 52 of [18]	5	3	HS 81 of [18]	5	3
HS 29 of [18]	3	1	HS 53 of [18]	5	3	HS 111 of [18]	10	3
HS 35 of [18]	3	1	HS 56 of [18]	7	4	Example of [15]	2	1

Table 2 Results of test problems

Prob	Iter		ObjEval		ConstEval		$f(x^{end})$		$h(x^{end})$	
	DFF1	DFF2	DFF1	DFF2	DFF1	DFF2	DFF1	DFF2	DFF1	DFF2
HS 6	24	49	76	151	103	103	3.050E-05	3.023E-05	8.644E-10	2.383E-11
HS 7	10	10	33	33	46	24	-1.732E00	-1.732E00	2.620E-13	9.645E-11
HS 8	3	7	5	9	18	12	-1.000E00	-1.000E00	2.764E-12	3.157E-07
HS 9	27	49	58	101	36	53	-5.000E-01	-5.000E-01	5.329E-15	3.695E-09
HS 14	3	5	5	7	16	9	1.393E00	1.393E00	2.428E-10	6.547E-09
HS 22	3	4	5	6	15	8	1.000E00	1.000E00	2.085E-09	5.116E-07
HS 26	15	18	74	107	93	72	8.787E-07	5.367E-08	1.858E-12	4.981E-11
HS 27	6	45	81	355	270	267	4.001E-02	4.005E-02	2.254E-11	1.113E-11
HS 29	19	24	80	148	106	104	-2.263E01	-2.262E01	1.040E-11	9.216E-08
HS 35	41	45	210	186	255	97	1.111E-01	1.111E-01	9.159E-09	5.271E-10
HS 39	31	36	127	156	193	202	-1.000E00	-9.999E-01	1.036E-07	1.644E-08
HS 40	21	14	68	87	138	76	-2.500E-01	-2.500E-01	9.352E-11	1.522E-08
HS 42	18	45	123	269	165	181	1.386E01	1.386E01	9.108E-13	3.686E-14
HS 43	24	32	97	156	181	187	-4.400E01	-4.400E01	1.882E-08	1.740E-07
HS 46	31	37	182	214	249	105	5.774E-05	5.265E-05	1.323E-08	4.828E-09
HS 47	30	40	129	182	222	105	1.461E-05	2.582E-05	1.852E-09	1.824E-07
HS 48	57	62	249	317	165	133	7.521E-09	1.150E-09	9.108E-09	7.326E-09
HS 52	41	41	289	286	230	207	5.327E00	5.327E00	1.959E-08	9.141E-09
HS 53	19	19	87	81	87	46	4.093E00	4.093E00	8.408E-09	8.067E-09
HS 56	58	79	364	437	685	207	-3.456E00	-3.346E00	8.545E-07	1.217E-05(*)
HS 60	11	18	67	105	85	70	3.257E-02	3.257E-02	4.679E-11	2.839E-08
HS 61	16	18	67	94	115	82	-1.436E02	-1.436E02	4.715E-09	2.985E-10
HS 63	12	30	43	93	78	67	9.617E02	9.617E02	3.141E-10	1.806E-10
HS 77	25	26	133	270	190	198	2.415E-01	2.415E-01	1.021E-11	4.608E-07
HS 78	5	30	27	167	50	110	-2.919E00	-2.919E00	5.694E-09	1.824E-08
HS 79	8	10	41	51	73	34	7.878E-02	7.878E-02	6.064E-12	1.570E-07
HS 80	11	10	48	44	89	27	5.395E-02	5.396E-02	7.441E-09	1.282E-07
HS 81	11	11	48	48	89	29	5.395E-02	5.395E-02	1.215E-08	1.835E-07
HS 111	66	101	595	1084	805	397	-4.776E01	-4.764E01	3.849E-07	8.760E-07
Ex. of [15]	11	20	46	64	70	45	-2.210E00	-2.211E00	1.278E-09	1.783E-09

(*): the final solution does not reach the enough decrease of the infeasibility measure.

the restoration phase is better. When we consider $h(x^{end})$ as a measure of the performance of the algorithms we can see that DFF1 outperforms DFF2 in 70 % of the problems.

From the results of test problems we can conclude that the restoration algorithm was successful in almost all iterations of all the problems. The only exception was the problem HS 56 for DFF2.

For algorithmic comparison we use *performance profile* described in [9] and *data profile* for derivative-free optimization presented in [30].

The performance profile of a solver s is defined as the fraction of problems where the performance ratio is at most α, that is, $\rho_s(\alpha) = \frac{1}{|\mathscr{P}|}\text{size}\{p \in \mathscr{P} : r_{p,s} \leq \alpha\}$, where $r_{p,s} =$

$\frac{t_{p,s}}{\{\min t_{p,s}: s\in\mathscr{S}\}}$, $t_{p,s}$ is the number of function evaluations required to satisfy the convergence test, $\mathscr{P}$ is the set of problems and $|\mathscr{P}|$ denotes the cardinality of $\mathscr{P}$.

We are also interested in the percentage of problems that can be solved, according to the convergence test mentioned in subsection 5.1, by a solver s with a particular number of function evaluations. The percentage of problems that can be solved with α function evaluations is computed by $d_s(\alpha) = \frac{1}{|\mathscr{P}|}\text{size}\{p \in \mathscr{P} : t_{p,s} \leq \alpha\}$.

As it was mentioned in [30], the definition of d_s is independent of the number of variables of the problem $p \in \mathscr{P}$. However we know that the number of function evaluations grows when the number of variables grows. We thus consider the data profile of a solver s by $d_s(\alpha) = \frac{1}{|\mathscr{P}|}\text{size}\{p \in \mathscr{P} : \frac{t_{p,s}}{n+1} \leq \alpha\}$, where n is the number of variables in $p \in \mathscr{P}$. The value of $d_s(\alpha)$ can be interpreted as the percentage of problems that can be solved with the equivalent of α simplex gradient estimates, considering that $n+1$ is the number of evaluations needed to compute a one-sided finite-difference estimate of the gradient [30].

We analyze separately the number of objective function evaluations (ObjEval) and the number of constraints evaluations (ConstEval).

In the following figures we compare DFF1 and DFF2 using the number of objective function evaluations as a measure of the performance.

In the performance profile of Figure 1, we can notice that DFF1 expended less objective function evaluations in more than 80% of the problems, while DFF2 expended less objective function evaluations in approximately 20% of the problems. The performance difference between DFF1 and DFF2 is approximately 20% when the performance ratio is 2.

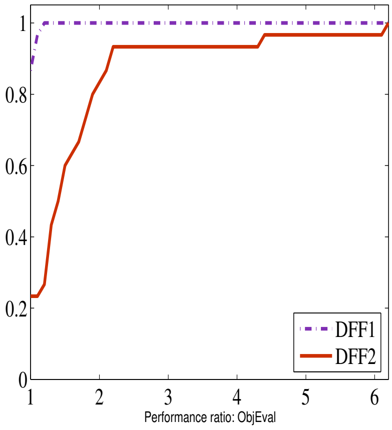


Fig. 1 Performance profile: Objective function evaluations

The data profile of Figure 2 (a) shows that DFF1 solves the largest percentage of problems for all sizes of the number of objective function evaluations. We can observe that DFF1 solves 80% of problems with 200 evaluations while DFF2 solves approximately 70%. The biggest difference is 30% and it happens when the number of function evaluations is approximately 180. We believe that this behavior is due to the fact that DFF1 uses a better approximation of $J_c(z^k)$ in many iterations as well as it makes fewer iterations.

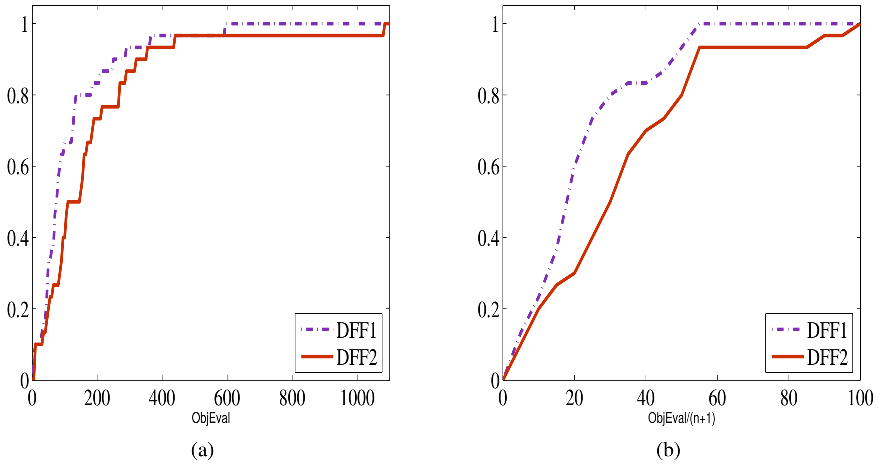


Fig. 2 Data profiles for the comparison between DFF1 and DFF2: Objective funtion evaluations

Figure 2 (b) shows that DFF1 solves the largest percentage of problems for all sizes of the number of simplex gradient estimates (ObjEval/$(n+1)$). With 60 evaluations DFF1 solves 100% of the problems while DFF2 requires 100 evaluations to solve all of them. The biggest difference between DFF1 and DFF2 happens when the number of function evaluations is approximately 30% and in this case DFF1 solves 80% of the problems while DFF2 solves approximately 50% of them.

In the following figures we compare DFF1 and DFF2 using the number of constraints evaluations as a measure of the performance.

In the performance profile of Figure 3 we can notice that DFF2 expended less constraints function evaluations in approximately 80% of the problems while DFF1 expended less constraints function evaluations in more than 20%.

In Figure 4 (a) the data profile shows that DFF2 solves the largest percentage of problems for all sizes of the number of constraints evaluations. We believe that this result is associated to the fact that DFF2 does not require new constraints evaluations to define the matrix A_k because it updates the last matrix used in the restoration phase. With 400 evaluations DFF2 solves all the problems, while DFF1 needs 800 evaluations to solve all of them.

Figure 4 (b) shows that DFF2 solves the largest percentage of problems for all sizes of the number of simplex gradient estimates (ConstEval/$(n+1)$). With 70 evaluations DFF2 solves almost 100 % of the problems, while DFF1 solves approximately 90% of the problems. The biggest difference between DFF1 and DFF2 happens when the number of constraints evaluations is 20% and in this case DFF2 solves 60% of the problems while DFF1 solves approximately 40% of them.

Taking into account the performance and data profiles we believe that better results can be obtained developing another alternative that combines DFF1 and DFF2 implementations. That could be made considering the DFF2 implementation, computing A_k by simplex gradients after a fix number of iterations. In addition, in the application of BCDF-QNB in the

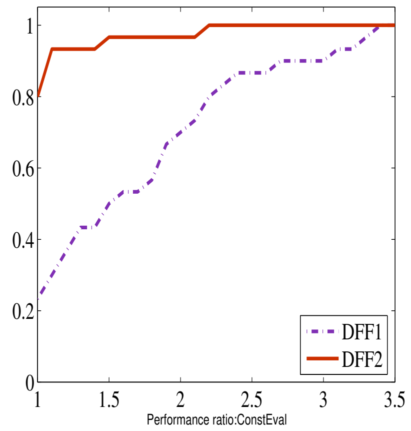
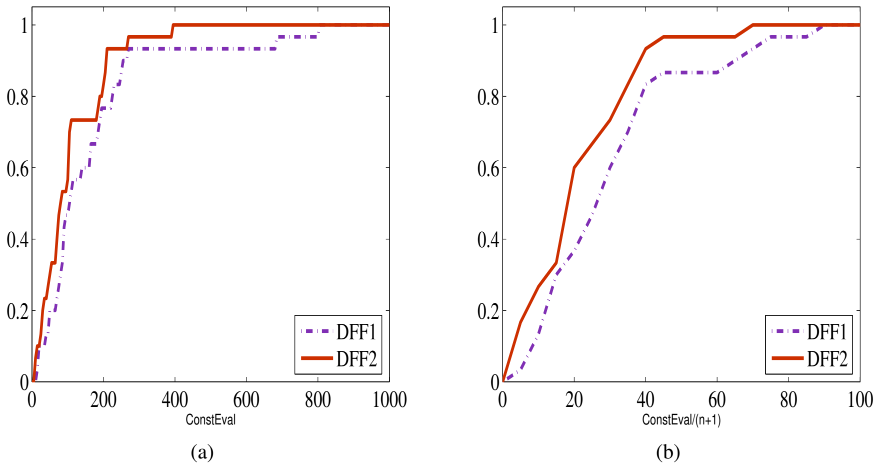


Fig. 3 Performance profile: Constraints evaluations



(a)

(b)

Fig. 4 Data profiles for the comparison between DFF1 and DFF2: Constraints evaluations

restoration phase, we could replace the use of finite differences to compute B_k by the use of simplex gradients. That will be a subject of future study.

6 Conclusions

We have presented an inexact restoration filter algorithm for equality constrained nonlinear programming without using derivatives. The main contribution of the paper is to extend the theory of a filter based optimization method to the derivative-free context, but future research about numerical behavior of the algorithm is still necessary to understand if there exists a class of problems that would be better solved with the DFF algorithm than with other benchmark DF algorithm.

From the theoretical point of view, under suitable conditions, we were able to prove global convergence to quasi-stationary points. Furthermore we have shown that if a quasi-stationary accumulation point satisfies the Linear Independence constraint qualification then this point is a KKT point of (1).

From the practical point of view, two versions of the proposed algorithm were implemented and tested considering a set of small scale problems. The main difference between the two versions is the way in which an approximation of the true Jacobian $J_c(z^k)$ is computed. Two main aspects can be taken into account from the numerical experiments:

1. They suggest plausible the use of Quasi Newton for computing the Jacobian approximations and this will be one of the subject of forthcoming research.
2. The implemented algorithms behave as expected, however, it will be desirable to test the execution of the algorithm with a more challenging set of problems. Also, we would like to compare the performance of the tested algorithms with other derivative-free algorithms defined for solving the same problem.

As the method proposed is the type of inexact restoration, different alternatives can be studied in order to solve the two phases. In particular, to solve the optimality phase we would like to define a derivative-free algorithm based on a quadratic model, instead of a linear one. In this case the use of quadratics models must be consistent with the theory, especially with the condition (C2), in order to preserve the convergence.

Acknowledgements The authors are indebted to the two anonymous referees whose comments helped a lot to improve the quality of the paper.

References

1. Alexandrov N., Hussaini M.Y., *Multidisciplinary Design Optimization: state of the art*, SIAM Proceedings Series, SIAM, Philadelphia (1997).
2. Arouxét M.B.,Echebest N., Pilotta E.A., *Active-set strategy in Powell's method for optimization without derivatives*, Computational and Applied Mathematics, 30 (1), 171-196 (2011).
3. Bertsekas D.P., *Nonlinear Programming, 2nd Edition*. Athena Scientific, Belmont, Massachusetts (1999).
4. Bueno L.F., Friedlander A., Martínez J.M., Sobral F.N.C., *Inexact restoration method for derivative-free optimization with smooth constraints*, SIAM Journal on Optimization, 23 (2), 1189-1213 (2013).
5. Conn A.R., Scheinberg K., Toint Ph. L., *Recent progress in unconstrained nonlinear optimization without derivatives*, Mathematical Programming, 79 (3), 397-414 (1997).
6. Conn A., Scheinberg K., Vicente L.N., *Introduction to derivative-free optimization*. SIAM Book Series on Optimization, Philadelphia (2009).
7. Custodio A.L., Vicente L.N., *Using sampling and simplex derivatives in pattern search methods*, SIAM Journal on Optimization, 18 (2), 537-555 (2007).
8. Diniz-Ehrhardt M.A., Martínez J.M., Pedroso L.G., *Derivative-free methods for nonlinear programming with general lower-level constraints*, Computational and Applied Mathematics, 30 (1), 19-52 (2011).
9. Dolan E., Moré J., *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91, 201-213 (2002).
10. Dongarra J.J., Bunch J.R., Moler C.B., Stewart G.W., *LINPACK Users'Guide*, SIAM, Philadelphia (1979).
11. Echebest N., Schuverdt M.L., Vignau R.P., *A derivative-free method for solving box-constrained underdetermined nonlinear systems of equations*, Applied Mathematics and Computation, 219 (6), 3198-3208 (2012).

12. Fletcher R., Gould N.I.M., Leyffer S., Toint Ph.L., Wächter A., *Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming*, SIAM Journal on Optimization, 13, 635-659 (2002).
13. Fletcher R., Leyffer S., *Nonlinear programming without a penalty function*, Mathematical Programming, 91 (2), 239-269 (2002).
14. Gomes-Ruggiero M.A., Martínez J.M., Santos S.A., *Spectral projected gradient method with inexact restoration for minimization with nonconvex constraints*, SIAM Journal on Scientific Computing, 31, 1628-1652 (2009).
15. Gonzaga C.C., Karas E.W., Vanti M., *A globally convergent filter method for nonlinear programming*, SIAM Journal on Optimization, 14 (3), 646-669 (2004).
16. Gratton S., Toint Ph.L., A. Tröltzsch A., *An active-set trust region method for bound-constrained nonlinear optimization without derivatives*, Optimization Methods and Software, 26 (4-5), pp. 875-896 (2011).
17. Grippo L., Lampariello F., Lucidi S., *A nonmonotone line search technique for Newton's method*, SIAM Journal on Numerical Analysis, 23, 707-716 (1986).
18. Hock W., Schittkowski K., *Test Examples for Nonlinear Programming Codes*. Springer Series Lectures Notes in Economics Mathematical Systems (1981).
19. Kolda T.G., Lewis R.M., Torczon V., *Optimization by direct search: new perspectives on some classical and modern methods*, SIAM Review, 45, 85-482 (2003).
20. Kolda T.G., Lewis R.M, Torczon V., *Stationarity results for generating set search for linearly constrained optimization*, SIAM Journal on Optimization, 17 (4), 943-968 (2006).
21. Lewis R.M., Torczon V., *Pattern search algorithms for bound constrained minimization*, SIAM Journal on Optimization, 9 (4), 1082-1099 (1999).
22. Lewis R.M., Torczon V., *Pattern search algorithms for linearly constrained minimization*, SIAM Journal on Optimization, 10 (3), 917-941 (2000).
23. 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).
24. Lewis R.M., Torczon V., *A direct search approach to nonlinear programming problems using an Augmented Lagrangian method with explicit treatment of linear constraints*, Technical Report WMCS-2010-01, College of William & Mary, Department of Computer Sciences (2010).
25. Li D.H., Fukushima M., *A derivative-free line search and global convergence of Broyden-like method for nonlinear equations*, Optimization Methods and Software, 13, 181-201 (2000).
26. Martínez J.M., *Inexact-Restoration method with Lagrangian tangent decrease and a new merit function for nonlinear programming*, Journal of Optimization Theory and Applications, 111 (1), 39-58 (2001).
27. Martínez J.M., Pilotta E.A., *Inexact Restoration algorithm for constrained optimization*, Journal of Optimization Theory and Applications, 104 (1), 135-163 (2000).
28. Martínez J.M., Pilotta E.A., *Inexact Restoration methods for nonlinear programming: Advances and perspectives*. In: Qi, Teo, Yang (eds.) Optimization and Control with Applications, 271-292. Springer, Berlin (2005).
29. Martínez J.M., Svaiter B.F., *A practical optimality condition without constraint qualifications for nonlinear programming*, Journal of Optimization Theory and Applications, 118 (1), 117-133 (2003).
30. Moré J.J., Wild S.M., *Benchmarking derivative-free optimization algorithms*, SIAM Journal on Optimization, 20 (1), 172-191 (2009).
31. Powell M.J.D., *The NEWUOA software for unconstrained optimization without derivatives*. In: Di Pillo, G., Roma, M. (eds) Large-Scale Nonlinear Optimization, 255-297. Springer, New York (2006).
32. Powell M.J.D., *The BOBYQA algorithm for bound constrained optimization without derivatives*, Cambridge NA Report NA2009/06, University of Cambridge, Cambridge (2009).