Comput Optim Appl (2015) 60:1–33
DOI 10.1007/s10589-014-9665-9

A class of derivative-free nonmonotone optimization algorithms employing coordinate rotations and gradient approximations

L. Grippo · F. Rinaldi

Received: 20 February 2013 / Published online: 7 June 2014

Abstract In this paper we study a class of derivative-free unconstrained minimization algorithms employing nonmonotone inexact linesearch techniques along a set of suitable search directions. In particular, we define globally convergent nonmonotone versions of some well-known derivative-free methods and we propose a new linesearch-based nonmonotone algorithm, where search directions are constructed by combining coordinate rotations with simplex gradients. Through extensive numerical experimentation, we show that the proposed algorithm is highly competitive in comparison with some of the most efficient direct search methods and model based methods on a large set of test problems.

Keywords Derivative-free optimization · Nonmonotone linesearch techniques · Coordinate rotation · Rosenbrock method · Hooke–Jeeves method · Simplex gradient

1 Introduction

We consider unconstrained minimization problems of the form $\min_{x \in R^n} f(x)$, where (in the theoretical analysis) the function $f : R^n \to R$ is assumed to be continuously differentiable on R^n. However, we suppose that the derivatives of f are not available and that cannot be easily approximated by finite difference methods. As remarked in [4], this situation frequently arises when f must be evaluated through black-

L. Grippo
Dipartimento di Ingegneria Informatica Automatica e Gestionale,
"Sapienza" University of Rome, Rome, Italy
e-mail: grippo@dis.uniroma1.it

F. Rinaldi (✉)
Dipartimento di Matematica, University of Padua, Padua, Italy
e-mail: rinaldi@math.unipd.it

box simulation packages, typically proprietary, and each function evaluation may be costly and noisy. Thus the development of derivative-free optimization techniques is currently an area of increasing interest, both from a theoretical and a practical point of view.

A recent comprehensive study on methods for derivative-free optimization is the book cited above [4], where several approaches, such as pattern search methods, model-based methods, linesearch methods employing approximate gradients are described and analyzed. It would seem that there is still the need of improving the current approaches and of developing new ideas, especially when the problem dimensions are relatively large or the function f is noisy.

The objective of the present paper is actually that of contributing to this active field, by proposing new efficient derivative-free algorithms based on nonmonotone inexact linesearches along a set of search directions satisfying appropriate conditions. The derivative-free inexact linesearches used here are based on the techniques introduced in [7] and employed in e.g. [18,25,26], for defining globally convergent algorithms.

We note that the definition of appropriate 'inexact' algorithms can be important, in the general case, even from a theoretical point of view. In fact, for a wide class of derivative-free algorithms, 'exact' linesearches (at least in the sense that a stationary point along the search direction is approximated with good accuracy) may not guarantee global convergence, unless rather restrictive assumptions are imposed on f.

We consider also modifications of these algorithms based on nonmonotone acceptance rules, which relax the descent requirements, while preserving the convergence properties of the monotone methods. The modified algorithms used here can be viewed as derivative-free extensions of nonmonotone linesearch algorithms employing first order derivatives (see, e.g. [14,16,17]) and have been also considered in connection with Jacobian-free techniques for solving nonlinear equations [15]. Different derivative-free inexact and nonmonotone linesearch algorithms have been proposed in [22,24], in the context of methods for the solution of nonlinear equations and in [8,10–12] with reference to optimization problems.

The introduction of nonmonotone acceptance rules can improve considerably both robustness and efficiency, especially in the case of noisy problems and in the case of objective function with contours exhibiting steep sided valleys or even non-differentiable.

On the basis of the results mentioned above, in this paper we construct a class of algorithms that combines different strategies for choosing the search directions and for performing the line searches. The general algorithm scheme, illustrated in the next section, allows us to unify the description and the convergence analysis of various new linesearch-based algorithms. In particular, first we construct new non-monotone globally convergent versions of known methods, such as the coordinate method, the Hooke–Jeeves method [19] and the Rosenbrock method [33]. Then, in the same framework, we define a new algorithm in which Rosenbrock rotation of the coordinate axes is combined with an approximate gradient constructed using previous information on function values. In particular, we use a form of (generalized) 'simplex gradient' [3,5,6,20]. Approximations of the gradient of this form have been used in various works, in order to improve the performance of derivative-free methods and error bounds have been established. In our algorithm the acceleration step along the negative simplex gradient has also the effect of defining the subsequent rotation of the coordinate axes.

The algorithms introduced here are compared on a large set of difficult test problems [13,27] with some of the best derivative-free methods currently available. The numerical results, evaluated through the performance and data profiles introduced in [28], show that the proposed technique is competitive with the other approaches.

The paper is organized as follows. In Sect. 2, we describe the general scheme of the class of algorithms considered in the sequel. In Sect. 3, we recall the basic features and the convergence properties of the linesearch algorithms, on the basis of the previous papers mentioned above. In Sect. 4, we specialize and extend some of the convergence results given for derivative-free methods [4,18,26]. In Sect. 5, we describe a new globally convergent implementation of the Hooke–Jeeves method, employing nonmonotone inexact linesearches. In Sect. 6, we describe a linesearch-based nonmonotone version of Rosenbrock's method and we prove global convergence. In Sect. 7, we describe and analyze the new nonmonotone linesearch-based algorithm that makes use of simplex gradients and of Rosenbrock rotations for defining the search directions. In Sect. 8, we report and discuss the results of our computational experimentation. Finally, Sect. 9 contains some concluding remarks and indications on future work.

2 A conceptual scheme of the algorithms

In this section, we give an informal outline of the class of methods proposed in the present paper; more precise descriptions of the specific methods will be reported in the sequel. All the algorithms we will consider generate an infinite sequence of iterations of the form

$$x_{k+1} = x_k + \alpha_k d_k, \tag{1}$$

where $d_k \in R^n$ is a search direction, $\alpha_k \in R$ is a stepsize and $x_0 \in R^n$ is a given point. We denote by

$$\mathcal{L}_0 = \{x \in R^n : f(x) \le f(x_0)\},$$

the level set of f corresponding to the initial value $f(x_0)$.

The algorithmic scheme we consider in the paper can be described as an infinite sequence of major steps, indexed by $\ell = 0, 1, \ldots$. Each major step ℓ consists of a finite number of iterations of the form (1), organized into three different phases, as indicated below.

(a) *Basic search* Starting from the current point x_k, indicated by y^0, we consider a finite set of r search directions

$$D = \{d^i \in R^n, i = 1, \ldots, r\}, \tag{2}$$

where typically $r \geq n$. For $i = 1, \ldots, r$ we use, in sequence, the search directions $d_k = d^i$ in D and we compute α_k by means of a derivative-free nonmonotone line search. Then, for each i, we generate a new point using the iteration (1) and we update k. During this phase, when needed, we can further store a set of tentative points $y^j \in R^n$ computed along the directions d^i and the corresponding function values $f(y^j)$, for $j = 0, 1, \ldots, q$. After r line searches, we obtain a new updated point x_k.

(b) *Acceleration step* Given the available information gathered during phase (a), that is y^j, $f(y^j)$ for $j = 0, 1, \ldots, q$, we determine a new search direction d_k on the basis of some local model of f. Using again a derivative-free nonmonotone line search technique along d_k we perform the iteration

$$x_{k+1} = x_k + \alpha_k d_k.$$

The attempt is that of improving substantially the results of phase (a) by computing, for instance, an approximation to the steepest descent direction or by determining a suitable pattern on the basis of the previous steps.

(c) *Rotation of the search directions* We perform a rotation (according to some given criterion) of the directions in D, thus obtaining a new set

$$\bar{D} = \left\{\bar{d}^i \in R^n, \quad i = 1, \ldots, r\right\}.$$

In the sequel, we will refer, in particular, to the Rosenbrock rotation [33] or to suitable modifications of this technique. Once that $\bar{D}$ has been determined, we set $D = \bar{D}$, we update k and a new major step can be started.

In our formulation the fundamental requirements for establishing global convergence are the conditions to be imposed on the line searches and the assumptions on the search directions in D. Thus, the basic search of step (a) is essential, while both the acceleration step and the rotation of the directions in D can be omitted. Therefore various algorithms can be derived from the scheme defined above. We mention here some of the most significant choices that will be studied in the sequel in more detail.

The simplest choice can be a (nonmonotone) linesearch-based version of the *coordinate method*. In this case the set D is defined by $D = \{e_1, \ldots, e_n\}$, where e_j is the j-th column of the identity $n \times n$ and the algorithm will consist only of the basic search step (a).

A second scheme can be defined by combining step (a) with step (b). An example could be a linesearch-based version of the *Hooke–Jeeves method*. In this case the set D is again the set of coordinate directions, no coordinate rotation is introduced and D remains constant during the iterations. The acceleration step (b) consists in defining the search direction

$$d_k = x_k - y^0$$

and in performing a (possibly nonmonotone) linesearch along d_k. A second example with a similar structure could be a nonmonotone extension of the algorithm defined in

[26], where step (b) is defined by selecting a point and a search direction on the basis of the points and the function values produced at step (a).

A different algorithm consists in executing only step (a) and (c), by performing a coordinate rotation at step (c), starting from the vector

$$\bar{d}^1 = x_k - y^0$$

and employing Rosenbrock rotations. In this way we get a (nonmonotone) linesearch-based version of the *Rosenbrock method.*

Finally, new algorithms can be constructed where the acceleration step (b) and the Rosenbrock rotation at step (c) are combined. In the scheme proposed in this paper each iteration starts with set D and the basic search (a) is executed as in the other cases. The acceleration step (b) is defined by first computing the *gradient approximation* g_k at the last point x_k of step (a) and then performing a (nonmonotone) derivative-free line search along $-g_k$. The gradient approximation is constructed by employing the data collected during the linesearches of step (a). Once that the point $x_{k+1} = x_k - \alpha_k g_k$ has been obtained, the set D is updated by employing Rosenbrock rotation. However, now the first element of $\bar{D}$ is given by

$$\bar{d}^1 = x_{k+1} - y^0$$

and hence the step along $-g_k$ has effect also on the rotation carried out within the Rosenbrock technique.

In order to perform a convergence analysis of the algorithms outlined above, we must first specify the line search algorithms employed, we must impose suitable conditions on the search directions that belong to D and then we must define exactly the different steps. This will be the object of the next sections.

3 Derivative-free line searches

We consider the derivative-free line search algorithms that will be used within the minimization method described in the preceding section, where it is assumed that, given x_k, the next point x_{k+1} is obtained through the iteration

$$x_{k+1} = x_k + \alpha_k d_k,$$

where $d_k \in R^n$ is a search direction and $\alpha_k \in R$ is a step size.

The algorithms defined in this section are essentially based on the techniques described in [15] and can be viewed as nonmonotone versions of derivative-free line search algorithms enforcing to zero the distance $\|x_{k+1} - x_k\|$. A sufficient reduction of the objective function is imposed with respect to a reference value W_k that satisfies the condition

$$f(x_k) \le W_k \le \max_{0 \le j \le \min(k,M)} [f(x_{k-j})], \tag{3}$$

for a given integer $M \geq 0$. In order to simplify notation, in the sequel we set $f_k = f(x_k)$ whenever convenient.

We consider two different types of line search, depending on the fact that the search is bidirectional, that is α_k can have any sign, or that only a nonnegative value for α_k is permitted.

The first scheme considered in the sequel is the bidirectional search. Starting from an arbitrary given initial tentative value $\alpha = \Delta_k > 0$, we search both along d_k and $-d_k$ and the stepsize α is computed in order to satisfy the condition of sufficient decrease expressed by

$$f(x_k + \alpha_k d_k) \leq W_k - \gamma \alpha_k^2 \|d_k\|^2, \quad \gamma > 0, \tag{4}$$

with a 'sufficiently large' stepsize. We note that, in this phase, when the initial tentative stepsize Δ_k or $-\Delta_k$ does not satisfy the condition (4), then at least three function values are available, and hence a safeguarded quadratic interpolation can be employed. The stepsize is reduced until either the condition of sufficient decrease is satisfied, and hence the sign of α_k is fixed, or the length of the tentative step $\alpha\|d_k\|$ becomes smaller than an adjustable bound ρ_k.

In the latter case, the value α_k determined by the algorithm is set equal to zero. This failure of the search along d_k may be due, in particular, to the fact that the directional derivative of f along d_k at x_k is near zero.

When the initial tentative stepsize Δ_k or $-\Delta_k$ satisfies the condition of sufficient decrease, then an expansion step is performed and $|\alpha|$ is increased, until suitable conditions are satisfied. Several different acceptability conditions can be combined for guaranteing that a sufficient displacement from x_k has been effected. In particular, an extension of Goldstein conditions can be based on the request that α_k satisfies both the condition of sufficient decrease and the condition

$$f(x_k + \alpha_k d_k) \geq f_k - \gamma_1 \alpha_k^2 \|d_k\|^2,$$

where $\gamma_1 > \gamma$.

Note that, in the nonmonotone case, an expansion step is not required if the initial tentative stepsize $\alpha \in \{-\Delta_k, \Delta_k\}$ satisfies the condition of sufficient decrease, but $f(x_k + \alpha d_k) > f_k$.

We remark that the expansion step is important both from a theoretical and from a practical point of view. In fact, in order to establish convergence we must impose that 'sufficiently large' stepsizes are employed and the expansion step allows us to start from an arbitrarily small value of the initial stepsize Δ_k. In the absence of the expansion step, we must choose a sufficiently large value of Δ_k, say, for instance, $\Delta_k \geq c/\|d_k\|$ for some $c > 0$. On the other hand, as $|\alpha_k|\|d_k\|$ is forced to zero, but $\|d_k\|$ can be constant in a derivative-free method (think, for instance, of the coordinate axes), it could be convenient to choose diminishing values for Δ_k, but then an *a priori* choice would not guarantee that the stepsizes are 'sufficiently large'.

A formal description of the algorithm outlined above is reported in the following scheme.

```
**Nonmonotone Derivative Free Line Search Algorithm 1 (NDFLS1)**
$d_k \in R^n, d_k \neq 0$, $W_k$ satisfying (3) and parameters:

    $0 < \theta_l < \theta_u < 1, \quad 1 < \mu_l < \mu_u, \quad \gamma_1 > \gamma > 0, \quad \rho_k > 0, \quad \Delta_k > 0.$

Step 1. Set $\alpha = \Delta_k$.
Step 2. While $f(x_k \pm \alpha d_k) > W_k - \gamma\alpha^2\|d_k\|^2$ do:
            If $\alpha\|d_k\| < \rho_k$ then
                set $\alpha_k = 0$ and terminate,
            else
                choose $\theta \in [\theta_l, \theta_u]$ and set $\alpha = \theta\alpha$.
            End if
        End while
Step 3. Let $t \in \{-1, 1\}$ be such that $f(x_k + t\alpha d_k) \le W_k - \gamma\alpha^2\|d_k\|^2$
        and set $\alpha = t\alpha$.
Step 4. If $|\alpha| < \Delta_k$, then set $\alpha_k = \alpha$ and terminate.
Step 5. Choose $\mu \in [\mu_l, \mu_u]$.
Step 6. While
            $f(x_k + \alpha d_k) < f_k - \gamma_1\alpha^2\|d_k\|^2$
              and
            $f(x_k + \mu\alpha d_k) < \min\{f(x_k + \alpha d_k), f_k - \gamma(\mu\alpha)^2\|d_k\|^2\}$
            set $\alpha = \mu\alpha$ and choose $\mu \in [\mu_l, \mu_u]$.
        End while
Step 7. Set $\alpha_k = \alpha$ and terminate.
```

In the sequel, whenever convenient, we will recall the preceding algorithm by indicating some of the inputs; in this case we will use the notation NDFLS1(d_k, Δ_k, ρ_k).

The properties of Algorithm NDFLS1 are stated in the next proposition, whose proof follows with minor modifications from the results given in Propositions 3 and 4 of [15].

Proposition 1 *Let $f : R^n \to R$ be continuously differentiable and assume that the level set $\mathcal{L}_0$ is compact.*

(i) *Algorithm* NDFLS1 *determines, in a finite number of steps, a scalar α_k such that*

$$f(x_k + \alpha_k d_k) \le W_k - \gamma\alpha_k^2\|d_k\|^2. \tag{5}$$

(ii) *Let $\{x_k\}$ be a the sequence of points in R^n and let K be an infinite index set such that*

$$x_{k+1} = x_k + \alpha_k d_k, \quad \textit{for all} \quad k \in K$$

where $d_k \in R^n$, $d_k \neq 0$ and $\alpha_k \in R$ is determined by means of Algorithm NDFLS1. Assume that $\{f(x_k)\}$ converges and that $\rho_k \to 0$ for every infinite subsequence of $\{x_k\}_K$ such that $\alpha_k = 0$. Then, we have:

$$\lim_{k\to\infty, k\in K} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} = 0.$$

□

Now we can define the linesearch algorithm where the searches are performed only for positive values of α, which can easily obtained from the scheme of NDFLS1 algorithm by simple modifications.

Nonmonotone Derivative Free Line Search Algorithm 2 (NDFLS2)

```
Data. $d_k \in R^n, d_k \neq 0$, $W_k$ satisfying (3) and parameters:
      $0 < \theta_l < \theta_u < 1, \quad 1 < \mu_l < \mu_u, \quad \gamma_1 > \gamma > 0, \quad \rho_k > 0, \quad \Delta_k > 0.$

Step 1. Set $\alpha = \Delta_k$.
Step 2. While $f(x_k + \alpha d_k) > W_k - \gamma\alpha^2\|d_k\|^2$ do:
            If $\alpha\|d_k\| < \rho_k$ then
                set $\alpha_k = 0$ and terminate,
            else
                choose $\theta \in [\theta_l, \theta_u]$ and set $\alpha = \theta\alpha$.
            End if
        End while
Step 3. If $\alpha < \Delta_k$, then set $\alpha_k = \alpha$ and terminate.
Step 4. Choose $\mu \in [\mu_l, \mu_u]$.
Step 5. While
            $f(x_k + \alpha d_k) < f_k - \gamma_1\alpha^2\|d_k\|^2$
                and
            $f(x_k + \mu\alpha d_k) < \min\{f(x_k + \alpha d_k), f_k - \gamma(\mu\alpha)^2\|d_k\|^2\}$
            set $\alpha = \mu\alpha$ and choose $\mu \in [\mu_l, \mu_u]$.
        End while
Step 6. Set $\alpha_k = \alpha$ and terminate.
```

The next proposition gives the convergence properties of the algorithm, that we may recall as NDFLS2(d_k, Δ_k, ρ_k). The proof can be established along the lines followed for proving Proposition 1.

Proposition 2 *Let $f : R^n \to R$ be continuously differentiable and assume that the level set $\mathcal{L}_0$ is compact.*

(i) *Algorithm* NDFLS2 *determines, in a finite number of steps, a scalar $\alpha_k \geq 0$ such that*

$$f(x_k + \alpha_k d_k) \leq W_k - \gamma \alpha_k^2 \|d_k\|^2. \tag{6}$$

(ii) *Let $\{x_k\}$ be a the sequence of points in R^n and let K be an infinite index set such that*

$$x_{k+1} = x_k + \alpha_k d_k, \quad \text{for all} \quad k \in K$$

where $d_k \in R^n$, $d_k \neq 0$ and $\alpha_k \in R$ is determined by means of Algorithm NDFLS2. *Assume that $\{f(x_k)\}$ converges and that $\rho_k \to 0$ for every infinite subsequence of $\{x_k\}_K$ such that $\alpha_k = 0$. Then, we have:*

$$\lim_{k\to\infty, k\in K} \frac{\nabla f(x_k)^T d_k}{\|d_k\|} \geq 0.$$

$\square$

4 Global convergence conditions

In this section we state convergence conditions that impose restrictions on the directions employed during the basic search defined at phase (a) of the scheme of Sect. 2. More specifically, we extend to nonmonotone methods some of the global convergence conditions already established for derivative-free line search-based methods [4,18,26]. In particular, we state conditions under which every limit point of the sequence generated by an algorithm of the form

$$x_{k+1} = x_k + \alpha_k d_k,$$

where $d_k \in R^n$ is a search direction and $\alpha_k \in R$ is a stepsize, is a stationary point of f. These conditions will be specialized in the sequel to the model algorithm introduced in Sect. 2 for proving global convergence.

The convergence results established in this section depend to a great extent on the lemma reported below, proved in [14], which follows essentially from the results established in [16,17]. This lemma plays a major role in the convergence analysis of nonmonotone methods, since it yields a sufficient condition for establishing the convergence of a nonmonotone sequence of function values and for proving that $\|x_{k+1} - x_k\| \to 0$.

Before stating this result we recall from [29] that a function $\sigma : R^+ \to R^+$ is called a *forcing function* if, for every sequence of numbers $t_k \in R^+$ such that $\lim_{k\to\infty} \sigma(t_k) = 0$,we have that

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

Then we can state the following lemma.

Lemma 3 *Let $\{x_k\}$ be a sequence of points such that*

$$f(x_{k+1}) \leq W_k - \sigma(\|x_{k+1} - x_k\|), \tag{7}$$

where $\sigma : R^+ \to R^+$ is a forcing function and W_k is a reference value that satisfies (3) for a given integer $M \geq 0$. Suppose that f is bounded below, and that it is Lipschitz continuous on $\mathcal{L}_0$, that is, that there exists a constant L such that

$$|f(x) - f(y)| \leq L\|x - y\|, \quad \textit{for all} \quad x, y \in \mathcal{L}_0.$$

Then:

(i) $x_k \in \mathcal{L}_0$ *for all k;*
(ii) *the sequence $\{f(x_k)\}$ is convergent;*
(iii) $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$. □

Suppose first that the sequence of search directions $\{d_k\}$ satisfies the following assumption, which requires, in essence, that ultimately a set of n uniformly linearly independent search directions is employed cyclically.

Assumption 4 There exist an integer $N > 0$ and n integer sequences $\{j_k^i\}$, $(i = 1, \ldots, n)$ such that:

a) $k \leq j_k^1 \leq j_k^2 \leq \ldots \leq j_k^n \leq k + N$ for each $k \geq 0$;
b) Every limit point of the sequence of $n \times n$ matrices

$$P_k = \left(\frac{d_{j_k^1}}{\|d_{j_k^1}\|} \frac{d_{j_k^2}}{\|d_{j_k^2}\|} \ldots \frac{d_{j_k^n}}{\|d_{j_k^n}\|} \right) \quad (k = 0, 1, \ldots),$$

where $\|d_{j_k^i}\| > 0$ for all $k \geq 0$ and $i = 1, \ldots, n$, is a non singular matrix in $R^{n\times n}$.

□

It is easily seen that the assumption given above, for a sufficiently large N, is satisfied in a scheme where the coordinate directions are employed cyclically. Other examples will be considered later.

We can now establish the following result.

Proposition 5 *Let $f : R^n \to R$ be a continuously differentiable function and assume that the level set $\mathcal{L}_0$ is compact. Let $\{x_k\}$ be the sequence of points produced by an algorithm of the form $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \neq 0$ for all k and $\alpha_k \in R$.*

Suppose that:

(a) *For every k we have*

$$f(x_{k+1}) \leq W_k - \sigma(\|x_{k+1} - x_k\|), \tag{8}$$

where $\sigma : R^+ \to R^+$ is a forcing function and W_k is a reference value that satisfies (3) for a given integer $M \geq 0$;

(b) *the sequence of search directions* $\{d_k\}$ *satisfies Assumption* 4;

(c) *in correspondence to the sequences* $\{j_k^i\}$, $(i = 1, \dots, n)$ *in Assumption* 4, *the stepsize* $\alpha_{j_k^i}$ *is computed using Algorithm* NDFLS1 *for each* k *and* i *and* $\rho_{j_k^i} \to 0$ *for every infinite subsequence such that* $\alpha_{j_k^i} = 0$.

Then the algorithm produces an infinite sequence that admits limit points and every limit point $\bar{x}$ *of* $\{x_k\}$ *is in* $\mathcal{L}_0$ *and satisfies* $\nabla f(\bar{x}) = 0$.

Proof Taking into account the assumptions on f and $\mathcal{L}_0$ and assumption (a), preliminarily we observe that all the hypotheses of Lemma 3 are satisfied and hence the assertions of Lemma 3 must hold. It follows, in particular, that $\{f(x_k)\}$ converges, that $x_k \in \mathcal{L}_0$ for all k, that $\{x_k\}$ has limit points and that every limit point of the sequence is in $\mathcal{L}_0$. We must show that every limit point of $\{x_k\}$ is also a stationary point of f.

Let $\bar{x} \in \mathcal{L}_0$ be a limit point of $\{x_k\}$ and let $\{x_k\}_K$ be a subsequence converging to $\bar{x}$. Consider the search directions $d_{j_k^i}$, for $i = 1, \dots n$, introduced in Assumption 4 and let $p_k^i = d_{j_k^i}/\|d_{j_k^i}\|$, for $i = 1, \dots, n$ be the columns of the matrix P_k defined there. As all the sequences $\{p_k^i\}$ are bounded there exists a subsequence $\{x_k\}_{K_1}$, with $K_1 \subseteq K$ such that

$$\lim_{k \in K_1, k \to \infty} p_k^i = \bar{p}^i, \qquad i = 1, \dots, n. \tag{9}$$

By Assumption 4, we have that the vectors $\bar{p}^i$, $i = 1, \dots, n$ are linearly independent. By Lemma 3(iii), we have that

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

and hence, as $j_k^i \le k + N$, it can be easily established, by induction, that all the points $x_{j_k^i}$ converge to $\bar{x}$ for $k \in K_1$, $k \to \infty$ and for all $i = 1, \dots, n$.

As α_k is computed through Algorithm NDFLS1 and the assumptions of Proposition 1 are satisfied, it follows from assertion (ii) of this proposition that:

$$\lim_{k \in K_1,\ k \to \infty} \frac{\nabla f(x_{j_k^i})^T d_{j_k^i}}{\|d_{j_k^i}\|} = \nabla f(\bar{x})^T \bar{p}^i = 0, \quad i = 1, \dots, n. \tag{10}$$

Since vectors $\bar{p}^i$ are linearly independent, we obtain $\nabla f(\bar{x}) = 0$. □

Remark 6 If we suppose that for each k, in the iteration $x_{k+1} = x_k + \alpha_k d_k$, the stepsize α_k is computed through Algorithm NDFLS1, then both assumption (a) and (c) in Proposition 5 are automatically satisfied and hence the statement can be simplified. However, the assumptions stated above are less restrictive, in the sense that some subsequence of iterations can be carried out without performing a line search, provided that a sufficient (non monotone) reduction is guaranteed at every step. This could be useful for introducing further relaxations of monotonicity, through, for instance, nonmonotone watchdog rules as in [15]. □

The convergence results given above can be easily extended to algorithms employing other types of search directions. To illustrate one of the most significant extensions, first we recall from [21] that the *positive span* of a set of vectors $\{v_1, \ldots, v_r\}$ is the cone

$$\left\{ y \in R^n : \; y = \sum_{i=1}^{r} \zeta_i v_i, \quad \zeta_i \geq 0, \quad i = 1, \ldots, r \right\}.$$

A *positive spanning set* in R^n is a set of vectors whose positive span is R^n. The set $\{v_1, \ldots, v_r\}$ is said to be *positively dependent* if one of its vectors is a positive combination of the others; otherwise, the set is *positively independent*. A *positive basis* in R^n is a positively independent set whose positive span is R^n. Two examples of positive basis for R^n are:

– The coordinate directions and their negative counterparts, that is

$$\{s_1, \ldots, s_{2n}\} = \{e_1, \ldots, e_n, -e_1, \ldots, -e_n\};$$

– The coordinate directions and the negative of their sum, that is

$$\{s_1, \ldots, s_{n+1}\} = \left\{ e_1, \ldots, e_n, -\sum_{i=1}^{n} e_i \right\}.$$

Now, let $\{x_k\}$ be a sequence of points produced through the iteration $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \neq 0$ for all k and $\alpha_k \geq 0$. We suppose that the sequence $\{d_k\}$ satisfies the following assumption, which can be related to the definition of 'class (a) of search directions' introduced in [26].

Assumption 7 There exist integers $N > 0$ and $r > 0$ and r integer sequences $\{j_k^i\}$, $(i = 1, \ldots, r)$, such that

a) $k \leq j_k^1 \leq j_k^2 \leq \ldots \leq j_k^r \leq k + N$ for each $k \geq 0$;
b) Every limit point of the sequence of $n \times r$ matrices

$$Q_k = \left(\frac{d_{j_k^1}}{\|d_{j_k^1}\|} \, \frac{d_{j_k^2}}{\|d_{j_k^2}\|} \cdots \frac{d_{j_k^r}}{\|d_{j_k^r}\|} \right) \quad (k = 0, 1, \ldots),$$

where $\|d_{j_k^i}\| > 0$ for all $k \geq 0$ and $i = 1, \ldots, r$, is a matrix in $R^{n \times r}$ whose columns positively span R^n. □

In this case, we suppose that the line search is carried out only for $\alpha \geq 0$ along each direction, and hence we refer to the line search algorithm NDFLS2. Then we can state a convergence result similar to that given in Proposition 5.

Proposition 8 *Let $f : R^n \to R$ be a continuously differentiable function and assume that the level set $\mathcal{L}_0$ is compact. Let $\{x_k\}$ be the sequence of points produced by an algorithm of the form $x_{k+1} = x_k + \alpha_k d_k$, where $d_k \neq 0$ for all k and $\alpha_k \in R$. Suppose that:*

(a) *For every k we have*

$$f(x_{k+1}) \le W_k - \sigma(\|x_{k+1} - x_k\|), \tag{11}$$

where $\sigma : R^+ \to R^+$ is a forcing function and W_k is a reference value that satisfies (3) *for a given integer $M \ge 0$.*

(b) *The sequence of search directions $\{d_k\}$ satisfies Assumption* 7*;*

(c) *In correspondence to the sequences $\{j_k^i\}$, $(i = 1, \dots, r)$ in Assumption* 7*, the stepsize $\alpha_{j_k^i}$ is computed using Algorithm* NDFLS2 *for each k and i and $\rho_{j_k^i} \to 0$ for every infinite subsequence such that $\alpha_{j_k^i} = 0$.*

Then the algorithm produces an infinite sequence that admits limit points and every limit point $\bar{x}$ of $\{x_k\}$ is in $\mathcal{L}_0$ and satisfies $\nabla f(\bar{x}) = 0$.

Proof Reasoning as in the proof of Proposition 5, we can establish that Lemma 3 holds. Let $\bar{x}$ be a limit point of $\{x_k\}$ and let $\{x_k\}_K$ be a subsequence converging to $\bar{x} \in \mathcal{L}_0$. Consider the search directions

$$d_{j_k^i}, \quad j_k^i \in \{k, k+1, \dots, k+N\}, \quad i = 1, \dots r,$$

introduced in Assumption 7 and let q_k^i be the columns of the $n \times r$ matrix Q_k defined there. As all the sequences $\{q_k^i\}$ are bounded there exists a subsequence $\{x_k\}_{K_1}$, with $K_1 \subseteq K$ such that

$$\lim_{k \in K_1, k \to \infty} q_k^i = \bar{q}^i, \qquad i = 1, \dots, r. \tag{12}$$

By Assumption 7, we have that $\bar{q}^i$, $i = 1, \dots, r$ represent a positive basis in R^n. Using Lemma 3(iii), we have that points $x_{j_k^i}$ converge to $\bar{x}$ for $k \in K_1, k \to \infty$ and for all $i = 1, \dots, r$. As α_k is computed through Algorithm NDFLS2 and all the assumptions of Proposition 2 hold, it follows from (ii) of this proposition that:

$$\lim_{k \in K_1,\ k \to \infty} \frac{\nabla f(x_{j_k^i})^T d_{j_k^i}}{\|d_{j_k^i}\|} = \nabla f(\bar{x})^T \bar{q}^i \ge 0, \quad i = 1, \dots, r. \tag{13}$$

Since vectors $\bar{q}^i$ represent a positive basis in R^n, we can write

$$-\nabla f(\bar{x}) = \sum_{i=1}^{r} \zeta_i \bar{q}^i, \quad \zeta_i \ge 0,$$

so that, by (13), we have

$$-\|\nabla f(\bar{x})\|^2 = \sum_{i=1}^{r} \zeta_i \nabla f(\bar{x})^T \bar{q}^i \ge 0.$$

Thus, we obtain

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

□

5 Hooke–Jeeves method with nonmonotone line searches

With reference to the scheme of Sect. 2 we consider here the case where the set D is not changed during the iterations and consists in the set defined by

$$D = \{e_1, \dots, e_n\},$$

where $e_j \in R^n$ is the j-th column of the identity $n \times n$.

We can describe, by means of a single scheme, the nonmonotone versions of the coordinate method and of the Hooke–Jeeves method; in particular in the following scheme the coordinate method can be obtained by terminating each major iteration at the end of Step 1.

NonMonotone Hooke-Jeeves (NMHJ) Algorithm

```
Data. Starting point $x_0 \in R^n$, $\theta \in (0,1)$, $\rho_0 > 0$, $D = \{e_1, e_2, \dots, e_n\}$.
Set $k = 0$.
For $\ell = 0, 1, \dots$
  Set $y^0 = x_k$.
  Step 1. Coordinate search
    For $i = 1, \dots, n$
        set $d_k = e_i$;
        choose an initial stepsize $\Delta_k > 0$ and calculate step $\alpha_k$ along $d_k$
        using Algorithm NDFLS1($d_k$, $\Delta_k$, $\rho_k$);
        set $x_{k+1} = x_k + \alpha_k d_k$;
        if $\alpha_k = 0$, set $\rho_{k+1} = \theta\rho_k$, otherwise set $\rho_{k+1} = \rho_k$
        set $k = k + 1$.
    End For
  Step 2. Pattern search
    Set $d_k = x_k - y^0$.
    Choose an initial stepsize $\Delta_k > 0$ and calculate step $\alpha_k$ along $d_k$
    using Algorithm NDFLS1($d_k$, $\Delta_k$, $\rho_k$);
    set $x_{k+1} = x_k + \alpha_k d_k$;
    if $\alpha_k = 0$ set $\rho_{k+1} = \theta\rho_k$, otherwise set $\rho_{k+1} = \rho_k$;
    set $k = k + 1$.
End For
```

The convergence of the algorithm is established in the next proposition.

Proposition 9 *Let $f : R^n \to R$ be a continuously differentiable function and assume that the level set $\mathcal{L}_0$ is compact. Let $\{x_k\}$ be the sequence of points produced by Algorithm* NMHJ. *Then the algorithm produces an infinite sequence of points in $\mathcal{L}_0$, such that there exist limit points and every limit point $\bar{x}$ of $\{x_k\}$ satisfies $\nabla f(\bar{x}) = 0$.*

Proof In order to prove the assertion, we need to show that the sequence generated by the algorithm satisfies conditions (a), (b) and (c) of Proposition 5. As for every k the condition of sufficient reduction used in Algorithm NDFLS1 must be met, in correspondence to the forcing function $\sigma(t) = \gamma t^2$, assumption (a) of Proposition 5 is satisfied. Now, let x_k be one of the points generated at the major step ℓ. Then, it is easily seen that, after at most n searches, at Step 1 of the next major step $\ell + 1$, we assume as search directions the coordinate directions and hence the sequence $\{d_k\}$ satisfies Assumption 4 with P_k defined as the identity matrix $n \times n$ for all k. Thus also condition (b) of Proposition 5 holds. Finally, the instructions of the algorithm guarantee that condition (c) of Proposition 5 holds. □

6 Rosenbrock method with nonmonotone line searches

We introduce, in this section, a linesearch-based version of the Rosenbrock method. First we illustrate the procedure used for constructing periodically a basic set of search directions. Then we describe two different algorithms:

1) A nonmonotone linesearch-based Rosenbrock method employing bidirectional searches along n directions;
2) A nonmonotone linesearch-based Rosenbrock method employing non negative searches along $n + 1$ directions.

In the Rosenbrock approach, a set of orthogonal directions is rotated at each major step, so that at least one of the new directions is more closely conformed to the local behavior of the function. We consider:

- An orthonormal set of vectors $D = \{d^i, i = 1, \dots, n\}$ given initially or determined during a cycle of previous iterations;
- A set of values $\sigma^i \in R,\ i = 1, \dots, n$ representing the movements performed along the vectors d^i in a cycle of previous iterations.

The new set of directions

$$\bar{D} = \{\bar{d}^i, i = 1, \dots, n\}$$

is obtained by using the following scheme:

```
                Search Directions Generation(SDG)
For $i = 1, \dots, n$
    Step 1. Set
        $a^i = \begin{cases} d^i & \text{if } \sigma^i = 0 \\ \sum_{j=i}^{n} \sigma^j d^j & \text{if } \sigma^i \neq 0 \end{cases}$
    Step 2. Set
        $b^i = \begin{cases} a^i & \text{if } i = 1 \\ a^i - \sum_{j=1}^{i-1} ((a^i)^\mathsf{T} \bar{d}^j) \bar{d}^j & \text{if } i \geq 2 \end{cases}$
        with $\bar{d}^i = \frac{b^i}{\|b^i\|}$
End For
```

At Step 1, if $\sigma^i \neq 0$, a vector a^i is calculated which represents the sum of all the movements made in the directions d^j for $j = i, \dots, n$. In particular, for $\sigma_1 \neq 0$ the vector $\bar{d}^1$ connects the point x from which we start our search and the point obtained at the end of the search, that is:

$$\bar{x} = x + \sum_{j=1}^{n} \sigma^j d^j.$$

At Step 2, the new orthogonal vector $\bar{d}^i$ is obtained by a Gram–Schmidt orthogonalization procedure. In other words, when $\sigma_1 \neq 0$ the vector $\bar{d}^1$ represents the direction of farthest advance, $\bar{d}^2$ is the best direction which can be found normal to $\bar{d}^1$, and so on.

When $\sigma^i = 0$, we simply set $a^i = d^i$ and it can be shown that $\bar{d}^i = d^i$. It has been proved (see [2]) that the new directions generated by the SDG Algorithm are linearly independent and orthogonal. More precisely, the following result can be established.

Proposition 10 *Let us assume that d^i, $i = 1, \dots, n$ are linearly independent and mutually orthogonal. Then the directions $\bar{d}^i$, $i = 1, \dots, n$ generated by the SDG Algorithm are also linearly independent and mutually orthogonal for any set σ^i, $i = 1, \dots, n$. Furthermore, if $\sigma^i = 0$, then $\bar{d}^i = d^i$.* □

Let us denote by $\sigma = (\sigma^1, \sigma^2, \dots, \sigma^n)^T$ the n-vector of the movements along the directions d^i, for $i = 1, \dots, n$, starting from x. Then, the computation performed with the preceding scheme will be indicated by

$$\bar{D} = \text{SDG}(D, \sigma).$$

At each step, the method of Rosenbrock originally proposed in [33] takes discrete steps along the search directions. In [2], a version of the method that utilizes exact line searches is presented and convergence results are reported, under rather restrictive assumptions on the objective functions.

Now we describe the two new versions of the Rosenbrock method, based on the use of nonmonotone, inexact derivative-free line search techniques, and we prove convergence towards stationary points under usual assumptions. We first consider the version which uses n bidirectional searches carried out by means of Algorithm NDFLS1. We report here the algorithm scheme.

NonMonotone LineSearch-based Rosenbrock (NMLSR1) Algorithm 1

```
Data. Starting point $x_0 \in R^n$, $\theta \in (0,1)$, $\rho_0 > 0$, $D = \{d^1, d^2, \dots, d^n\}$, where
$d^i = e_i$, $i = 1, \dots, n$, $k = 0$.
For $\ell = 0, 1, \dots$
  Step 1. Coordinate search
    For $i = 1, \dots, n$
        set $d_k = d^i$;
        choose an initial stepsize $\Delta_k > 0$ and calculate step $\alpha_k$ along $d_k$
        using Algorithm NDFLS1($d_k$, $\Delta_k$, $\rho_k$).
        set $x_{k+1} = x_k + \alpha_k d_k$,   $\sigma^i = \alpha_k$; if $\alpha_k = 0$ set $\rho_{k+1} = \theta\rho_k$,
        otherwise set $\rho_{k+1} = \rho_k$;
        set $k = k + 1$.
    End For
  Step 2. Coordinate rotation
    Compute the new set of search directions through Rosenbrock rotation,
    that is $\bar{D} = \mathrm{SDG}(D, \sigma)$;
    set $D = \bar{D}$.
End For
```

In the following proposition, we report the main result about the convergence of the algorithm.

Proposition 11 *Let $f : R^n \to R$ be a continuously differentiable function and assume that the level set $\mathcal{L}_0$ is compact. Let $\{x_k\}$ be the sequence of points produced by Algorithm* NMLSR1. *Then the algorithm produces an infinite sequence of points in $\mathcal{L}_0$, such that there exist limit points and every limit point $\bar{x}$ of $\{x_k\}$ satisfies $\nabla f(\bar{x}) = 0$.*

Proof In order to prove the assertion, we need to show that the sequence generated by the algorithm satisfies assumptions (a), (b) and (c) of Proposition 5. First of all, since at every iteration the acceptability condition of sufficient reduction used in Algorithm

NDFLS1 must hold, assumption (a) of Proposition 5 is satisfied in correspondence to the forcing function $\sigma(t) = \gamma t^2$.

Now, by Proposition 10, we have that, every time Step 2 is performed in the algorithm, a new orthonormal set of n search directions is generated. Therefore, for each k it is possible to find an index $\bar{k} \geq k$, with $\bar{k} \leq k + n - 1$, where Step 2 is performed, and a finite positive value $N \leq 2n - 1$ such that the n indices

$$k \leq j_k^1 \leq j_k^2 \leq \cdots \leq j_k^n \leq k + N$$

are related to the orthonormal set of directions $d_{j_k^i}$, $i = 1, \dots, n$ previously generated at Step 2. As the columns of the matrix P_k appearing in Assumption 4 are defined by $p_k^i = d_{j_k^i}/\|d_{j_k^i}\|$ for $i = 1, \dots, n$, we have that also the vectors p_k^i, $i = 1, \dots, n$ constitute an orthonormal set. Then we have, by continuity, that every limit point $(\bar{p}^1, \bar{p}^2, \dots \bar{p}^n)$ of $\{(p_k^1, p_k^2, \dots p_k^n)\}$ yields n orthonormal vectors $\bar{p}^i \in R^n, i = 1, \dots, n$. Thus, assumption (b) of Proposition 5 is satisfied. Finally, the instructions at Step 1 of the algorithm guarantee that the assumptions on ρ_k are satisfied and also (c) of Proposition 5 holds. $\square$

As we have already said at the beginning of the section, we can consider, in alternative to Algorithm 1, a version of the Rosenbrock algorithm with (nonmonotone) derivative-free line searches and $n + 1$ search directions. In Sect. 4, we have seen that a positive basis can be obtained by the coordinate directions and the negative of their sums. In general, if we have a set of linearly independent mutually orthogonal vectors $D_a = \{d^1, \dots, d^n\}$, we can always build a positive basis by adding to the set D_a the negative of their sums, possibly scaled with an arbitrary positive constant $\xi > 0$, that is

$$D = \left\{ d^1, \dots, d^n, -\xi \sum_{j=1}^{n} d^j \right\}.$$

Now we define a modified version of the Rosenbrock algorithm that uses $n + 1$ line searches for non negative values of the stepsize and hence makes use of Algorithm NDFLS2.

```
NonMonotone LineSearch-based Rosenbrock (NMLSR2) Algorithm 2

Data. Starting point $x_0 \in R^n$, $\theta \in (0,1)$, $\rho_0 > 0$, $\ell = 0$, $k = 0$,
      $D_a = \{d^1, d^2, \dots, d^n\}$,  $D = D_a \cup \{d^{n+1}\}$, where $d^i = e_i$,  $i = 1, \dots, n$,
      $d^{n+1} = -\sum_{i=1}^n d^i$.

For $\ell = 0, 1, \dots$
  Step 1. Set $y^0 = x_k$;
  Step 2. Search on a positive basis
    For $i = 1, \dots, n+1$
        set $d_k = d^i$;
        choose an initial stepsize $\Delta_k > 0$ and calculate step $\alpha_k \geq 0$ along $d_k$ using
        Algorithm NDFLS2($d_k$, $\Delta_k$, $\rho_k$);
        set $x_{k+1} = x_k + \alpha_k d_k$; if $\alpha_k = 0$ set $\rho_{k+1} = \theta\rho_k$, otherwise set $\rho_{k+1} = \rho_k$;
        set $k = k + 1$.
    End For
  Step 3. Set $\sigma_i = (x_k - y^0)^T d^i$,   $i = 1, \dots, n$.
  Step 4. Coordinate rotation
    Compute a new set of search directions by first employing Rosenbrock
    rotation of $D_a$, that is
        $\bar{D}_a = \mathrm{SDG}(D_a, \sigma)$,
    and then setting
        $\bar{d}^{n+1} = -\sum_{i=1}^{n} \bar{d}^i$;   $\bar{D} = \bar{D}_a \cup \{\bar{d}^{n+1}\}$;
    set $D = \bar{D}$.
End For
```

In the following proposition, we establish the convergence of the algorithm.

Proposition 12 *Let $f : R^n \to R$ be a continuously differentiable function and assume that the level set $\mathcal{L}_0$ is compact. Let $\{x_k\}$ be the sequence of points produced by Algorithm* NMLSR2 *and assume that $d_k \neq 0$ for all k. Then the algorithm produces an infinite sequence such that every limit point $\bar{x}$ of $\{x_k\}$ satisfies $\nabla f(\bar{x}) = 0$.*

Proof We show that the sequence generated by our algorithm satisfies assumptions (a), (b) and (c) of Proposition 8. By the instructions at Step 2 of the algorithm, the stepsize α_k along d_k is calculated using Algorithm NDFLS2 for every k and hence (a) is satisfied.

By Proposition 10, every time Step 4 is performed in the algorithm, a new positive basis in R^n is obtained by adding the negative sum to the set of linearly independent and mutually orthogonal directions generated by means of Algorithm SDG. As Step 4 is performed after all $n + 1$ directions have been explored, for each k it is possible to find an iteration index $\bar{k}$, $k \le \bar{k} \le k+n$, where Step 4 is carried out, and a finite positive value $N \le 2(n+1)$ such that the $n+1$ indices $k \le j_k^1 \le j_k^2 \le \cdots \le j_k^{n+1} \le k+N$ are related to the vectors $d_{j_k^i}$, $i = 1, \ldots, n+1$ previously generated at Step 4.

By construction, we have that $\{d_{j_k^1}, \ldots, d_{j_k^n}\}$ is an orthonormal set and that

$$d_{j_k^{n+1}} = -\sum_{i=1}^{n} d_{j_k^i},$$

where $\|d_{j_k^{n+1}}\| > 0$, because of the linear independence of $\{d_{j_k^1}, \ldots, d_{j_k^n}\}$. As the columns $\{q_k^i\}$, $i = 1, \ldots, n+1$, of the matrix Q_k appearing in Assumption 7 are defined by

$$q_k^i = d_{j_k^i}/\|d_{j_k^i}\|, \quad (i = 1, \ldots, n+1),$$

we have that $\{q_k^1, q_k^2, \ldots, q_k^n\}$ is an orthonormal set. Therefore at every limit point $(\bar{q}^1, \bar{q}^2, \ldots, \bar{q}^{n+1})$ of the sequence $\{(q_k^1, q_k^2, \ldots, q_k^{n+1})\}$ we have, by continuity, that $\{\bar{q}^1, \bar{q}^2, \ldots, \bar{q}^n\}$ is an orthonormal set too, and that

$$\bar{q}^{n+1} = -\frac{1}{\|\sum_{i=1}^{n} \bar{q}^i\|} \sum_{i=1}^{n} \bar{q}^i,$$

where $\|\sum_{i=1}^{n} \bar{q}^i\| > 0$, bacause of the linear independence of $\{\bar{q}^1, \bar{q}^2, \ldots \bar{q}^n\}$. Then it can be easily verified, as already observed, that the vectors $\bar{q}^1, \bar{q}^2, \ldots \bar{q}^{n+1}$ constitute a positive basis of R^n and hence assumption (b) is satisfied. Finally, as $\rho_{k+1} = \theta\rho_k$ if $\alpha_k = 0$ also assumptions (c) is satisfied. $\square$

7 Linesearch-based algorithms employing gradient approximations and Rosenbrock rotations

In this section, we describe a new class of derivative-free algorithms, structured as indicated in the general scheme of Sect. 2, which makes use of *simplex gradients* and of Rosenbrock rotations for defining the search directions. Along these directions the stepsize are computed through nonmonotone, inexact, derivative-free line searches.

In the scheme of Sect. 2, after a cycle of line searches at Step (a) is carried out along a set of given directions d^i, for $i = 1, \ldots, n$, we introduce at Step (b) the computation of a simplex gradient g (in an extended sense). During the searches at Step (a) we further store (al least) $n+1$ points $y^i \in R^n$ and the corresponding function values $f(y^i)$, for $i = 0, 1, \ldots, n$ and we compute the simplex gradient g using these data. Then, given the set of points $\{y^0, \ldots, y^n\}$, we can define (see, e.g. [5]) the gradient approximation at $x_k \equiv y^n$ as the solution of the least squares problem

$$\min_g \left\| S^T g - \delta(f) \right\|^2,$$

where:

$$S = \left[y^0 - y^n, \ldots, y^{n-1} - y^n \right] \quad \text{and}$$

$$\delta(f) = \left[f(y^0) - f(y^n), \ldots, f(y^{n-1}) - f(y^n) \right]^\mathsf{T}.$$

In principle, if we assume that S is square and nonsingular, we could solve the system

$$S^T g = \delta(f).$$

However, even in this case, it could be more convenient to give an approximate solution of the system through the approximate minimization of an error function. The solution of a least squares problem is obviously required when S singular or nonsquare, for instance because of the fact that we make use of other past data gathered during previous linesearches.

The illustration of a major step of our algorithm is given in Fig. 1, where we have assumed, for simplicity, that all stepsizes are positive.

We now formally state the algorithm where we combine simplex gradients with Rosenbrock rotations.

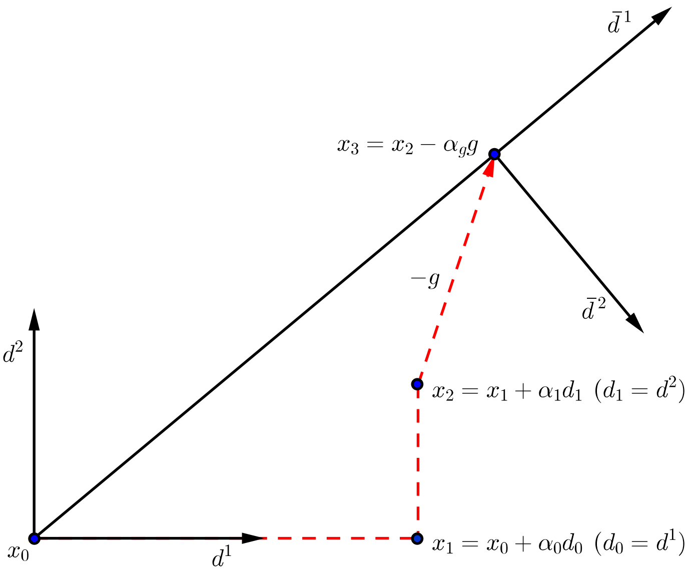
$\bar{d}^1$
$x_3 = x_2 - \alpha_g g$
$-g$
$\bar{d}^2$
d^2
$x_2 = x_1 + \alpha_1 d_1 \ (d_1 = d^2)$
x_0
d^1
$x_1 = x_0 + \alpha_0 d_0 \ (d_0 = d^1)$

Fig. 1 Illustration of a major step of Algorithm NMDFU

NonMonotone Derivative-Free Unconstrained (NMDFU) Algorithm

```
Data. Starting point $x_0 \in R^n$, $\theta \in (0,1)$, $\rho_0 > 0$, $k = 0$ and
      $D = \{d^1, d^2, \dots, d^n\}$, where $d^i = e_i,\ i = 1, \dots, n$.
For $\ell = 0, 1, \dots$
  Set $y^0 = x_k$.
  Step 1. Coordinate search
    For $i = 1, \dots, n$
        set $d_k = d^i$, choose an initial stepsize $\Delta_k > 0$ and calculate step $\alpha_k$ along $d_k$
        using Algorithm NDFLS1($d_k$, $\Delta_k$, $\rho_k$);
        if $\alpha_k = 0$ set $y^i = x_k + \Delta_k d^i$ else set $y^i = x_k + \alpha_k d^i$;
        set $x_{k+1} = x_k + \alpha_k d_k$,   $\sigma^i = \alpha_k$; if $\alpha_k = 0$ set $\rho_{k+1} = \theta\rho_k$,
        otherwise set $\rho_{k+1} = \rho_k$;
        set $k = k + 1$.
    End For
  Step 2. Gradient approximation calculation
    Set $S = [y^0 - y^n, \dots, y^{n-1} - y^n]$,   $\delta(f) = [f(y^0) - f(y^n), \dots, f(y^{n-1}) - f(y^n)]^\mathsf{T}$;
    calculate the gradient approximation $g$ by solving the least squares problem
                $\min_g \|S^T g - \delta(f)\|^2$,
    and set $d_k = -g$.
  Step 3. Line search along the gradient approximation
    Choose an initial stepsize $\Delta_k > 0$, calculate step $\alpha_k \geq 0$ along $d_k = -g$ using
    Algorithm NDFLS2($d_k$, $\Delta_k$, $\rho_k$);
    set $x_{k+1} = x_k + \alpha_k d_k$; if $\alpha_k = 0$ set $\rho_{k+1} = \theta\rho_k$, otherwise set $\rho_{k+1} = \rho_k$;
    if $\alpha_k \neq 0$ set $\sigma^i = (x_{k+1} - y^0)^T d^i$, with $i = 1, \dots, n$;
    set $k = k + 1$.
  Step 4. Coordinate rotation
    Compute the new set of search directions through Rosenbrock rotation, that is
                $\bar{D} = \mathrm{SDG}(D, \sigma)$.
  Step 5. Set $D = \bar{D}$.
End For
```

The convergence of the algorithm is established in the next proposition.

Proposition 13 *Let $f : R^n \to R$ be a continuously differentiable function and assume that the level set $\mathcal{L}_0$ is compact. Let $\{x_k\}$ be the sequence of points produced by Algorithm* NMDFU. *Then the algorithm produces an infinite sequence of points in $\mathcal{L}_0$, such that there exist limit points and every limit point $\bar{x}$ of $\{x_k\}$ satisfies $\nabla f(\bar{x}) = 0$.*

Proof In order to prove the assertion, we need to show that the sequence generated by the algorithm satisfies assumptions (a), (b) and (c) of Proposition 5.

Since at every iteration either the acceptability condition of sufficient reduction used in Algorithm NDFLS1 or in Algorithm NDFLS2 must hold, also assumption (a) of Proposition 5 is satisfied in correspondence to the forcing function $\sigma(t) = \gamma t^2$. It is easily seen that we can follow essentially the same arguments used in the proof of Proposition 11. In fact, every time Step 4 is performed in the algorithm, a new orthonormal set of n search directions is generated and used within $N \leq 2n$ iterations and assumption (b) of Proposition 5 can be established. Finally, the instructions at Step 1 guarantee that also condition (c) of Proposition 5 is satisfied. $\square$

8 Numerical results

In this section we analyze the effects of using coordinate rotations and gradient approximations in a derivative-free context. Our numerical experience can be divided into three parts:

1. In the first part, we compare algorithms NMDFU and NMLSR (Algorithm 1) with a nonmonotone version of the Coordinate Search algorithm (NMCS). The aim of this experiment is analyzing the effects of using coordinate rotations and gradient approximations;
2. In the second part, in order to analyze the impact of nonmonotone linesearches in the proposed framework, we compare two different version of the NMDFU algorithm where we respectively use a nonmonotone and a monotone linesearch;
3. In the third part, we compare the NMDFU Algorithm with NEWUOA [31] and NOMAD [23] two well-known and widely used codes for derivative-free unconstrained optimization. The aim of this experiment is evaluating the efficiency of our approach when compared with other derivative-free solvers.

Consequently, we adopt the same procedure as that used in [28] to evaluate the behavior of the different solvers. We use the following convergence condition:

$$f(x_0) - f(x_k) \geq (1 - \tau)(f(x_0) - f_L) \tag{14}$$

where $0 \leq \tau \leq 1$ is a suitably chosen tolerance and f_L is the smallest function value obtained by any solver within the same maximum computational budget. We consider a set $\mathcal{A}$ of n_a algorithms, a set $\mathcal{P}$ of $|\mathcal{P}|$ problems and a performance measure $m_{p,a}$ (e.g. in our case, number of function evaluations). We compare the performance on problem p by algorithm a with the best performance by any algorithm on this problem using the following *performance ratio*

$$r_{p,a} = \frac{m_{p,a}}{\min\{m_{p,a} \;:\; a \in \mathcal{A}\}}.$$

Then, we obtain a first overall assessment of the performance of the algorithm a by defining the *performance profile*:

$$\rho_a(\alpha) = \frac{1}{|\mathcal{P}|}\text{size}\{p \in \mathcal{P} \ : \ r_{p,a} \leq \alpha\},$$

which approximates the probability for algorithm $a \in \mathcal{A}$ that the performance ratio $r_{p,a}$ is within a factor $\alpha \in R$ of the best possible ratio. The function ρ_a approximates the distribution function for the performance ratio. Thus $\rho_a(1)$ gives the fraction of problems for which the algorithm a was the most effective, $\rho_a(2)$ gives the fraction of problems for which the algorithm a is within a factor of 2 of the best algorithm, and so on. The convention $r_{p,a} = \infty$ is used when algorithm a fails to satisfy the convergence test (14) for problem p.

We further measure performances of the different solvers by the percentage of problems that can be solved (for a given tolerance τ) within a certain number of function evaluations. We define $t_{p,a}$ the number of function evaluations needed for algorithm a to satisfy (14) for a given tolerance τ, and we obtain the percentage of problems solved with ν function evaluations by means of the so called *data profile*:

$$d_a(\nu) = \frac{1}{|\mathcal{P}|}\text{size}\left\{p \in \mathcal{P} \ : \ \frac{t_{p,a}}{n_p + 1} \leq \nu\right\},$$

where n_p is the number of variables in $p \in P$. If the convergence test (14) cannot be satisfied within the assigned computational budget, we set $t_{p,a} = \infty$.

The test set $\mathcal{P}$ we consider in the experiments consists of 38 smooth problems (2 fixed dimension problems with $n = 6$, and 18 variable dimension problems with $n \in \{10, 20\}$) from the CUTEr collection [13] gathered from papers on derivative-free optimization (see, e.g. [1,5]), and 49 nonsmooth problems from the collection of Lukšan and Vlček [27,32]. Since the problems in $\mathcal{P}$ have at most 50 variables, in our numerical experience we set the maximum computational budget to be 5000 and we investigate the behavior of the solvers within this computational budget. We use both performance and data profile with the test (14) where $\tau = 10^{-l}$ with $l \in \{3, 6\}$.

In NMDFU, NMLSR Algorithm 1 and NMCS we used the following reference value:

$$W_k = \max_{0 \leq j \leq \min(k,M)} [f(x_{k-j})],$$

with $M = 3$.

In Fig. 2, we report performance and data profiles related to the first experiment. As we can see, looking at the performance profiles, NMDFU is the fastest solver in at least 55 % of the problems, while NMLSR and NMCS are the fastest solvers respectively in less than 40 and 20 % of the problems at most.

Furthermore, NMDFU guarantees better results than NMLSR and NMCS in terms of robustness and the performance difference between NMLSR and the other two solvers increases as the tolerance decreases. The data profiles show that NMDFU is slightly better than NMLSR as it solves a higher percentage of problems when the number of simplex gradients is sufficiently large. We can also notice that both the algorithms are better than NMCS as they solve a larger percentage of problems for all sizes of computational budget and levels of accuracy τ.

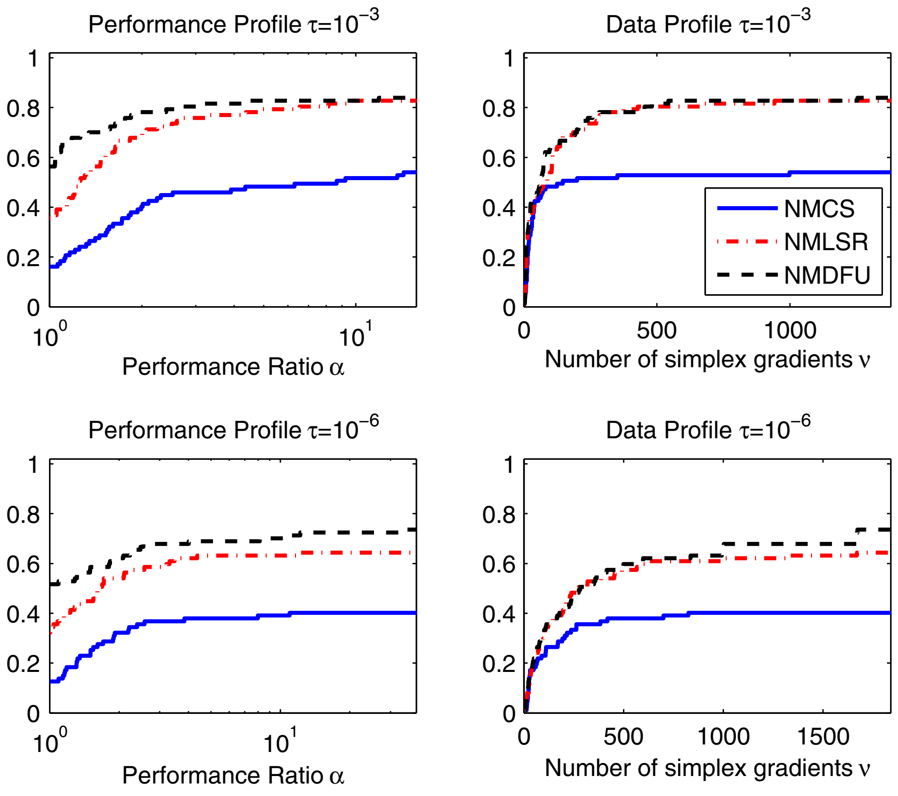

Fig. 2 Performance and data profiles of NMDFU, NMLSR (Algorithm 1) and NMCS

What we can conclude from this first experiment is that, in a derivative-free context, using coordinate rotations and gradient approximations can be beneficial.

According to some preliminary tests (not reported here), NMLSR Algorithm 1 outperforms NMLSR Algorithm 2. This is the reason why we decided not to include Algorithm 2 in our analysis. Anyway, a more careful choice of the $n+1$-th search direction (i.e. a choice that somehow takes into account the information gathered so far) might help improving the performance of Algorithm 2. This is beyond the scope of the present paper, and could be the subject of future research.

As the goal of the second experiment was analyzing the impact of nonmonotone linesearches in the proposed framework, in Fig. 3 we show the performances of two different version of the NMDFU algorithm that respectively use a nonmonotone and a monotone linesearch. In the reference value which defines the sufficient decrease, we set $M = 3$ for the nonmonotone linesearch and $M = 0$ for the monotone one.

By looking at the performance profiles, we can easily see that nonmonotone linesearches guarantee better results both in terms of efficiency and robustness for any value of the tolerance τ. The good behavior of the nonmonotone version of the code is confirmed by the data profiles as they basically say that the nonmonotone version solves a larger percentage of problems for all sizes of computational budget and levels

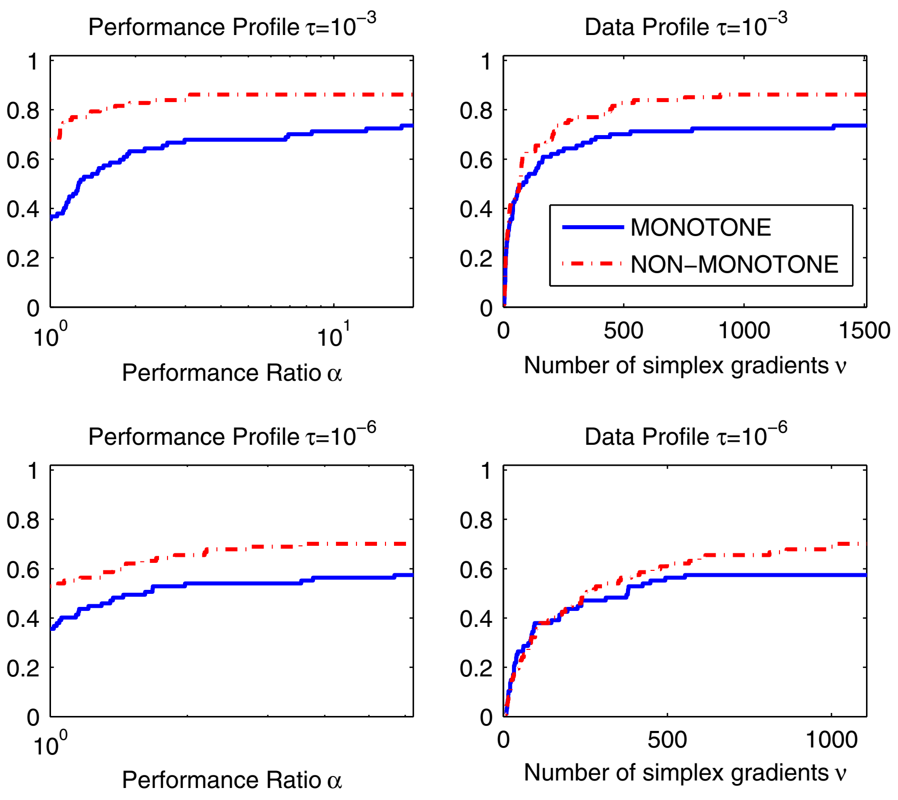

Fig. 3 Monotone versus nonmonotone—performance and data profiles

of accuracy τ. Furthermore, the performance difference between the two tends to be significantly large when the computational budget is large enough.

The results of the third experiment are reported in Figs. 4–7. In order to provide enough information on the progress of all the codes, we set standard convergence tests very tight (e.g. RHOEND=10^{-20} in NEWUOA). All the other parameters of the algorithms tested were set to their standard values.

The performance profiles (Fig. 4) show that NMDFU is competitive with NEWUOA and that they are both better than NOMAD. Furthermore, NMDFU guarantees quite better results than both NEWUOA and NOMAD in terms of robustness.

More specifically, both for $\tau = 10^{-3}$ and for $\tau = 10^{-6}$ NEWUOA and NMDFU show a similar behavior, in terms of efficiency, as they are the fastest in about 40 % of the problems, while NOMAD is the fastest in about 20 % of the problems. When the performance ratio is larger than 1.5 for $\tau = 10^{-3}$ or larger than 2 for $\tau = 10^{-6}$, NMDFU guarantees better results than both NEWUOA and NOMAD. We further have that, for any value of τ, the performance differences between NMDFU and the other two codes becomes significantly large as the ratio increases.

From the data profiles (Fig. 5), we can see that, when the computational budget is small, (say less than 70 simplex gradient evaluations for $\tau = 10^{-3}$ or

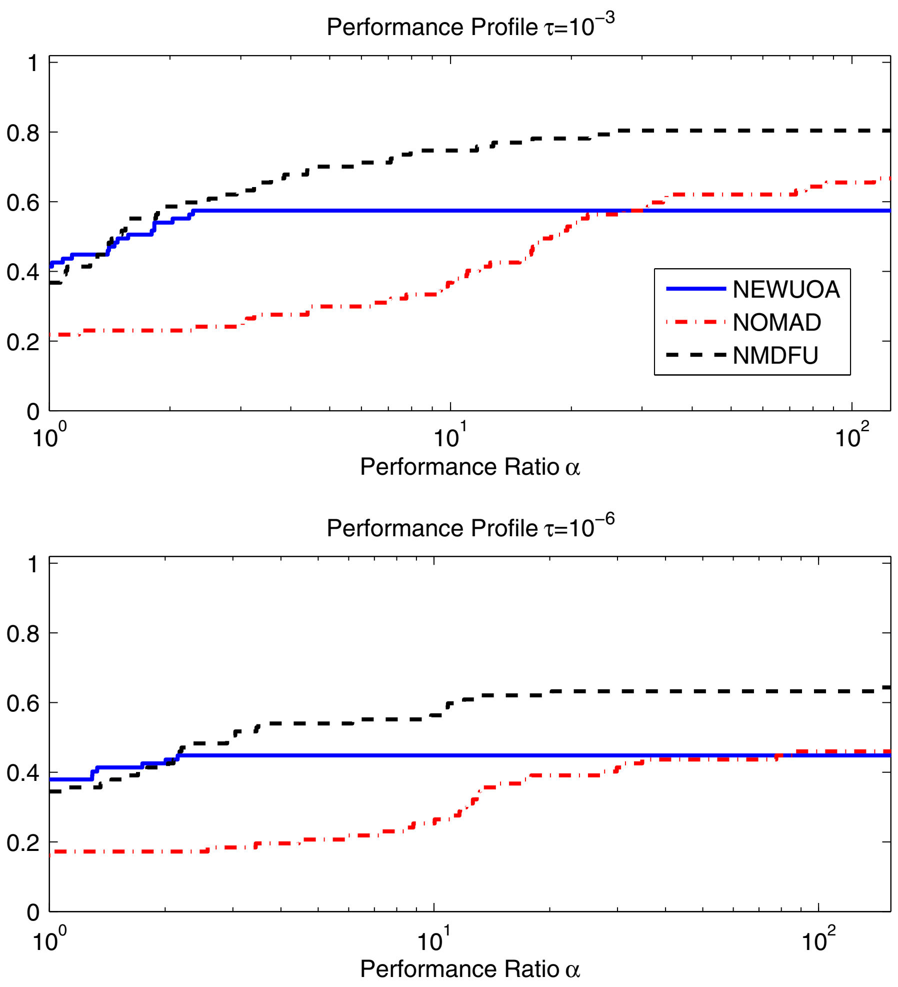

Fig. 4 Performance profiles of NMDFU, NEWUOA and NOMAD

200 for $\tau = 10^{-6}$) NEWUOA guarantees better results than NMDFU, and that these two algorithms are both much better than NOMAD. As the computational budget increases NMDFU solves a larger number of problems than the other solvers and the difference is significantly large as the number of simplex gradients increases.

We can note, in particular, that, from a certain point onward (that is when the number of simplex gradients is larger than 100 for $\tau = 10^{-3}$ or than 200 for $\tau = 10^{-6}$) NMDFU outperforms both NOMAD and NEWUOA. Once the number of simplex gradient evaluations becomes larger than 350, the performance difference between NMDFU and NEWUOA is about 20 % and the difference between NMDFU and NOMAD is about 15–20 %.

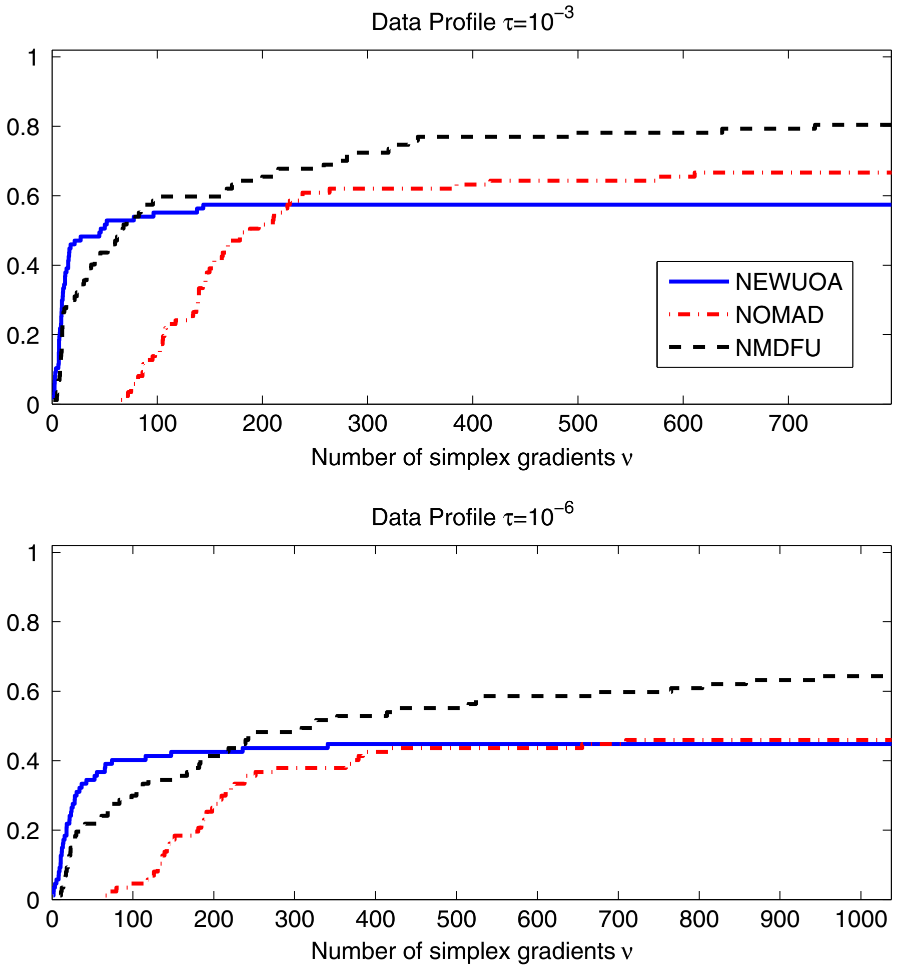

Fig. 5 Data profiles of NMDFU, NEWUOA and NOMAD

In order to better understand the behavior of the codes in difficult cases, we also separately report, in Figs. 6 and 7, performance and data profiles related to the 49 nonsmooth problems taken from [27], extracted from our test set.

As we can easily see by looking at the performance profiles, the NMDFU algorithm guarantees better results both in terms of efficiency and robustness for any value of the tolerance τ. The good behavior of our code is confirmed by the data profiles, as they basically say that NMDFU solves a larger percentage of problems for a computational budget respectively larger than 30 when $\tau = 10^{-3}$ and larger than 100 when $\tau = 10^{-6}$. Furthermore, the performance differences between NMDFU and the other two codes becomes significantly large as the ratio increases.

Finally, we analyze the performances of the NMDFU algorithm when the number of variables of the problem to be solved is (relatively) large. In order to do that,

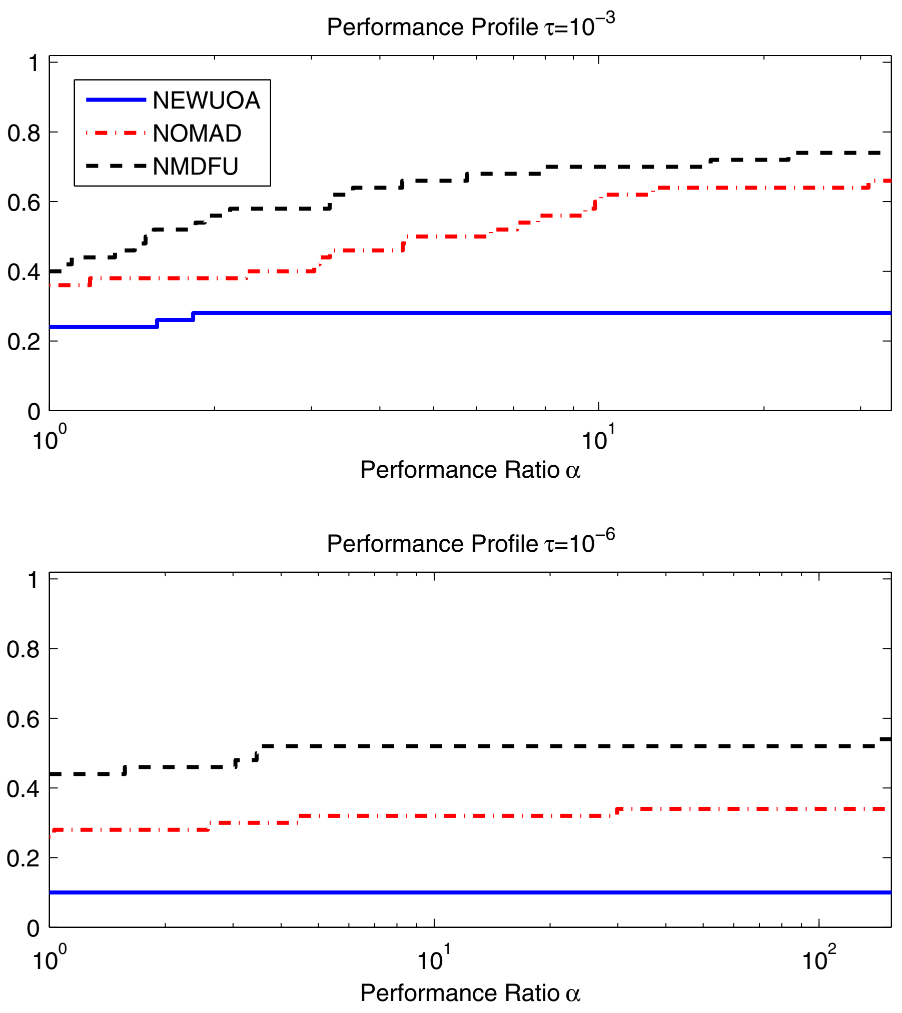

Fig. 6 Performance profiles of NMDFU, NEWUOA and NOMAD (nonsmooth problems)

we compare NMDFU with NEWUOA, which is suitably developed for solving large dimensional problems. We consider all the 18 smooth problems with variable dimension included in $\mathcal{P}$, where we set $n = 100$, and all the 15 nonsmooth and nonconvex problems with $50 \leq n \leq 200$ used in [32], thus obtaining a set of 33 problems. In Fig. 8, we show performance and data profiles related to the comparison of NMDFU and NEWUOA on the 33 large dimensional problems. In this case, we set the computational budget to 20000.

By considering the performance profiles, we can see that NMDFU guarantees better results both in terms of efficiency and robustness for any value of the tolerance τ. The good results are confirmed once again by the data profiles, as we have that our code

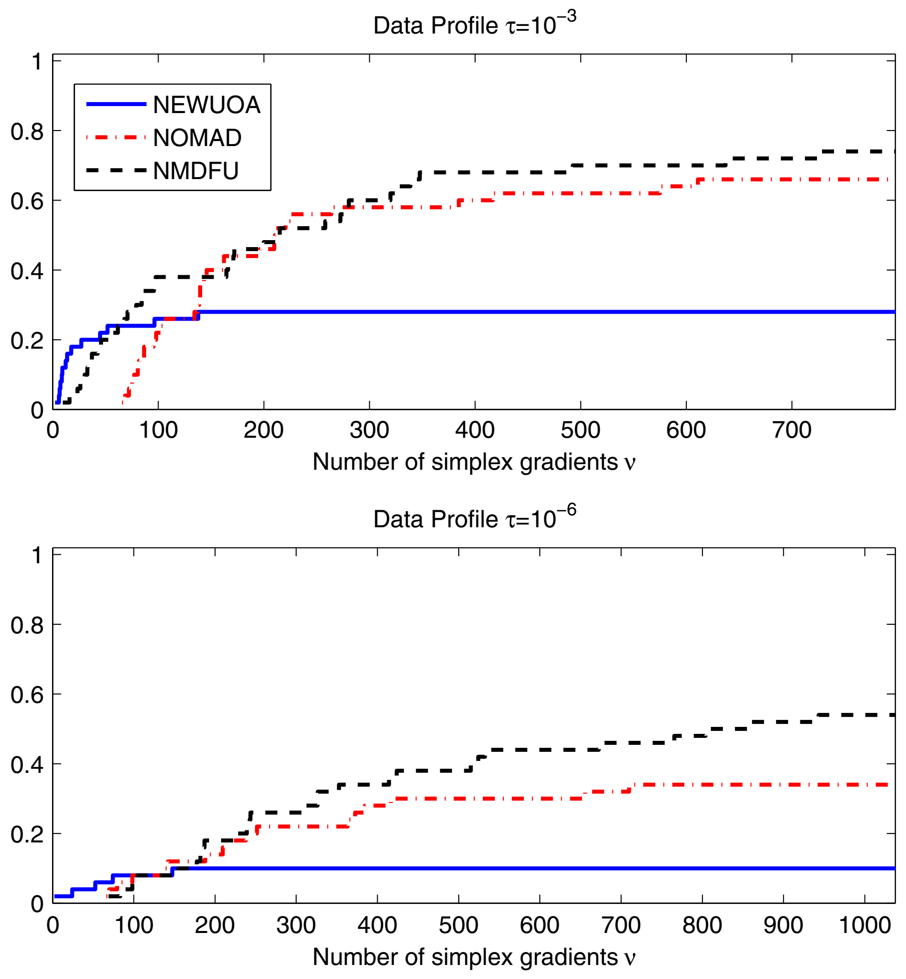

Fig. 7 Data profiles of NMDFU, NEWUOA and NOMAD (nonsmooth problems)

solves a larger percentage of problems, for almost all sizes of computational budget at any levels of accuracy τ.

9 Concluding remarks and future work

The results presented in this paper show that the linesearch-based nonmonotone method proposed here, employing simplex gradients and coordinate rotations appear quite competitive and often superior to some of the best derivative-free techniques presently available. More specifically, on the basis of our preliminary computational experience, it would seem that in most of cases the proposed technique can be much more efficient than mesh adaptive direct search methods (such as NOMAD). In comparison with good model-based methods (such as NEWUOA) the advantages of our

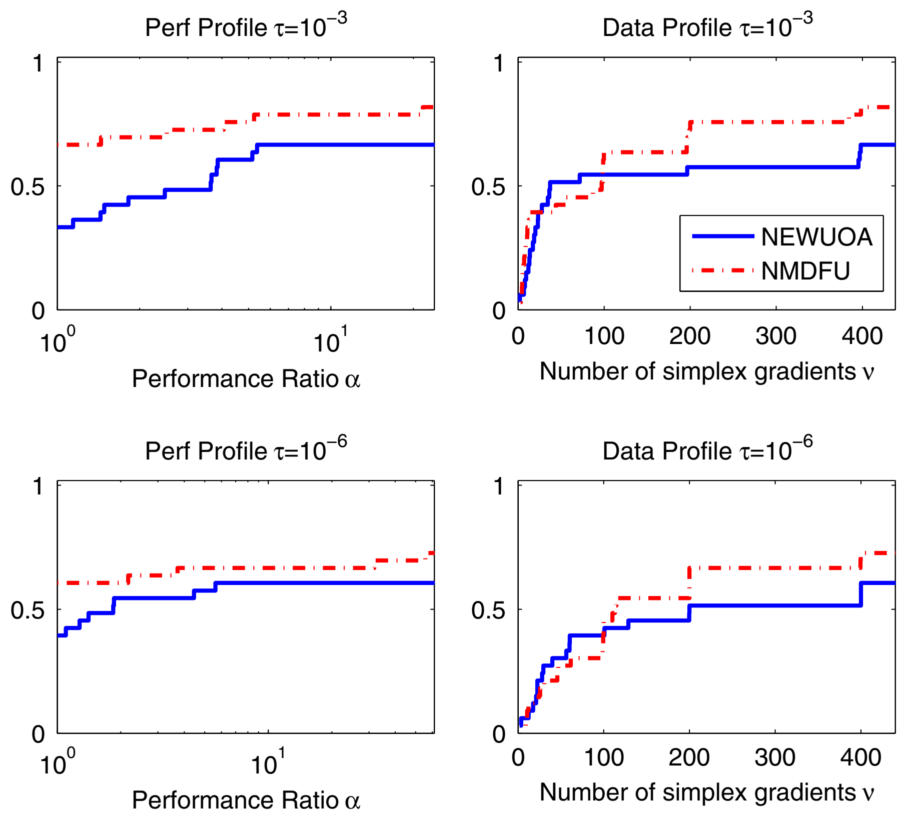

Fig. 8 Comparison of NMDFU and NEWUOA on large dimensional problems

algorithm can be significant in difficult (possibly nonsmooth) problems and in (relatively) large dimensional problems. For small dimensional, well conditioned smooth problems it would appear that NEWUOA performs better in most of cases. Hence, a possible way to further improve the performance of NMDFU could be that of including techniques developed in NEWUOA to generate search directions in the Acceleration step.

Additional work may be needed for improving our code, for evaluating the effect of some parameters and for experimenting other possible choices in the general framework considered here. Future research will include:

- The study of algorithms for large dimensional systems employing decomposition techniques and parallel searches;
- The definition of enhanced nonmonotone acceptance rules, such as, for instance, the combination of nonmonotone linesearches with nonmonotone *watchdog* techniques (see, e.g. [15] in the case of nonlinear equations);
- The definition of more efficient algorithms employing simplex gradients, such as, for instance, spectral gradients or reduced memory quasi-Newton methods;
- The study of extensions to constrained problems.

Acknowledgments The authors would like to thank the anonymous reviewers for their helpful and constructive comments that greatly contributed to improve the paper.

References

1. Abramson, M.A., Audet, C., Dennis Jr, J.E., Le Digabel, S.: Orthomads: a deterministic MADS instance with orthogonal directions. SIAM J. Optim. **20**(2), 948–966 (2009)
2. Bazaraa, M.S., Sherali, H.D., Shetty, C.M.: Nonlinear Programming: Theory and Algorithms, 3rd edn. Wiley, Hoboken (2006)
3. Bortz, D.M., Kelley, C.T.: The simplex gradient and noisy optimization problems. In: Borggaard, J.T., Burns, J., Cliff, E., Schreck, S. (eds.) Computational Methods in Optimal Design and Control, Progress in Systems and Control Theory, vol. 24, pp. 77–90. Birkhäuser, Boston (1998)
4. Conn, A.R., Scheinberg, K., Vicente, L.N.: Introduction to Derivative-Free Optimization. MPS-SIAM Series on Optimization. SIAM, Philadelphia (2008)
5. Custódio, A.L., Vicente, L.N.: Using sampling and simplex derivatives in pattern search methods. SIAM J. Optim. **18**, 537–555 (2007)
6. Custódio, A.L., Dennis, J.E., Vicente, L.N.: Using simplex gradients of nonsmooth functions in direct search methods. IMA J. Numer. Anal. **28**, 770–784 (2008)
7. De Leone, R., Gaudioso, M., Grippo, L.: Stopping criteria for linesearch methods without derivatives. Math. Program. **30**, 285–300 (1984)
8. Diniz-Ehrhardt, M.A., Martnez, J.M., Raydan, M.: A derivative-free nonmonotone line-search technique for unconstrained optimization. J. Comput. Appl. Math. **219**, 383–397 (2008)
9. Frimannslund, L., Steihaug, T.: A generating set search method using curvature information. Comput. Optim. Appl. **38**, 105–121 (2007)
10. García-Palomares, U.M., Rodríguez, J.F.: New sequential and parallel derivative free algorithms for unconstrained minimization. SIAM J. Optim. **13–1**, 79–96 (2002)
11. García-Palomares, U.M., González-Castano, F.J., Burguillo-Rial, J.C.: A combined global and local search approach to global optimization. J. Glob. Optim. **34**, 409–426 (2006)
12. García-Palomares, U.M., García-Urreac, I.J., Rodríguez-Hernández, P.S.: On sequential and parallel non-monotone derivative-free algorithms for box constrained optimization. Optim. Methods Softw. **28**(6), 1233–1261 (2013)
13. Gould, N.I.M., Orban, D., Toint, PhL: CUTEr and SifDec: a constrained and unconstrained testing environment revisited. ACM Trans. Math. Softw. **29**, 373–394 (2003)
14. Grippo, L., Sciandrone, M.: Nonmonotone globalization techniques for the Barzilai–Borwein gradient method. Comput. Optim. Appl. **23**, 143–169 (2002)
15. Grippo, L., Sciandrone, M.: Nonmonotone derivative-free methods for nonlinear equations. Comput. Optim. Appl. **27**, 297–328 (2007)
16. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
17. Grippo, L., Lampariello, F., Lucidi, S.: A class of nonmonotone stabilization methods in unconstrained optimization. Numer. Math. **59**, 779–805 (1986)
18. Grippo, L., Lampariello, F., Lucidi, S.: Global convergence and stabilization of unconstrained minimization methods without derivatives. J. Optim. Theory Appl. **56**, 385–406 (1988)
19. Hooke, R., Jeeves, T.A.: "Direct Search" solution of numerical and statistical problems. J. ACM **8**, 212–229 (1961)
20. Kelley, C.T.: Iterative Methods for Optimization. SIAM Publications, Philadelphia (1999)
21. Kolda, T.G., Lewis, R.M., Torczon, V.: Optimization by direct search: new perspective on some classical and modern methods. ICASE SIAM Rev. **45**, 385–482 (2003)
22. La Cruz, W., Martinez, J.M., Raydan, M.: Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. Math. Comput. **75**, 1449–1466 (2006)
23. Le Digabel, S.: Algorithm 909: NOMAD: nonlinear optimization with the MADS algorithm. ACM Trans. Math. Softw. **37**(4), 44:1–44:15 (2011)
24. Li, D.H., Fukushima, M.: A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. Optim. Methods Softw. **13**, 181–201 (2000)
25. Liuzzi, G., Lucidi, S., Rinaldi, F.: Derivative-free methods for bound constrained mixed-integer optimization. Comput. Optim. Appl. **53**(2), 505–526 (2012)

26. Lucidi, S., Sciandrone, M.: On the global convergence of derivative free methods for unconstrained optimization. SIAM J. Optim. **13**, 97–116 (2007)
27. Lukšan, V., Vlček, J.: Test Problems for Nonsmooth Unconstrained and Linearly Constrained Optimization. Technical Report. Institute of Computer Science, Academy of Sciences of the Czech Republic (2000)
28. Moré, J.J., Wild, S.M.: Benchmarking derivative-free optimization algorithms. SIAM J. Optim. **20**, 17–191 (2009)
29. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York (1970)
30. Powell, M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. Comput. J. **7**, 155–162 (1964)
31. Powell, M.J.D.: The NEWUOA software for unconstrained optimization without derivatives. In: Di Pillo, G., Roma, M. (eds.) Large Scale Nonlinear Optimization, pp. 255–297. Springer, Netherlands (2006)
32. Rios, L.M., Sahinidis, N.V.: Derivative-free optimization: a review of algorithms and comparison of software implementations. J. Glob. Optim. **56**(3), 1247–1293 (2013)
33. Rosenbrock, H.H.: An automatic method for finding the greatest or least value of a function. Comput. J. **3**, 175–184 (1960)