Malaysian Journal of Mathematical Sciences 14(3): 451–471 (2020)

MALAYSIAN JOURNAL OF MATHEMATICAL SCIENCES

Journal homepage: http://einspem.upm.edu.my/journal

Nonmonotone Coordinate Search Method for Bound Constrained Optimization

Frau, J. A. and Pilotta, E. A. *

Facultad de Matemática, Astronomía, Física y Computación – Universidad Nacional de Córdoba, CIEM – CONICET, Medina Allende s/n, Ciudad Universitaria (5000), Córdoba, Argentina

E-mail: elvio.pilotta@unc.edu.ar
* *Corresponding author*

Received: 22 May 2020
Accepted: 12 August 2020

ABSTRACT

A new coordinate search method for bound constrained optimization is introduced. The proposed algorithm employs coordinate directions, in a suitable way, with a nonmonotone line search for accepting the new point, without using derivatives of the objective function. The main global convergence results are strongly based on the relationship between the step length and a stationarity measure. Also, a detailed benchmark study comparing different line search strategies is presented using a well known set of test problems.

Keywords: Pattern search methods, bound constrained optimization, global convergence, nonmonotone line search, numerical experiments.

1. Introduction

We propose a new algorithm to solve bound constrained optimization problems where the derivatives of the objective function are not available. So, the problem of interest is

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in \Omega \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $\Omega = \{x \in \mathbb{R}^n \,|\, l \leq x \leq u\}$ with $-\infty \leq l < u \leq \infty$. We assume that the objective function is continuously differentiable on Ω but the derivative information is unreliable or non-existent.

There are many real world optimization problems where the objective function contains noise, the function evaluation requires complex simulations or there is not an explicit function formulation (blackbox optimization problems). So, Quasi-Newton or finite difference methods could not be applicable. This kind of situations may arise in applications from molecular geometry (see Alberto et al. (2004), Meza and Martínez (1994)), medical image registration (see Oeuvray and Bierlaire (2007)), radiobiology (see Rashid et al. (2018)), shape and design optimization (see Lee et al. (2018)).

Derivative-free optimization has received considerable attention from the optimization community during the last decades, including the establishment of solid mathematical foundations for many of the methods considered in practice. Particularly, pattern search methods have succeeded where more elaborate approaches fail. These methods belong to the family of direct search methods, characterized by unsophisticated implementations, the absence of the construction of a model of the objective function and the use of a set of search directions to explore around the current iterate. See Audet and Dennis Jr (2002), Audet and Hare (2017), Conn et al. (2009), Diniz-Ehrhardt et al. (2019), Gardeux et al. (2017), Torczon (1997).

Direct search methods were initially introduced by Hooke and Jeeves (1961) for unconstrained optimization problems and lately analized and formally presented by Kolda et al. (2003), Torczon (1997). Recently, some strategies were adapted from derivative-based methods and incorporated to pattern search methods. In this way, they have been provided with a solid mathematical framework as well as derivative-based line search techniques. For instance, Diniz-Ehrhardt et al. (2008) introduced a global strategy, based in the ideas developed by Grippo et al. (1986), La Cruz et al. (2006), Li and Fukushima (2000), Lucidi and Sciandrone (2002), that uses a nonmonotone line search scheme in a pattern search algorithm for unconstrained optimization. Lewis and Torczon (1999) extended the pattern search method for the bound constrained case although they did not carry out numerical experiments. This problem was also studied by Arouxét et al. (2011) using polynomial interpolation and trust region strategies, which is a quite different approach to pattern search methods.

In this article, on one hand, we propose a coordinate search method that includes a nonmonotone line search as a globalization strategy for the bound constrained optimization problem (1). It is a particular pattern search method where the set of search directions are the coordinate directions. The new algorithm is based on the ideas introduced by Diniz-Ehrhardt et al. (2008) which has been developed for the unconstrained case whereas our algorithm deals with bound constraints. Furthermore, the proofs of the main convergence results of our method use a completely different philosophy from Diniz-Ehrhardt et al. (2008). To prove global convergence, we use the stationarity measure $\chi(x)$ defined by Conn et al. (2000). This measure takes into account the degree to which the directions of the steepest descent point outward with respect to the portion of the feasible region near x. On the other hand, the other main contribution of our work consists in a numerical study of different nonmonotone line search strategies. First, in order to validate our algorithm, we perform numerical experiments and comparison with another well known pattern search algorithm. Then, we extend our numerical benchmark by incorporating other line search strategies, which were initially proposed in the last two decades for solving unconstrained minimization problems and nonlinear systems of equations with derivative based and derivative-free methods (see Cheng and Li (2009), Nikolovski and Stojkovska (2013), Ulbrich (2001), Yu and Pu (2008), Zhang and Hager (2004)).

This paper is organized as follows: some definitions and preliminary results are given in Section 2. The new algorithm is introduced in Section 3. Convergence results are stated in Section 4. Numerical experiments and comparison are presented and analyzed in Section 5. Conclusions are given in Section 6.

Notation. We denote by $e^{(i)}$ the i-th canonical vector in $\mathbb{R}^n$ and $\|\cdot\|$ the Euclidean norm, and $\mathrm{int}(\Omega)$ as the largest open set contained in Ω.

2. Definitions and preliminary results

Now, we recall some definitions and results that are necessary in order to guarantee convergence of our method.

The following two definitions, given by Kolda et al. (2003), are the basis of the theory of convergence and they are widely used in the context of optimization. The first one refers to the cone K generated by the set of all nonnegative linear combinations of vectors of a given set. The second one includes those vectors that make an angle of 90° or more with each element of K.

Definition 2.1. *Let $D = \{v^{(1)}, v^{(2)}, \ldots, v^{(r)}\}$ be a set of r vectors in $\mathbb{R}^n$. The set D generates the cone K if*

$$K = \{u | u = \sum_{i=1}^{r} c^{(i)} v^{(i)}, c^{(i)} \geq 0, \text{ for } i = 1, 2, \ldots, r\}.$$

Definition 2.2. *The polar cone of a cone K, denoted by K°, is defined by*

$$K^\circ = \{w | w^T u \leq 0, \text{ for all } u \in K\}.$$

When minimizing a function in a feasible region, we are particularly interested in choosing search directions (descent directions at best) that improve the objective function and remain feasible at the same time. Given $x \in \Omega$, we define $K(x, \epsilon)$ as the cone generated by 0 and the outward pointing normals of the constraints within a distance ϵ of x, namely

$$\{e^{(i)} | u^{(i)} - x^{(i)} \leq \epsilon\} \cup \{-e^{(i)} | x^{(i)} - l^{(i)} \leq \epsilon\}.$$

In other words, $K(x, \epsilon)$ is generated by the normals to the faces of the feasible region within distance ϵ of x. Choosing properly the distance ϵ, the polar cone $K^\circ(x, \epsilon)$, associated to the cone $K(x, \epsilon)$, approximates the feasible region near x. Hence, the cone $K(x, \epsilon)$ is an important tool in this kind of problems. See Kolda et al. (2003) for more details.

As in the theory of methods explicitly based on derivatives, in derivative-free optimization, we need a measure that lets us know how close a point x is to be a stationary point. In this article, we adopted the following measure of stationarity, defined by Conn et al. (2000), given by

$$\chi(x) = \max_{\substack{x+\omega \in \Omega, \\ \|\omega\| \leq 1}} -\nabla f(x)^T \omega.$$

This measure $\chi(x)$ makes it possible to achieve the degree to which the direction of steepest descent is outward pointing with respect to the fraction of the feasible region near x. Conn et al. (2000) proved that if Ω is convex, χ is a continuous function, $\chi(x) \geq 0$ and $\chi(x) = 0$ if and only if x is a KKT point for the problem (1), i.e., a point that satisfy the first-order optimality condition.

Thus, showing that $\chi(x_k) \to 0$ as $k \to \infty$ establishes a global first-order convergence result, which will be one of our main objectives of this work.

3. The bound constrained nonmonotone pattern search algorithm `nmps`

Before introducing the algorithm, named `nmps`, we are going to define some algorithmic parameters and notations.

Let M be a positive integer indicating how many previous functional values will be considered on the nonmonotone line search. Besides, given an iteration k we define $m(k) = \min\{k, M-1\}$. Let $\Delta_{\text{tol}} > 0$ be the tolerance for the convergence criterion. Let $D_{\oplus}$ be a finite set of $\mathbb{R}^n$ given by the coordinate directions, that is $D_{\oplus} = \{\pm e^{(i)} | i = 1, 2, \ldots, n\}$.

Assume that $\{\eta_k\}$ is a sequence such that $\eta_k > 0$, for all $k = 0, 1, 2, \ldots$, and

$$\sum_{k=0}^{\infty} \eta_k = \eta < \infty \tag{2}$$

is a convergent series.

Suppose that $x_0 \in \Omega$ is an initial approximation to the solution and $f(x_0)$ its corresponding functional value. Let $\Delta_0 = 1 > \Delta_{\text{tol}}$ be the initial value for the step length and $K_{\max}$ be a predefined maximum number of iterations.

Algorithm 1: nmps (nonmonotone patern search algorithm)
Given x_0, M, $D_{\oplus}$, $\{\eta_k\}$, Δ_{tol}, $K_{\max}$, the following steps determine the best solution of the minimization problem:

Step 1: set $k \leftarrow 0$ and $\Delta_0 \leftarrow 1$, the initial value for the step length
Step 2: While $k < K_{\max}$, do
$f_{\max}(x_k) \leftarrow \max\{f(x_k), \ldots, f(x_{k-\min\{k,M-1\}})\} = \max_{0 \leq j \leq m(k)} \{f(x_{k-j})\}$
Step 3: If exists $d \in D_{\oplus}$: $(x_k + \Delta_k d) \in \Omega$ and

$$f(x_k + \Delta_k d) \leq f_{\max}(x_k) + \eta_k - \Delta_k^2 \tag{3}$$

Step 3a: then
$\Delta_{k+1} \leftarrow \min\{1, 2\Delta_k\}$
$x_{k+1} \leftarrow x_k + \Delta_k d$
$f(x_{k+1}) \leftarrow f(x_k + \Delta_k d)$
Step 3b: else
$x_{k+1} \leftarrow x_k$
$f(x_{k+1}) \leftarrow f(x_k)$
$\Delta_{k+1} \leftarrow \dfrac{\Delta_k}{2}$

Step 4: If $\Delta_{k+1} < \Delta_{\text{tol}}$ then finish the algorithm and declare $x_k, f(x_k)$ as the solution (Convergence criterion).

Step 5: $k \leftarrow k + 1$, and terminate while (Step 2)

Remark 1. It is worth noting that an iteration of the `nmps` algorithm is well defined due to the fact that $\eta_k > 0$ for all k assures that the condition (3) is satisfied when Δ_k is sufficiently small. Besides that, we observe that if Δ_k is small, then Δ_k^2 is much smaller.

4. Theoretical results and convergence analysis

In order to prove our main convergence result we need to demonstrate some auxiliary results. The following proposition concerns about the nonmonotonicity of $\{f(x_k)\}$.

Proposition 4.1. *If $l(k)$ is an integer such that $kM - M + 1 \leq l(k) \leq kM$, and*

$$\begin{aligned} f(x_{l(k)}) &= \max_{0 \leq j \leq M-1} \{f(x_{kM-j})\} \\ &= \max\{f(x_{kM}), f(x_{kM-1}), \ldots, f(x_{kM-M+1})\} \end{aligned}$$

then

$$f(x_{l(k+1)}) \leq f(x_{l(k)}) + \eta_{kM} + \ldots + \eta_{kM-M+1} - \Delta^2_{l(k+1)-1}. \tag{4}$$

Proof. It is analogous to that presented by Birgin et al. (2003), see Lemma 2.3. □

The next proposition is an important tool to prove global convergence of the `nmps` algorithm where the inequality of Proposition 4.1 is applied iteratively. This idea has also been introduced by Birgin et al. (2003) and we adapted it for derivative-free optimization problem (1).

Proposition 4.2. *Let $\{x_k\}$ be a sequence generated by the `nmps` algorithm then the sequence $\{f(x_k)\}_{k\in\mathbb{N}}$ is bounded below and $\lim_{k\to\infty} \Delta^2_{l(k)-1} = 0$.*

Proof. By the compactness of Ω and the continuity of f it is easy to conclude that the sequence $\{f(x_k)\}_{k\in\mathbb{N}}$ is bounded below. By applying the inequality (4) we have

$$f(x_{l(k+1)}) \leq f(x_0) + \sum_{k=0}^{\infty} \eta_k - \sum_{k=1}^{\infty} \Delta^2_{l(k)-1},$$

equivalently

$$\sum_{k=1}^{\infty} \Delta_{l(k)-1}^2 \leq f(x_0) - f(x_{l(k+1)}) + \sum_{k=0}^{\infty} \eta_k.$$

Now, since f is bounded below we have $-f(x_k) \leq -C$ for all k, and due to summability of the sequence $\{\eta_k\}$, we obtain

$$\sum_{k=1}^{\infty} \Delta_{l(k)-1}^2 < +\infty,$$

so $\lim_{k\to\infty} \Delta_{l(k)-1}^2 = 0$, as we want to prove. □

In consequence, we observe that $\lim_{k\to\infty} \Delta_{l(k)-1} = 0$, since the steps Δ_k are small enough and positives.

Now, using the sequence of index $\{l(k)\}$ defined in Proposition 4.1, we define the set of index

$$U = \{l(1) - 1, l(2) - 1, l(3) - 1, \ldots\}.$$

The following two results have been demonstrated by in Kolda et al. (2003).

Proposition 4.3. *Let $x \in \Omega$ and $\varepsilon \leq 0$, and let $K = K(x, \varepsilon)$ and $K^\circ = K^\circ(x, \varepsilon)$ for the bound constrained problem (1). Let $G_{K^\circ} \subseteq D_\oplus$ the set of generators of K°. Then, if $[-\nabla f(x)]_{K^\circ} \neq 0$, there is $d \in G_{K^\circ}$ such that*

$$\frac{1}{\sqrt{n}} \|[-\nabla f(x)]_{K^\circ}\| \leq -\nabla f(x)^T d.$$

Proposition 4.4. *Let $x \in \Omega$ and $\varepsilon \geq 0$, and let $K^\circ = K^\circ(x, \varepsilon)$ and $K = K(x, \varepsilon)$ for the bound constrained problem (1). Then*

$$\chi(x) \leq \|[-\nabla f(x)]_{K^\circ}\| + \sqrt{n}\|[-\nabla f(x)]_K\|\varepsilon.$$

Next, we present the main global convergence result of `nmps` algorithm, which is based on the results of Kolda et al. (2003) using the line search (3).

Theorem 4.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable, and suppose $\nabla f(x)$ is Lipschitz continuous with constant L, $\|\nabla f(x)\| \leq \gamma$, for all $x \in \Omega$ and $\{f(x_k)\}_{k\in\mathbb{N}}$ is bounded below. If $\{x_k\}_{k\in U}$ is the sequence generated by the `nmps` algorithm then*

$$\chi(x_k) \leq \sqrt{n}(L + \gamma)\Delta_k \ \text{ for all } \ k \in U.$$

Proof. We will consider two cases.

Case 1. If $x_k + \Delta_k d \notin \text{int}(\Omega)$ for all $d \in D_\oplus$, then $x_k + \Delta_k d$ is either on the boundary of or outside of Ω for all directions $d \in D_\oplus$.

In other words, if $l^{(i)} \leq x_k^{(i)} \leq u^{(i)}$ then $x_k^{(i)} - \Delta_k \leq l^{(i)}$ and $x_k^{(i)} + \Delta_k \geq u^{(i)}$ for all $i = 1, 2, \ldots, n$.

The last inequalities imply that if $x_k + \omega \in \Omega$, the vector ω cannot have their components greater than Δ_k, that is, $\omega^{(i)} \leq \Delta_k$ for all i. Therefore, $\|\omega\| \leq \sqrt{n}\Delta_k$. So,

$$\begin{array}{rcl}\chi(x_k) & = & \displaystyle\max_{\substack{x_k+\omega\in\Omega,\\ \|\omega\|\leq 1}} -\nabla f(x_k)^T\omega \\ & \leq & \displaystyle\max_{\substack{x_k+\omega\in\Omega,\\ \|\omega\|\leq 1}} \|\nabla f(x_k)\|\|\omega\| \\ & \leq & \|\nabla f(x_k)\|\sqrt{n}\Delta_k \\ & \leq & \sqrt{n}\gamma\Delta_k,\end{array}$$

which completes the proof for the Case 1.

Case 2. Now we suppose that there is at least $d \in D_\oplus$ such that $x_k + \Delta_k d \in \text{int}(\Omega)$. Thus, the cone $K^\circ(x_k, \Delta_k)$ is generated by all the directions $d \in D_\oplus$ such that $x_k + \Delta_k d \in \text{int}(\Omega)$. By the mean value theorem, we have that

$$f(x_k + \Delta_k d_k) - f(x_k) = \Delta_k \nabla f(x_k + \lambda_k \Delta_k d_k)^T d_k, \tag{5}$$

for some $\lambda_k \in [0, 1]$. Since $k \in U$, this implies

$$0 \leq f(x_k + \Delta_k d_k) - f_{\max}(x_k) - \eta_k + \Delta_k^2.$$

Taking into account that $-f_{\max}(x_k) \leq -f(x_k)$ and $\eta_k > 0$ for all k, we obtain

$$0 \leq f(x_k + \Delta_k d_k) - f(x_k) + \Delta_k^2. \tag{6}$$

Then, we replace (5) in (6)

$$0 \leq \Delta_k \nabla f(x_k + \lambda_k \Delta_k d_k)^T d_k + \Delta_k^2.$$

Next, we divide the last inequality by Δ_k and adding $-\nabla f(x_k)^T d_k$, we get the following inequality

$$-\nabla f(x_k)^T d_k \leq (\nabla f(x_k + \lambda_k \Delta_k d_k) - \nabla f(x_k))^T d_k + \Delta_k.$$

Using Proposition 4.3 we have

$$\frac{1}{\sqrt{n}}\|[-\nabla f(x)]_{K^\circ}\| \leq (\nabla f(x_k + \lambda_k \Delta_k d_k) - \nabla f(x_k))^T d_k + \Delta_k.$$

Then by the Cauchy-Schwarz inequality, the fact that $\|d_k\| = 1$ for all k and boundedness hypothesis of the gradient, we obtain

$$\frac{1}{\sqrt{n}}\|[-\nabla f(x)]_{K^\circ}\| \leq \|(\nabla f(x_k + \lambda_k \Delta_k d_k) - \nabla f(x_k))\| + \Delta_k \leq L\Delta_k + \Delta_k.$$

In consequence,

$$\|[-\nabla f(x)]_{K^\circ}\| \leq \sqrt{n}L\Delta_k + \sqrt{n}\Delta_k \leq \sqrt{n}L\Delta_k.$$

Finally, combining Proposition 4.4 with the above result

$$\begin{aligned}\chi(x) &\leq \|[-\nabla f(x)]_{K^\circ}\| + \sqrt{n}\|[-\nabla f(x)]_K\|\varepsilon \\ &\leq \|[-\nabla f(x)]_{K^\circ}\| + \sqrt{n}\gamma\Delta_k \leq \sqrt{n}L\Delta_k + \sqrt{n}\gamma\Delta_k\end{aligned}$$

consequently,

$$\chi(x) \leq \sqrt{n}(L+\gamma)\Delta_k,$$

and the proof is complete. □

Remark 2. It is worth mentioning that the nmps algorithm either generates a sequence of iterates with $\Delta_k \to 0$ or stops in a finite number of steps. In the first situation, $\chi(x) \to 0$ by Theorem 4.1. The other case could happen when the maximum number of iterations or the maximum number of function evaluations established is reached.

5. Numerical results

In this section we show and analyze the numerical results obtained using our algorithm. All the numerical experiments were carried out on a computer with a 2.3 GHz Intel Core i5–6200u processor (8 GB RAM). We implemented the `nmps` algorithm in `matlab R2016b 64-bit`.

To the purpose of carefully analyze the performance of our algorithm we organize our study in two parts. First, we compare the `nmps` algorithm with the `patternsearch` routine from `matlab`'s optimization toolbox, since both algorithms are based on pattern search methods. Then, we study the performance of `nmps` algorithm using different nonmonotone line search strategies.

We have selected a set of 63 bound constrained problems from the Hock and Schittkowski (1980) collection. Since this collection has only 9 bound constrained problems, we have modified other 54 problems with general constraints, extracting the linear and nonlinear constraints from each one of them. The detailed list of these problems and their characteristics is given in Table 1.

Table 1: Characteristics of test problems

Prob.	N°. HS	n	bound constraints	objective function
1	1	2	1	Generalized polynomial
2	2	2	1	Generalized polynomial
3	3	2	1	Generalized polynomial
4	4	2	2	Generalized polynomial
5	5	2	4	General
6	25	3	6	Sum of squares
7	38	4	8	Generalized polynomial
8	45	5	10	Constant
9	110	10	20	General
10	13	2	2	Quadratic
11	15	2	1	Generalized polynomial
12	16	2	3	Generalized polynomial
13	17	2	3	Generalized polynomial
14	18	2	4	Quadratic
15	19	2	4	Generalized polynomial
16	20	2	2	Generalized polynomial
17	21	2	4	Quadratic
18	23	2	4	Quadratic
19	24	2	2	Generalized polynomial
20	30	3	6	Quadratic
21	31	3	6	Quadratic
22	32	3	3	Quadratic
23	33	3	4	Generalized polynomial
24	34	3	6	Linear
25	35	3	3	Quadratic
26	36	3	6	Generalized polynomial
27	37	3	6	Generalized polynomial
28	41	4	8	Generalized polynomial
29	42	4	2	Quadratic
30	44	4	4	Quadratic
31	53	5	10	Quadratic
32	54	6	12	General
33	55	6	8	General
34	57	2	2	Sum of squares
35	59	2	4	General
36	60	3	6	Generalized polynomial
37	62	3	6	General
38	63	3	3	Quadratic
39	64	3	3	Generalized polynomial
40	65	3	6	Quadratic
41	66	3	6	Linear
42	68	4	8	General
43	69	4	8	General
44	71	4	8	Generalized polynomial
45	72	4	8	Linear
46	73	4	4	Linear
47	74	4	8	Generalized polynomial
48	75	4	8	Generalized polynomial
49	76	4	4	Quadratic
50	80	5	10	General
51	81	5	10	General
52	83	5	10	Quadratic
53	84	5	10	Quadratic
54	86	5	5	Generalized polynomial
55	93	6	6	Generalized polynomial
56	101	7	14	Generalized polynomial
57	102	7	14	Generalized polynomial
58	103	7	14	Generalized polynomial
59	104	8	16	Generalized polynomial
60	106	8	16	Linear
61	108	9	2	Quadratic
62	114	10	20	Quadratic
63	119	16	32	Generalized polynomial

As it is usual in derivative-free optimization literature, we are interested in the number of functional values needed to satisfy the stopping criteria, which are: reaching a sufficiently small step length ($\Delta_k < \Delta_{\mathrm{tol}}$), attaining the maximum number of function evaluations MaxFE or attaining the maximum number of iterations MaxIt. We adopt the convergence test proposed by Moré and Wild (2009) to measure the ability of an algorithm to improve an initial approximation and to declare that a problem has been solved if the condition

$$f(x_0) - f(x) \geq (1 - \tau)(f(x_0) - f_{\rm L}) \tag{7}$$

holds where x_0 is the initial feasible approximation, $\tau > 0$ is the level of accuracy and $f_{\rm L}$ is the smallest functional value obtained among the considered solvers. We use the performance profile graphs (see Dolan and Moré (2002), Moré and Wild (2009)) to illustrate the results obtained with (7).

We used the same initial approximation x_0 as Hock and Schittkowski (1980), projecting onto the bound constraints if the initial approximation was not feasible. After some preliminary tests we adopted $\eta_k = 1.1^{-k}$ for all k. Furthermore, the following algorithmic parameters were set: $\Delta_0 = 1.0$, as the initial step length, $M = 15$, MaxFE $= 2500$, MaxIt $= 5000$ and $\Delta_{\rm tol} = {\rm TOL} = 10^{-6}$.

It is worth mentioning two important implementation details of our algorithm. First, the function f is evaluated in all possible coordinate directions and the accepted new approximation is the one that produces the minimum functional value of $f(x_k + \Delta_k d)$. Second, the accepted points are stored in memory for the purpose to avoid revisiting older points without slowing down the implementation. In other words, because of the characteristics of the pattern search algorithms, the search could revisit points increasing the number of function evaluations considerably. Therefore, a new point x will not be evaluated if $\|x - x_{\rm old}\| < 10^{-8}\|x\|$, where the point $x_{\rm old}$ has already been evaluated in previous iterations. For more details, see Section 8.5 in Lewis et al. (2007).

5.1 Numerical benchmark between `nmps` and `patternsearch`

We tested our algorithm `nmps` using the set of test problems and we compared it with the well established routine `patternsearch` from `matlab`. Since both codes are based on a pattern search scheme, we set the same algorithmic parameters. The numerical results are shown in Tables 2–3. In these tables, the numbers 5000 and 2500 in columns *It.* and *FE.*, respectively, mean that maximum number of Iterations or the maximum number of Function Evaluations has been reached. Otherwise, the convergence criterion has been attained.

Table 2: Results of numerical experiments

Prob.	nmps		patternsearch		λpatternsearch		Cpatternsearch		nmps-M1	
	It.	*FE.*	*It.*	*FE.*	*It.*	*FE.*	*It.*	*FE.*	*It.*	*FE.*
1	341	352	625	2500	339	352	530	352	1003	2500
2	402	323	58	207	5000	6	628	281	277	155
3	277	83	39	119	1660	2500	307	83	1659	2500
4	277	46	34	84	5000	14	307	46	277	48
5	293	305	36	145	269	337	305	314	269	336
6	700	2500	513	2500	5000	1330	670	2500	552	2500
7	1124	2500	176	1409	5000	177	974	2500	455	2500
8	278	219	50	290	5000	134	307	132	278	248
9	262	2500	129	2500	141	2500	211	2500	141	2500
10	278	64	26	77	278	64	307	64	278	64
11	277	158	625	2500	5000	106	306	156	277	539
12	278	150	93	343	5000	98	306	139	278	143
13	278	150	93	343	5000	98	306	139	278	143
14	278	45	37	92	5000	17	307	45	278	45
15	278	77	48	133	278	77	5000	77	278	77
16	278	151	93	345	5000	99	306	140	278	144
17	277	63	21	64	277	63	307	63	277	63
18	288	260	34	137	277	289	305	314	270	369
19	834	2500	627	2500	834	2500	834	2500	834	2500
20	278	108	26	131	5000	23	307	108	278	108
21	278	106	26	128	5000	21	307	106	278	106
22	277	76	47	183	277	76	306	76	277	82
23	277	712	37	129	277	708	307	791	277	704
24	278	480	128	747	278	480	307	480	278	480
25	289	480	130	774	278	508	305	522	273	625
26	281	192	95	402	281	192	5000	192	281	192
27	5000	443	34	86	5000	443	5000	437	5000	443
28	278	81	20	81	5000	13	307	81	278	81
29	278	146	43	259	5000	44	307	146	278	146
30	625	2500	418	2500	625	2500	625	2500	625	2500
31	288	800	39	391	277	939	306	965	270	1130
32	278	13	247	2500	278	13	307	287	278	13
33	277	1818	51	398	384	2500	306	2011	277	1810
34	280	140	61	191	5000	77	307	120	278	134
35	278	326	119	410	277	320	307	335	277	509
36	403	737	125	751	294	746	322	755	278	744
37	285	287	48	210	284	122	5000	275	284	122
38	500	2500	417	2500	500	2500	500	2500	500	2500
39	392	1501	325	1945	5000	1135	5000	1447	634	2500
40	278	106	70	297	5000	61	306	106	278	106
41	278	592	140	782	278	750	307	592	420	1842
42	277	1241	46	288	277	1234	307	1377	278	1238
43	282	1373	48	299	278	1239	445	2069	278	1226
44	278	110	53	252	5000	38	307	110	278	110
45	278	107	76	431	5000	39	306	107	278	107
46	278	100	66	352	278	100	306	100	278	100
47	278	121	20	117	278	121	307	107	278	121
48	278	106	20	113	278	106	307	104	278	106
49	278	195	60	426	278	195	306	195	277	177
50	278	457	74	484	278	457	307	181	278	457
51	287	538	46	396	364	2500	318	663	361	2500
52	284	1289	20	101	284	1289	5000	155	284	1289
53	405	2500	625	2500	405	2500	405	2500	405	2500
54	602	1443	85	826	249	1016	450	1275	252	1045
55	277	1856	106	824	278	1861	307	2055	277	1849

(continue on the next page)

Table 3: Results of numerical experiments (cont.)

Prob.	nmps		patternsearch		λpatternsearch		Cpatternsearch		nmps-M1	
	$It.$	$FE.$	$It.$	$FE.$	$It.$	$FE.$	$It.$	$FE.$	$It.$	$FE.$
56	357	2500	233	2500	5000	547	361	2500	247	2500
57	376	2500	233	2500	5000	547	347	2500	255	2500
58	376	2500	232	2500	5000	964	345	2500	257	2500
59	278	621	100	1294	213	2500	306	621	215	2500
60	167	2500	157	2500	167	2500	167	2500	167	2500
61	167	2500	157	2500	167	2500	167	2500	167	2500
62	149	2500	150	2500	149	2500	149	2500	149	2500
63	278	2446	104	2500	5000	2142	307	2446	278	2446

In Figure 1 we show the performance profile pictures using condition (7) with three levels of accuracy: $\tau = 10^{-1}$, $\tau = 10^{-3}$ and $\tau = 10^{-5}$, where a smaller value of τ means the satisfaction of condition (7) is more strict. In a performance profile graph, the top curve represents the most efficient method within a factor τ of the best measure.

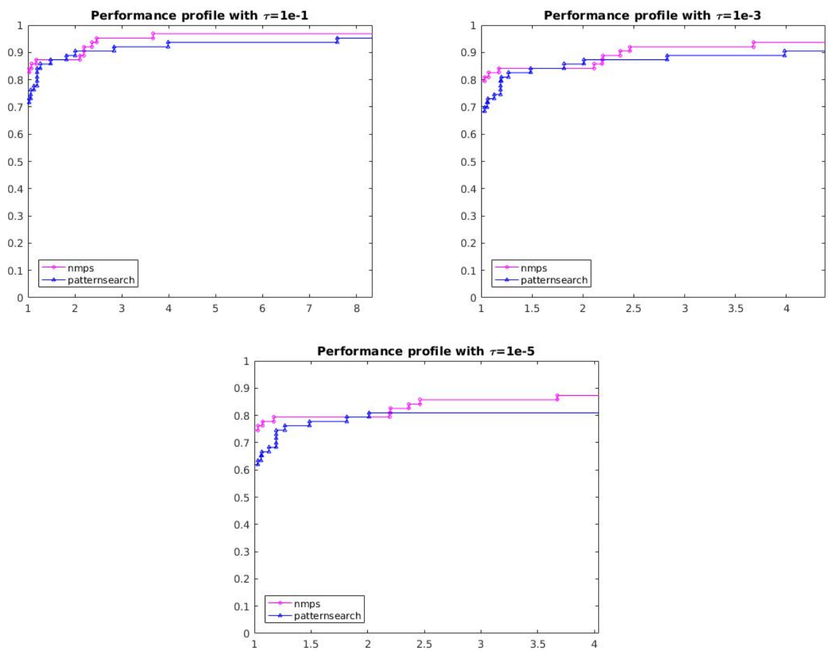

Figure 1: Performance profiles of nmps and patternsearch.

In Figure 1, with $\tau = 10^{-1}$, we observe that `nmps` algorithm attains the best metric value (in this case the minimum number of objective function evaluations to convergence with the given accuracy) in 82% of the set problems while `patternsearch` does it in 71%. We also see, within a factor of 1.7 of the best solver, both algorithms have a similar behavior and the performance profile shows these algorithms can solve a problem with a probability of 0.87 with respect to the best solver. If the goal is to solve efficiently 95% of the problems, the `nmps` algorithm accomplishes this by using 2.4 times the minimum number of function evaluations while `patternsearch` needs a factor of 7.5.

Now, using $\tau = 10^{-3}$ (Figure 1), we note that `nmps` wins in the 79% of the problems in comparison with the 68% of `patternsearch`. Also, both solvers are equivalent if the solution is required within a factor of 1.7 of the best solver, with a probability of 0.84. Although both solvers can reach the solution in, at most, 95% of the problems, `nmps` was closer than the other since it solved 94% of them, using 3.7 times the minimum number of function evaluations.

Finally in Figure 1, we observe that the performance of both solvers deteriorates using $\tau = 10^{-5}$. In any case, the algorithm `nmps` wins in 74% of the problems meanwhile `patternsearch` wins in 62%. As before, `nmps` algorithm performs better than `patternsearch` if one consider a solver that finds the solution using 1.7 times the minimum number of function evaluations, with a probability of 0.8. In such a case, we could expect at most that `nmps` algorithm solves 87% of the problems while `patternsearch` solves 81%.

We conclude that, regardless the level of accuracy, `nmps` outperforms the `patternsearch` routine in the set of test problems considered. In fact, `nmps` always has a probability 10% greater than `patternsearch` to get the solution.

Next, we will analyze the performance of our algorithm using other line search strategies.

5.2 Numerical study using other line search strategies

Recently, some authors proposed different nonmonotone line search strategies for solving unconstrained minimization problems and nonlinear systems of equations, with derivatives-based and derivative-free methods. We decided to adopt them for derivative-free optimization problem (1).

Originally Zhang and Hager (2004) proposed a nonmonotone strategy for unconstrained optimization problems with derivatives, which guarantees that an average of the succesive functional values is decreasing. Later, Cheng and Li

(2009) combined this approach with the ideas from Li and Fukushima (2000) to solve derivative-free nonlinear systems of equations. Finally, Nikolovski and Stojkovska (2013) and Krejić et al. (2015) applied the last strategy for solving derivative-free unconstrained optimization problems. This is the first approach that we used for numerical comparison. It was called *C-line search* and we implemented it in `Cpatternsearch` algorithm. This scheme is similar to the nonmonotone line search (3), where $f_{\max}(x_k)$ is replaced by the sequence $\{C_k\}$ given by

$$Q_{k+1} = r_k Q_k + 1, \qquad C_{k+1} = \frac{r_k Q_k (C_k + \eta_k) + f_{k+1}}{Q_{k+1}} \tag{8}$$

with $Q_0 = 1$, $C_0 = f(x_0)$, $r_k \in [0,1]$ and the sequence $\{\eta_k\}$ satisfying $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$ for all $k = 0, 1, 2, \dots$

The idea behind the second approach considered by us came up in an article published by Ulbrich (2001), where a trust-region method for bound constrained semismooth systems of equations was proposed. Ulbrich reformulated it in a constrained differentiable minimization problem. After, this scheme motivated Yu and Pu (2008) to propose a nonmonotone technique for differentiable unconstrained minimization. Then Nikolovski and Stojkovska (2013) adapted this rule for solving derivative-free unconstrained optimization problems and it was taken by us for our numerical experiments. It was named *λ-line search* and we implemented it in λ`patternsearch` algorithm. It is also analogous to (3) but in this case $f_{\max}(x_k)$ is defined by

$$f_{\max}(x_k) = \max\{f(x_k), \sum_{r=0}^{m(k)-1} \lambda_{k_r} f(x_{k-r})\} \tag{9}$$

with $M \in \mathbb{N}$, $m(k) = \min\{k, M-1\}$, $\lambda_{k_r} \geq \lambda$ and $\sum_{r=0}^{m(k)-1} \lambda_{k_r} = 1$ for all k.

Finally, we have considered the special case of Algorithm `nmps` with $M = 1$, i. e., we have replaced $f_{\max}(x_k)$ by $f(x_k)$, obtaining

$$f(x_k + \Delta_k d) \leq f(x_k) + \eta_k - \Delta_k^2, \tag{10}$$

where the sequence $\{\eta_k\}$ and the step length Δ_k are defined as in Algorithm `nmps`. It was implemented in `nmps-M1` algorithm.

Again, we chose $\eta_k = 1.1^{-k}$ for all k, for the three new conditions. Also, we adopted $M = 15$ and $\lambda_{k_r} = 1/m(k)$ for all r in λ`patternsearch` and $r_k = 0.85$ for all k in `Cpatternsearch`.

Next, we show the performance profiles with convergence test (7) for solving our set of test problems using `nmps`, `patternsearch`, `Cpatternsearch`, λ`patternsearch` and `nmps-M1` algorithms. The numerical results are also presented in Tables 2–3.

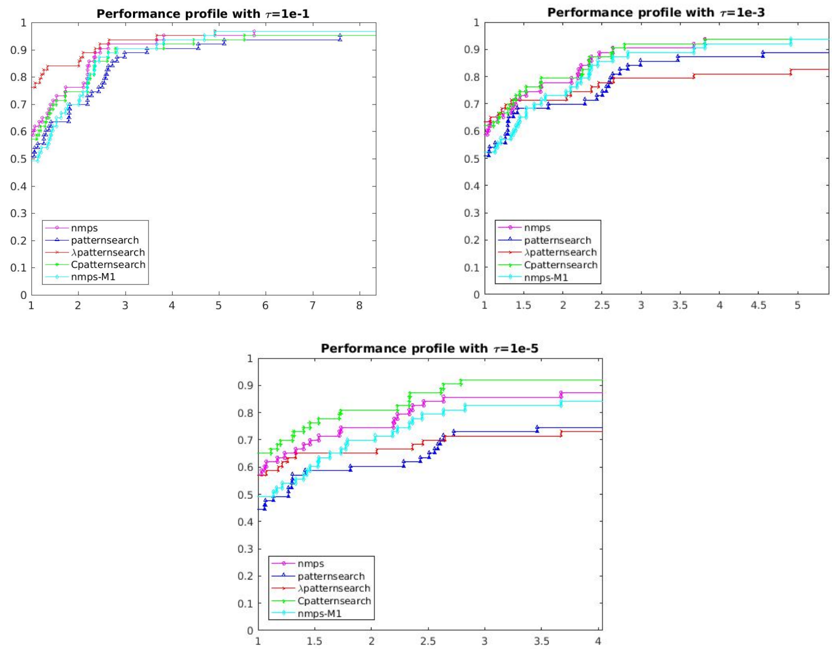

Figure 2: Performance profiles for the five line search strategies

In Figure 2, for $\tau = 10^{-1}$, we see that λ`patternsearch` attains the best performance in 76% of the problems, followed by `nmps` with 59% and `Cpatternsearch` with 57%, below we find `patternsearch` and `nmps-M1` with 56% and 51%, respectively. We observe, within a factor of 2.5 of the best solver, λ`patternsearch` reaches the greater probability of solving a problem (around 0.92) and `nmps` follows it with 0.9. Furthermore, in this case, `patternsearch` is the solver with the lowest performance, with a probability of 0.76. Moreover, with this level of accuracy, we note that `nmps-M1` solves almost 97% of the problems using a factor of 4.9 times the minimum number of function evaluations, meanwhile `nmps` has the same behavior requiring a factor of 5.75.

Figure 2 for $\tau = 10^{-3}$, increasing in this way the level of accuracy. Once more, λ`patternsearch` attains the minimum number of objective function evaluations with a probability of 0.63, followed by Cpatternsearch and nmps with a probability of 0.62 and 0.6, respectively. In the last positions, we find nmps-M1 and patternsearch with 0.52 and 0.51, respectively. Now, within a factor of 2.5 of the best solver, nmps exhibits the best performance solving 89% of the problems. Later, Cpatternsearch and nmps-M1 get 87% and 85% on the resolution of our set of problems. With this level of accuracy, the most we can expect is to solve 94% of the problems within a factor of 3.8 of the best solver, as we see in the behavior of Cpatternsearch and nmps algorithms.

Finally, Figure 2 for $\tau = 10^{-5}$, requiring in this way a greater descent in the objective function. We observe that Cpatternsearch wins in 65% of the problems, followed by nmps and λpatternsearch with 57%, nmps-M1 with 49% and patternsearch with 44%. We see, within a factor of 2.5 of the best solver, Cpatternsearch obtains the higher probability for solving a problem (0.87), followed by nmps (0.84). In the last picture in Figure 2, we can see how the performance of the different methods are slightly separated from each other. Also, Cpatternsearch algorithm is always on top of all the remaining solvers. In this case, Cpatternsearch achieves the solution in 92% of the problems within a factor of 2.79 followed by nmps that solves 87% of the problems using 3.6 times the minimum number of function evaluations. The other solvers can only solve at most 80% of the problems employing greater factors.

From the analysis of Figure 2 we obtain the following conclusions. First, in most cases the use of a nonmonotone line search of the kind (3), (8) or (9) turns into an advantage in the performance of the algorithms compared to the nmps-M1 (10). In other words, a greater effort devoted to building $f_{\max}(x_k)$ or C_k results in a decrease in the number of function evaluations carried out by the algorithm, which is one of the main goals in derivative-free methods. Second, although the strategies (3) and (9) have similar definitions, the above analysis shows, while λpatternsearch reduces its performance as the level of accuracy increases, nmps remains stable for all values of τ. So, if we should choose between this two strategies, nmps would be the most suitable. Third, the Cpatternsearch algorithm could be considered as the solver with the best performance because it is always above the other solvers as the level of accuracy increases. Our algorithm nmps, on its turn, seems to be a good competitor to Cpatternsearch, attaining a similar performance with regards to the latter in several cases. Besides, nmps always obtains the second place regarding the probability to solve all the problems. At the end, to our surprise, we observe that the performance of patternsearch algorithm is in many cases below all remaining methods.

6. Conclusions

We have proposed a coordinate search algorithm, `nmps`, to solve bound constrained optimization problems, which uses a nonmonotone line search strategy. We have proved that, under mild assumptions, we can guarantee global convergence of our method to a KKT point. This result is strongly based on the relationship between the step length and the stationarity measure (defined by Conn et al. (2000)). Moreover, we have carried out exhaustive numerical experiments in order to validate our algorithm. First, we have compared the `nmps` algorithm with the routine `patternsearch` from `matlab`. Second, we have extended our numerical study using other line search strategies that were originally proposed for different optimization approaches in the last two decades. We have implemented and adapted it to our bound constrained optimization problem. The benchmark results were satisfactory on the set of problems considered, so we can conclude that the `nmps` algorithm is competitive in comparison with the other solvers considered in this work, as the numerical experiments and performance profiles reveal.

Acknowledgement

This work was partially supported by research project Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET) PIP 112-201101-00050, PUE 22920160100134CO and Secretaría de Ciencia y Tecnología (SECyT-UNC).

References

Alberto, P., Nogueira, F., Rocha, H., and Vicente, L. N. (2004). Pattern search methods for user-provided points: Application to molecular geometry problems. *SIAM Journal on Optimization*, 14(4):1216–1236.

Arouxét, M. B., Pilotta, E. A., and Echebest, N. (2011). Active-set strategy in Powell's method for optimization without derivatives. *Computational & Applied Mathematics*, 30(1):171–196.

Audet, C. and Dennis Jr, J. E. (2002). Analysis of generalized pattern searches. *SIAM Journal on Optimization*, 13(3):889–903.

Audet, C. and Hare, W. (2017). *Derivative-free and blackbox optimization*. Switzerland: Springer.

Birgin, E., Martínez, J. M., and Raydán, M. (2003). Inexact spectral projected gradient methods on convex sets. *IMA Journal of Numerical Analysis*, 23(4):539–559.

Cheng, W. and Li, D. H. (2009). A derivative-free nonmonotone line search and its application to the spectral residual method. *IMA Journal of Numerical Analysis*, 29(3):814–825.

Conn, A. R., Gould, N. I., and Toint, P. L. (2000). *Trust region methods*. Philadelphia, USA: Society for Industrial and Applied Mathematics.

Conn, A. R., Scheinberg, K., and Vicente, L. N. (2009). *Introduction to derivative-free optimization*. Philadelphia, USA: Society for Industrial and Applied Mathematics.

Diniz-Ehrhardt, M. A., Ferreira, D. G., and Santos, S. A. (2019). A pattern search and implicit filtering algorithm for solving linearly constrained minimization problems with noisy objective functions. *Optimization Methods and Software*, 34(4):827–852.

Diniz-Ehrhardt, M. A., Martínez, J. M., and Raydán, M. (2008). A derivative-free nonmonotone line-search technique for unconstrained optimization. *Journal of Computational and Applied Mathematics*, 219(2):383–397.

Dolan, E. D. and Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical Programming*, 91(2):201–213.

Gardeux, V., Omran, M. G., Chelouah, R., Siarry, P., and Glover, F. (2017). Adaptive pattern search for large-scale optimization. *Applied Intelligence*, 47(2):319–330.

Grippo, L., Lampariello, F., and Lucidi, S. (1986). A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, 23(4):707–716.

Hock, W. and Schittkowski, K. (1980). Test examples for nonlinear programming codes. *Journal of Optimization Theory and Applications*, 30(1):127–129.

Hooke, R. and Jeeves, T. A. (1961). Direct search solution of numerical and statistical problems. *Journal of the ACM*, 8(2):212–229.

Kolda, T. G., Lewis, R. M., and Torczon, V. (2003). Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482.

Krejić, N., Lužanin, Z., Nikolovski, F., and Stojkovska, I. (2015). A nonmonotone line search method for noisy minimization. *Optimization Letters*, 9(7):1371–1391.

La Cruz, W., Martínez, J. M., and Raydán, M. (2006). Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. *Mathematics of Computation*, 75(255):1429–1448.

Lee, K., Kim, M., Ha, M. Y., and Min, J. K. (2018). Investigation of heat-exchanger-sizing methods using genetic, pattern search, and simulated annealing algorithms and the effect of entropy generation. *Journal of Mechanical Science and Technology*, 32(2):915–928.

Lewis, R. M., Shepherd, A., and Torczon, V. (2007). Implementing generating set search methods for linearly constrained minimization. *SIAM Journal on Scientific Computing*, 29(6):2507–2530.

Lewis, R. M. and Torczon, V. (1999). Pattern search algorithms for bound constrained minimization. *SIAM Journal on Optimization*, 9(4):1082–1099.

Li, D. H. and Fukushima, M. (2000). A derivative-free line search and global convergence of broyden-like method for nonlinear equations. *Optimization Methods and Software*, 13(3):181–201.

Lucidi, S. and Sciandrone, M. (2002). On the global convergence of derivative-free methods for unconstrained optimization. *SIAM Journal on Optimization*, 13(1):97–116.

Meza, J. C. and Martínez, M. L. (1994). Direct search methods for the molecular conformation problem. *Journal of Computational Chemistry*, 15(6):627–632.

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

Nikolovski, F. and Stojkovska, I. (2013). New derivative-free nonmonotone line search methods for unconstrained minimization. *In: Proceedings of the Fifth International Scientific Conference-FMNS2013*, pages 47–53.

Oeuvray, R. and Bierlaire, M. (2007). A new derivative-free algorithm for the medical image registration problem. *International Journal of Modelling and Simulation*, 27(2):115–124.

Rashid, H., Siam, F. M., Maan, N., and Rahman, W. N. W. A. (2018). Parameter estimation for a model of ionizing radiation effects on targeted cells using genetic algorithm and pattern search method. *MATEMATIKA: Malaysian Journal of Industrial and Applied Mathematics*, 34(3):1–13.

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

Ulbrich, M. (2001). Nonmonotone trust-region methods for bound-constrained semismooth equations with applications to nonlinear mixed complementarity problems. *SIAM Journal on Optimization*, 11(4):889–917.

Yu, Z. and Pu, D. (2008). A new nonmonotone line search technique for unconstrained optimization. *Journal of Computational and Applied Mathematics*, 219(1):134–144.

Zhang, H. and Hager, W. W. (2004). A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal on Optimization*, 14(4):1043–1056.