INTERNATIONAL JOURNAL OF COMPUTER MATHEMATICS
2019, VOL. 96, NO. 1, 33–50
https://doi.org/10.1080/00207160.2017.1413552

Taylor & Francis
Taylor & Francis Group

ARTICLE

Check for updates

A non-monotone pattern search approach for systems of nonlinear equations

Keyvan Amini[a], Morteza Kimiaei[b] and Hassan Khotanlou[c]

[a]Department of Mathematics, Faculty of Science, Razi University, Kermanshah, Iran; [b]Faculty of Mathematics, University of Vienna, Vienna, Austria; [c]Department of Computer Engineering, Bu-Ali Sina University, Hamedan, Iran

ABSTRACT

In this paper, a new pattern search is proposed to solve the systems of nonlinear equations. We introduce a new non-monotone strategy which includes a convex combination of the maximum function of some preceding successful iterates and the current function. First, we produce a stronger non-monotone strategy in relation to the generated strategy by Gasparo *et al.* [*Nonmonotone algorithms for pattern search methods*, Numer. Algorithms 28 (2001), pp. 171–186] whenever iterates are far away from the optimizer. Second, when iterates are near the optimizer, we produce a weaker non-monotone strategy with respect to the generated strategy by Ahookhosh and Amini [*An efficient nonmonotone trust-region method for unconstrained optimization*, Numer. Algorithms 59 (2012), pp. 523–540]. Third, whenever iterates are neither near the optimizer nor far away from it, we produce a medium non-monotone strategy which will be laid between the generated strategy by Gasparo *et al.* [*Nonmonotone algorithms for pattern search methods*, Numer. Algorithms 28 (2001), pp. 171–186] and Ahookhosh and Amini [*An efficient nonmonotone trust-region method for unconstrained optimization*, Numer. Algorithms 59 (2012), pp. 523–540]. Reported are numerical results of the proposed algorithm for which the global convergence is established.

ARTICLE HISTORY
Received 11 June 2016
Revised 9 August 2017
Accepted 19 October 2017

KEYWORDS
Nonlinear equation; pattern search; coordinate search; non-monotone technique; theoretical convergence

2010 AMS SUBJECT CLASSIFICATIONS
90C30; 93E24; 34A34

1. Introduction

Consider the following nonlinear system of equations

$$F(x) = 0, \quad x \in \mathbb{R}^n, \tag{1}$$

for which $F : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable mapping. Suppose that $F(x)$ has a zero. Then every solution x^* of the nonlinear equation problem (1) is a solution of the following nonlinear unconstrained least-squares problem

$$\begin{aligned} \min \quad & f(x) := \frac{1}{2}\|F(x)\|^2 \\ \text{s.t.} \quad & x \in \mathbb{R}^n, \end{aligned} \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm. Conversely, if x^* solves Equation (2) and $f(x^*) = 0$, then x^* is a solution of (1). There are variant methods to solve nonlinear system (1), as conjugate gradient methods [33,35], line-search methods [5,12,14–16,32] and trust-region methods [2,3,7–11,34,36,37,39],

CONTACT Morteza Kimiaei kimiaeim83@univie.ac.at Faculty of Mathematics, University of Vienna, Oskar-Morgenstern-Platz 1, A-1090 Vienna, Austria

which are quite fast and robust; but they may have some shortcomings. First, by a ratio, trust-region algorithm tries to control the agreement between the actual and predicted reduction essentially only along with a direction, for more details on the trust-region algorithm, cf. [26] If this ratio is near one and the Jacobin matrix $\nabla F(x)$ is ill-conditioned or f is a highly nonlinear function for which approximated quadratic is not good, then the trust-region radius may increase before reaching a narrow curved valley. Afterwards, we need to reduce several times the radius to get around this narrow curved valley that leads to increase computational cost and also produce unsuitable solution for the cases in which highly accurate solutions are necessary. Second, solving the trust-region subproblems leads to increase CPU times. Third, these methods need to compute both $\nabla F(x)$ and $\nabla F(x)^{\mathrm{T}}\nabla F(x)$ to determine the approximated quadratic in each iteration. Pattern search methods represent a derivative free subclass of direct search algorithms to minimize a continuous function (see, e.g. [4,17,21,22]). Box [4] and Hooke and Jeeves [17] were the first researchers to introduce the original pattern search methods. Some researchers have shown that pattern search algorithms converge globally, see [13,19,20,30,31]. Lewis and Torczon successfully extended these algorithms to obtain bound and linearly constrained minimization [19,20]. Torczon [29,30] presented a multidirectional search algorithm for parallel machines. In ill-conditioned problems, using monotone pattern search auxiliary algorithm may have unsuitable influence on the performance of the whole procedure, cf. [13]. Hence, we are going to introduce a new non-monotone pattern search framework that decreases the total number of function evaluations and CPU times. This development enables us to produce a suitable non-monotone strategy at each iteration and maintains the global convergence. Numerical results show that the new modification of pattern search is efficient to solve systems of nonlinear equations.

Notation: The Euclidean vector norm or the associated matrix norm is denoted by the symbol $\|\cdot\|$. A set of directions $\{d_k^1, \dots, d_k^p\}$ is called positively span $\mathbb{R}^n$ if for each $y \in \mathbb{R}^n$ there exist $\lambda_i \geq 0$, for $i = 1, \dots, p$, such that

$$y = \sum_{j=1}^{p} \lambda_i d_k^i.$$

Moreover, e_i, for $i = 1, \dots, p$, is considered as the orthonormal set of the coordinate directions. To simplify our notation, we set $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$.

Organization. The rest of this paper is organized as follows. In Section 2, we first describe the exploratory moves and then the generalized pattern search is presented. A new non-monotone pattern search algorithm is presented in Section 3. In Section 4, the global convergence of the new algorithm is investigated. Numerical results are provided in Section 5 to show that the proposed algorithm is efficient and promising for systems of nonlinear equations. Finally, some concluding remarks are given in Section 6.

2. The generalized pattern search method

First of all, we define two components, namely a basis matrix and a generating matrix, cf. [31].

Definition 2.1: Any arbitrary non-singular matrix $B \in \mathbb{R}^{n\times n}$ is called a basis matrix.

Definition 2.2: The generating matrix $C_k \in \mathbb{Z}^{n\times p}$ with $p > 2n$, divided into two parts, is considered as

$$C_k := [\Gamma_k \quad L_k],$$

in which $\Gamma_k := [M_k \quad -M_k]$, $M_k \in M \subset \mathbb{Z}^{n\times n}$, M is a finite set of non-singular matrices and $L_k \in \mathbb{Z}^{n\times(p-2n)}$ is a matrix that contains at least a column zeros.

Defined by the columns of the matrix $P_k = BC_k$, in which B is a basis matrix, is a pattern P_k. By the definition C_k and this fact that M_k has rank n, it is clear that C_k also has rank n. This fact implies that the columns of P_k span $\mathbb{R}^n$. To act better, the partition of the generating matrix C_k to partition P_k is used, as follows:

$$P_k := BC_k = [B\Gamma_k \quad BL_k]. \tag{3}$$

Given x_k, a step-size $\Delta_k > 0$, we define a trial step d_k^i to be any vector of the form

$$d_k^i := \Delta_k B c_k^i, \quad \forall\, i = 1, \ldots, p,$$

in which c_k^i indicates the ith column of C_k, the vectors Bc_k^i, named exploratory moves as proposed in [31], determine the step directions and Δ_k is considered as a step-size parameter. Furthermore, a trial point as any point of the form $x_k^i := x_k + d_k^i$ will be defined, where x_k is the current iterate. Before declaring a new iterate and updating the associated information, pattern search methods use the series of exploratory moves in order to produce the new iterate. To prove the convergence property of pattern search methods, we require that the exploratory moves are obtained by the following two procedures:

Procedure 1(Weak monotone hypotheses on exploratory moves)

(S.1) Compute P_k by Equation (3) and then choose $d_k \in \Delta_k P_k$.
(S.2) If $\min\{f(x_k + y),\ y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + d_k) < f(x_k)$; **stop**. Otherwise, go to (S.1).

In Procedure 1, note that $y \in A$ means that the vector y is contained in the set of columns of the matrix A. (S.2) is more interesting; hence, let us describe how it works. As long as there exists a decrease on the function value at each iterate among any of the $2n$ steps presented by $\Delta_k B\Gamma_k$, the exploratory moves must return a decrease step on the function value at each iterate, without satisfying $f(x_k + d_k) \le \min\{f(x_k + y),\ y \in \Delta_k B\Gamma_k\}$.

Procedure 2(Strong monotone hypotheses on exploratory moves)

(S.1) Compute P_k by Equation (3) and then choose $d_k \in \Delta_k P_k$.
(S.2) If $\min\{f(x_k + y),\ y \in \Delta_k B\Gamma_k\} < f(x_k)$, then $f(x_k + d_k) \le \min\{f(x_k + y),\ y \in \Delta_k B\Gamma_k\}$; **stop**. Otherwise, go to (S.1).

In Procedure 2, (S.2) is replaced by a strong version, as presented above.

Algorithm 1 situates the generalized pattern search method for systems of nonlinear equations, cf. [31].

In Algorithm 1, if $\rho_k > 0$ (Line 8), then it is called a ***successful iteration***. Otherwise, it is called a ***unsuccessful iteration***. The parameter θ is considered the ***shrinkage parameter*** with the role $\theta := \tau^{w_0}$, in which $\tau > 1$ and w_0 is a negative integer, and λ_k is called the ***expanding factor*** such that

$$\lambda_k \in \{\tau^{w_1}, \tau^{w_2}, \ldots, \tau^{w_l}\},$$

in which $w_1, w_2, \ldots, w_l$ are positive integers, with $l < \infty$. In Line 4 of this algorithm, the step d_k can be obtained either by Procedure 1 or Procedure 2. This algorithm is called ***generalized weak pattern search*** (***GWPS***) if d_k is obtained by Procedure 1; otherwise, if d_k is obtained by Procedure 2, it is called ***generalized strong pattern search*** (***GSPS***).

Algorithm 1: GPS (Generalized Pattern Search)

Input: An initial point $x_0 \in \mathbb{R}^n$, an initial integer matrix $C_0 \in \mathbb{Z}^{n\times p}$, $\epsilon > 0, \theta \in (0,1), \lambda_k \geq 1$ and $\Delta_0 > 0$;
Output: x_b, f_b;
begin
  $k := 0; F_k := F(x_k); f_k := \frac{1}{2}\|F_k\|^2$;
  while $\Delta_k \geq \epsilon$ **do**
    choose d_k by either Procedure 1 or Procedure 2; compute $F(x_k + d_k)$;
    $f(x_k + d_k) := \frac{1}{2}\|F(x_k + d_k)\|^2$;
    compute $\rho_k := f_k - f(x_k + d_k)$;
    if $\rho_k > 0$ **then**
      $x_{k+1} := x_k + d_k$; $\Delta_{k+1} := \lambda_k \Delta_k$;
    else
      $x_{k+1} := x_k$; $\Delta_{k+1} := \theta\Delta_k$;
    end
    $F_{k+1} := F(x_{k+1}); f_{k+1} := \frac{1}{2}\|F_{k+1}\|^2$;
    update C_{k+1};
    $k \leftarrow k+1$;
  end
  $x_b := x_k$; $f_b := f_k$;
end

Both GWPS and GSPS contain a drawback. This fact that the quantity ρ_k can not truly prevent the production of unsuccessful iterations in the presence of narrow curved valley leads to the increase of CPU time and the total number of function evaluations. In order to overcome this drawback, Gasparo *et al.* [13] modified the quantity ρ_k.

Torczon [31] showed in Theorem 3.2 that each iterate x_n generated by GWPS can be considered as

$$x_n := x_0 + (\beta^{r_{\text{LB}}}\alpha^{-r_{\text{UB}}})\Delta_0 B \sum_{k=0}^{n-1} z_k, \tag{4}$$

in which α and β are relatively prime positive integers while satisfying $\beta/\alpha := \tau$, $r_{\text{LB}} := \min\{r_0, \dots, r_{n-1}\}$, $r_{\text{UB}} := \max\{r_0, \dots, r_{n-1}\}$ and $z_k \in \mathbb{Z}^n$. Moreover, Torczon showed that Δ_k can be written as follows:

$$\Delta_k := \tau^{r_k}\Delta_0, \tag{5}$$

in which $r_k \in \mathbb{Z}$. Both Equations (4) and (5) help us to prove Lemma 4.6 in Section 4.

3. The new non-monotone strategy

It is believed that some globalization techniques such as pattern search can generally guarantee the global convergence of the traditional direct search approaches. A monotonicity of the sequence of objective function values is generated by this globalization technique, which usually leads to produce short steps. Due to this fact, a slow numerical convergence is created for highly nonlinear problems, see [1,5,13,14,27,28,38]. As an example, the generalized pattern search framework exploits the quantity ρ_k which guarantees

$$f_k - f_{k+1} > 0,$$

this means that the sequence $\{f_k\}_{k\geq 0}$ is monotone. In order to avoid this drawback of globalization techniques, Gasparo *et al.* [13] based on the definition introduced by Grippo *et al.* [14], proposed a non-monotone strategy in pattern search algorithms with the quantity $\hat{\rho}_k$ satisfying

$$\hat{\rho}_k := f_{l(k)} - f_{k+1},$$

for which

$$f_{l(k)} := \max_{0\leq j\leq m(k)} \{f_{k-j}\}, \quad k \in \mathbb{N}_0 \tag{6}$$

in which $m(0) := 0$ and $0 \leq m(k) \leq \min\{m(k-1)+1, N\}$ with $N \geq 0$. This strategy has excellent results having caused many researchers to investigate the effects of these strategies in a wide variety of optimization procedures and to propose some other non-monotone techniques, see [1,13,14,27,28,38]. Although the non-monotone technique (6) has many advantages, this rule contributes to some drawbacks as well, see [1,38]. Recently, Ahookhosh and Amini [1] have presented a weaker non-monotone strategy of Grippo *et al.* [14] which overcomes some of its disadvantages [14] with the quantity $\bar{\rho}_k$ satisfying

$$\bar{\rho}_k := R_k - f_{k+1},$$

where

$$R_k := \eta_k f_{l(k)} + (1-\eta_k) f_k, \tag{7}$$

in which $\eta_k \in [\eta_{\min}, \eta_{\max}]$, $\eta_{\min} \in [0, 1)$ and $\eta_{\max} \in [\eta_{\min}, 1]$. Although, this proposal generates the more efficient algorithm, it depends on choosing η_k. An unsuitable choice of η_k can cause some shortcomings. According to the characteristics and expectations of our algorithm, we further propose an appropriate η_k. In this regard, let us first define the following ratio

$$\Theta_k := \frac{f_{l(k)}}{f_k},$$

which can help us to compare the distance between the members of $\{f_k\}_{k\geq 0}$ and $\{f_{l(k)}\}_{k\geq 0}$. It is clear that $\Theta_k \geq 1$ because $f_{l(k)} \geq f_k > 0$ and Lemma 4.5 show that $\lim_{k\to\infty} \Theta_k = 1$. Also, it can be seen that if $\Theta_k \geq \beta$ ($\beta > 1$), then $\{f_k\}_{k\geq 0}$ and $\{f_{l(k)}\}_{k\geq 0}$ are far away from each other and otherwise they are close. Now, after representing of

$$\hat{\eta}_k := \begin{cases} \dfrac{\eta_k}{\Theta_k} & \text{if } \Theta_k \geq \beta, \\ \eta_k \Theta_k & \text{else,} \end{cases} \tag{8}$$

a new non-monotone pattern search formula is defined by

$$\Lambda_k := \hat{\eta}_k f_{l(k)} + (1-\hat{\eta}_k) f_k, \tag{9}$$

for which the new quantity is considered as

$$\tilde{\rho}_k := \Lambda_k - f_{k+1}. \tag{10}$$

The theoretical and numerical results show that the new choice of $\tilde{\rho}_k$ has remarkable positive effects on pattern search to get faster convergence, especially for highly nonlinear problems. Let us now use the following procedure to compute the non-monotone strategy (9)

Procedure NM(Non-monotone)

Input: $\tilde{\rho}_k, f_{k+1}, f_{l(k)}, \eta_{\min}, \eta_{\max}, \Lambda_k, m(k), N$

begin
  if $\tilde{\rho}_k > 0$ **then**
    choose $m(k+1) \in [0, \min\{m(k)+1, N\}]$;
    calculate $f_{l(k+1)}$ by Equation (3) and choose $\eta_{k+1} \in [\eta_{\min}, \eta_{\max}]$;
    compute $\hat{\eta}_{k+1}$ by Equation (5) and Λ_{k+1} by Equation (6);
  else
    $m(k+1) := m(k); \eta_{k+1} := \eta_k; f_{l(k+1)} := f_{l(k)}; \Lambda_{k+1} := \Lambda_k$;
  end
end

Output: $f_{l(k+1)}, \eta_{k+1}, m(k+1), \eta_{k+1}$ and Λ_{k+1}

Remark 3.1: The sequence $\{\Lambda_k\}_{k\geq 0}$ generates the convergence results obtained by stronger non-monotone strategy whenever iterates are far away from the optimizer and the members of $\{f_k\}_{k\geq 0}$ and $\{f_{l(k)}\}_{k\geq 0}$ are close to each other while this sequence generates the convergence results obtained by weaker non-monotone strategy whenever iterates are close to the optimizer and the members of $\{f_k\}_{k\geq 0}$ and $\{f_{l(k)}\}_{k\geq 0}$ are far away from each other.

Before presenting our algorithm, we describe how to determine the step d_k by the following two procedures:

Procedure 3(Weak non-monotone hypotheses on exploratory moves)

(S.1) Compute P_k by Equation (3) and then choose $d_k \in \Delta_k P_k$.
(S.2) If $\min\{f(x_k + y),\ y \in \Delta_k B \Gamma_k\} < \Lambda_k$, then $f(x_k + d_k) < \Lambda_k$; **stop**. Otherwise, go to (S.1).

Procedure 3 tries to reach the condition $\Lambda_k \geq f_k$ at each iterate among any of the $2n$ steps presented by $\Delta_k B \Gamma_k$ without satisfying $f(x_k + d_k) \leq \min\{f(x_k + y),\ y \in \Delta_k B \Gamma_k\}$.

Procedure 4(Strong non-monotone hypotheses on exploratory moves)

(S.1) Compute P_k by Equation (3) and then choose $d_k \in \Delta_k P_k$.
(S.2) If $\min\{f(x_k + y),\ y \in \Delta_k B \Gamma_k\} < \Lambda_k$, then $f(x_k + d_k) \leq \min\{f(x_k + y),\ y \in \Delta_k B \Gamma_k\}$; **stop**. Otherwise, go to (S.1).

Now, to investigate the effectiveness of the new pattern search, we add the new non-monotone strategy to the framework of pattern search method.

Note that in Algorithm 2, if d_k is obtained by Procedure 3, then it is called ***non-monotone weak pattern search*** (***NMWPS-N***) while if d_k is obtained by Procedure 4, then it is considered as ***non-monotone strong pattern search*** (***NMSPS-N***). To guarantee the global convergence of NMWPS-N using Procedure 3 to determine d_k, we need to update Δ_k by

$$\Delta_{k+1} := \begin{cases} \lambda_k \Delta_k & \text{if } \tilde{\rho}_k > 0, \\ \theta \Delta_k & \text{else}, \end{cases} \tag{11}$$

Algorithm 2: NMPS-N (Non-monotone Pattern Search)

Input: An initial point $x_0 \in \mathbb{R}^n$, an initial integer matrix $C_0 \in \mathbb{Z}^{n\times p}$, $\epsilon > 0, \theta \in (0,1)$, $\eta_0 \in [\eta_{\min}, \eta_{\max}]$, $\lambda_k \geq 1$ and $N > 0$;

Output: x_b, f_b;

begin

  $k := 0$; $F_k := F(x_k)$; $f_k := \frac{1}{2}\|F_k\|^2$; $\Lambda_k := f_k$;

  while $\Delta_k \geq \epsilon$ **do**

    determine d_k by either Procedure 3 or Procedure 4; compute $F(x_k + d_k)$; $f(x_k + d_k) := \frac{1}{2}\|F(x_k + d_k)\|^2$;

    compute $\tilde{\rho}_k$ by (7);

    if $\tilde{\rho}_k > 0$ **then**

      $x_{k+1} := x_k + d_k$;

    else

      $x_{k+1} := x_k$;

    end

    set $F_{k+1} := F(x_{k+1})$; $f_{k+1} := \frac{1}{2}\|F_{k+1}\|^2$;

    update η_{k+1}, $m(k+1)$, $f_{l(k+1)}$ and Λ_{k+1} by NM procedure;

    update C_{k+1} and Δ_{k+1};

    $k \leftarrow k+1$;

  end

  $x_b := x_k$; $f_b := f_k$;

end

and NMSPS-N, d_k obtained by Procedure 4, updates Δ_k by

$$\Delta_{k+1} := \begin{cases} \Delta_k & \text{if } \tilde{\rho}_k > 0, \\ \theta\Delta_k & \text{else}, \end{cases} \tag{12}$$

where both θ and λ_k are updated similar to Algorithm 1. We determine how to update C_k in Section 5.

The global convergence results of both NMWPS-N and NMSPS-N require the following assumptions necessary:

(H1) The level set $L(x_0) := \{x \in \mathbb{R}^n \mid f(x) \leq f(x_0)\}$ is bounded.
(H2) $F(x)$ is continuously differentiable on a compact convex set Ω containing $L(x_0)$.

It can be easily seen that in Algorithm 2 for any index k, one of the following cases can occur

$$\mathcal{I}_1 := \{k \mid \Theta_k \geq \beta\}, \quad \mathcal{I}_2 := \{k \mid \Theta_k < \beta, \hat{\eta}_k \in (0,1]\} \quad \text{and} \quad \mathcal{I}_3 := \{k \mid \Theta_k < \beta, \hat{\eta}_k > 1\}.$$

Lemma 3.1: *Suppose that the sequence $\{x_k\}_{k\geq 0}$ is generated by Algorithm 2. Then, we have the following properties:*

(P1) *If $k \in \mathcal{I}_1$, then $f_k \leq \Lambda_k \leq R_k$.*
(P2) *If $k \in \mathcal{I}_2$, then $R_k < \Lambda_k \leq f_{l(k)}$.*
(P3) *If $k \in \mathcal{I}_3$, then $\Lambda_k > f_{l(k)}$.*

Proof: (1) This fact that $\Theta_k \geq \beta > 1$ implies $\hat{\eta}_k \leq \eta_k$ and consequently

$$\Lambda_k = \hat{\eta}_k f_{l(k)} + (1 - \hat{\eta}_k) f_k = \hat{\eta}_k (f_{l(k)} - f_k) + f_k \leq \eta_k (f_{l(k)} - f_k) + f_k = R_k.$$

On the other hand, because $f_{l(k)} \geq f_k$, it is easily seen that

$$\Lambda_k = \hat{\eta}_k f_{l(k)} + (1 - \hat{\eta}_k) f_k \geq \hat{\eta}_k f_k + (1 - \hat{\eta}_k) f_k = f_k.$$

So, (P1) is correct.

(2) Using the definition $f_{l(k)}$ along with this fact that $1 \leq \Theta_k < \beta$ implies that $\hat{\eta}_k \geq \eta_k$ and so

$$R_k = \eta_k (f_{l(k)} - f_k) + f_k \leq \hat{\eta}_k (f_{l(k)} - f_k) + f_k = \Lambda_k = (1 - \hat{\eta}_k)(f_k - f_{l(k)}) + f_{l(k)} \leq f_{l(k)},$$

which gives (P2).

(3) The definition of $f_{l(k)}$ and $\hat{\eta}_k > 1$ results in

$$\Lambda_k = \hat{\eta}_k f_{l(k)} + (1 - \hat{\eta}_k) f_k = (\hat{\eta}_k - 1)(f_{l(k)} - f_k) + f_{l(k)} > f_{l(k)},$$

so, (P3) is correct. ■

Based on Lemma 3.1, using the new sequence $\{\hat{\eta}_k\}_{k\geq 0}$ causes some appropriate properties. If $k \in \mathcal{I}_1$, (P1) concludes $\Lambda_k \leq R_k$, so in this case where iterates are close to the optimizer, the definition (9) proposes a weaker non-monotone strategy in relation to the non-monotone strategy (7). Otherwise, if $k \in \mathcal{I}_2$, then (P2) concludes that $R_k < \Lambda_k \leq f_{l(k)}$ and so it leads us to produce a medium non-monotone strategy whenever iterates are not far away from the optimizer. Finally, if $k \in \mathcal{I}_3$, far away from the optimizer, (P3) results in $\Lambda_k > f_{l(k)}$ and so algorithm uses a strong non-monotone strategy with respect to the non-monotone strategy (6).

4. Convergence analysis

In this section, we investigate the global convergence results of the new proposed algorithm.

Lemma 4.1: *Suppose that Assumption* (H1) *holds and the sequence* $\{x_k\}_{k\geq 0}$ *is generated by Algorithm* 2. *Then, for all* $k \in \mathbb{N}_0$, *we have* $x_k \in L(x_0)$ *and the sequence* $\{f_{l(k)}\}$ *for all* $k \in \mathcal{I}_1 \cup \mathcal{I}_2$ *is a convergent decreasing sequences and also for all* $k \in \mathcal{I}_3$ *provided that* $f_{k+1} \leq f_{l(k)}$.

Proof: If x_{k+1} is not accepted by Algorithm 2, then $f_{k+1} = f_k$ and $f_{l(k+1)} = f_{l(k)}$. Otherwise, we have

$$f_{k+1} = f(x_k + d_k) \leq \Lambda_k \quad \forall\, k \in \mathbb{N}_0. \tag{13}$$

In the sequel, we divide the proof into two parts.

(a) $k \in \mathcal{I}_1 \cup \mathcal{I}_2$. (P1), (P2) of Lemma 3.1 along with (13) imply that $f_{k+1} \leq f_{l(k)}$. In order to prove that the sequence $\{f_{l(k)}\}_{k\in\mathcal{I}_1\cup\mathcal{I}_2}$ is decreasing, we consider the following two cases:

(i) $k < N$. In this case $m(k+1) = k+1$. It is easily seen that

$$f_{l(k+1)} = \max_{0\leq j\leq k+1} \{f_{k+1-j}\} = \max\{f_{l(k)}, f_{k+1}\} = f_{l(k)}.$$

(ii) $k \geq N$. In this case, we have $m(k+1) = N$, for all k. Therefore, inequality $f_{k+1} \leq f_{l(k)}$ results in

$$f_{l(k+1)} = \max_{0\leq j\leq N} \{f_{k+1-j}\} \leq \max\left\{\max_{0\leq j\leq N} \{f_{k-j}\}, f_{k+1}\right\} = \max\{f_{l(k)}, f_{k+1}\} = f_{l(k)},$$

while the last inequality along with $k \in \mathcal{I}_1 \cup \mathcal{I}_2$ is conclusion of (13).

(b) $k \in \mathcal{I}_3$ and $f_{k+1} \leq f_{l(k)}$. The proof is similar to the cases (i) and (ii) of the part (a).

Now, by a strong induction, assuming $x_i \in L(x_0)$, for all $i = 1, \ldots, k$, it is sufficient to show $x_{k+1} \in L(x_0)$. Now, we can obtain

$$f_{k+1} \leq f_{l(k)} \leq f_0.$$

Thus, the sequence $\{x_k\}_{k\geq 0}$ is contained in $L(x_0)$. Finally, Assumption (H1) along with $x_k \in L(x_0)$ for all $k \in \mathbb{N}_0$ implies that the sequence $\{f_{l(k)}\}_{k\geq 0}$ is bounded. Thus, the sequence $\{f_{l(k)}\}_{k\geq 0}$ is convergent. ■

Lemma 4.2: *Suppose that Assumption* (H1) *holds and the sequence* $\{x_k\}_{k\geq 0}$ *is generated by Algorithm* 2*. Then, for all* $k \in \mathbb{N}_0$*, we have* $x_k \in L(x_0)$ *and whenever* $f_{k+1} > f_{l(k)}$*, the sequence* $\{\Lambda_k\}_{k\geq 0}$ *for all* $k \in \mathcal{I}_3$ *is a convergent decreasing sequences.*

Proof: If x_{k+1} is not accepted by Algorithm 2, then $f_{k+1} = f_k$ and $f_{l(k+1)} = f_{l(k)}$. Otherwise, we have

$$f_{k+1} = f(x_k + d_k) \leq \Lambda_k \quad \forall\, k \in \mathbb{N}_0.$$

This fact along with $f_{k+1} > f_{l(k)}$ and the definition $f_{l(k+1)}$ results in $f_{l(k+1)} = f_{k+1}$ and also

$$\Lambda_{k+1} = \hat{\eta}_{k+1} f_{l(k+1)} + (1 - \hat{\eta}_{k+1}) f_{k+1} = \hat{\eta}_{k+1} f_{k+1} + (1 - \hat{\eta}_{k+1}) f_{k+1} = f_{k+1} \leq \Lambda_k.$$

Now, by a strong induction, assuming $x_i \in L(x_0)$, for all $i = 1, \ldots, k$, it is sufficient to show $x_{k+1} \in L(x_0)$. Now, we can obtain

$$f_{k+1} = \Lambda_{k+1} \leq \Lambda_k \leq f_0.$$

Thus, the sequence $\{x_k\}_{k\geq 0}$ is contained in $L(x_0)$. Finally, Assumption (H1) along with $x_k \in L(x_0)$ for all $k \in \mathbb{N}_0$ implies that the sequence $\{\Lambda_k\}_{k\geq 0}$ is bounded. Thus, the sequence $\{\Lambda_k\}_{k\geq 0}$ is convergent. ■

Lemma 4.3: *Let* $\{x_k\}_{k\geq 0}$ *be a bounded sequence of vectors in* $\mathbb{R}^n$ *by the NMSPS-N algorithm and* $\eta \in \mathbb{R}$ *such that* $\|\nabla f_k\| \geq \eta > 0$*. Then, under Assumptions* (H1) *and* (H2)*, there exist* $\delta > 0$*, such that for all* $\Delta_k > 0$*, if* $\Delta_k \leq \delta$*, then the kth iteration of NMSPS-N will be successful* ($\tilde{\rho}_k > 0$) *and* $\Delta_{k+1} \geq \Delta_k$*.*

Proof: Similar to Proposition 6.4 in [31], for $i = 1, \ldots, p$, If $\Delta_k < \delta$, then we can get

$$f(x_k + d_k^i) - f(x_k) \leq -\tfrac{1}{2}\xi \|\nabla f_k\| \|d_k^i\| < 0,$$

in which $\xi > 0$ is a constant. Hence, there exists at least one $i \in \{1, \ldots, p\}$ such that $d_k^i \in \Delta_k B C_k$. Whenever $\Delta_k < \delta$, $f(x_k + d_k^i) < f(x_k) \leq \Lambda_k$. If $\min\{f(x_k + y),\ y \in \Delta_k B \Gamma_k\} < \Lambda_k$, then Procedures 3 guarantees $f(x_k + d_k) < \Lambda_k$ and consequently $\tilde{\rho}_k > 0$. According NMWSP-N, we have $\Delta_{k+1} \geq \Delta_k$. ■

Lemma 4.3 gives the following corollary, see Corollary 6.5 in [31].

Corollary 4.4: *Let* $\{x_k\}_{k\geq 0}$ *be a bounded sequence of vectors in* $\mathbb{R}^n$ *by NMWPS-N and* $\eta \in \mathbb{R}$ *such that* $\|\nabla f_k\| \geq \eta > 0$*. Then, under Assumptions* (H1) *and* (H2)*, there exist* $\zeta, \delta > 0$*, such that for all* $\Delta_k > 0$*, if* $\Delta_k \leq \delta$*, then*

$$f_{k+1} \leq f_k - \zeta \|\nabla f_k\| \|d_k\|.$$

The above corollary helps us to establish the following lemma.

Lemma 4.5: *Suppose that Assumptions* (H1) *and* (H2) *hold and the sequence* $\{x_k\}_{k\geq 0}$ *is generated by the NMWPS-N algorithm. Then, we have*

$$\lim_{k\to\infty} f_{l(k)} = \lim_{k\to\infty} f_k.$$

Proof: Using the fact that x_k is not the optimum of (2), we can conclude that there exists a constant $\epsilon > 0$ such that $\|\nabla f_k\| \geq \epsilon$. This fact along with Lemma 4.4 and $f_k \leq f_{l(k)}$, for some $\zeta > 0$, implies that

$$\begin{aligned} f_{k+1} &= f(x_k + d_k) \\ &\leq f_k - \zeta \|\nabla f_k\| \|d_k\| \\ &\leq f_k - \zeta\epsilon \|d_k\| \\ &\leq f_{l(k)} - \omega \|d_k\|, \end{aligned} \tag{14}$$

where $\omega = \zeta\epsilon$. By replacing k with $l(k) - 1$ in Equation (14), we have

$$f_{l(l(k)-1)} - f_{l(k)} \geq \omega \|d_{l(k)-1}\|. \tag{15}$$

This fact along with Lemma 4.2 results in

$$\lim_{k\to\infty} \|d_{l(k)-1}\| = 0. \tag{16}$$

Assumption (H2) and (16) give

$$\lim_{k\to\infty} f(x_{l(k)}) = \lim_{k\to\infty} f(x_{l(k)-1}). \tag{17}$$

By letting $\hat{l}(k) = l(k + N + 2)$ and using the induction, for all $j \geq 1$, we can prove

$$\lim_{k\to\infty} \|d_{\hat{l}(k)-j-1}\| = 0. \tag{18}$$

This fact that $\{\hat{l}(k)\} \subset \{l(k)\}$ and according to Equation (16), for $j = 1$, Equation (18) is satisfied. Assume that Equation (18) holds for a given j and take k large enough so that $\hat{l}(k) - (j + 1) > 0$. Using Equation (14) and substituting k with $\hat{l}(k) - j - 1$, we have

$$f(x_{\hat{l}(k)-j-1}) - f(x_{\hat{l}(k)-j}) \leq \omega \|d_{\hat{l}(k)-j-1}\|.$$

Following the same argument to derive (17), we deduce that

$$\lim_{k\to\infty} \|d_{\hat{l}(k)-j-1}\| = 0.$$

and also

$$\lim_{k\to\infty} f(x_{\hat{l}(k)-j-1}) = \lim_{k\to\infty} f(x_{l(k)}).$$

Similar with Equation (17), for any given $j \geq 1$, we have

$$\lim_{k\to\infty} f(x_{\hat{l}(k)-j}) = \lim_{k\to\infty} f(x_{l(k)}).$$

On the other hand, we can generate

$$x_{k+1} = x_{\hat{l}(k)} - \sum_{j=1}^{\hat{l}(k)-k-1} d_{\hat{l}(k)-j}, \quad \forall k.$$

This fact along with Equation (18) and $\hat{l}(k) - j - 1 \leq N + 1$ implies that

$$\lim_{k\to\infty} \|x_{k+1} - x_{\hat{l}(k)}\| = 0.$$

Hence, Assumption (H2) leads to

$$\lim_{k\to\infty} f(x_{l(k)}) = \lim_{k\to\infty} f(x_{\hat{l}(k)}) = \lim_{k\to\infty} f(x_k). \qquad \blacksquare$$

Using Lemma 4.5, we can obtain the following corollary.

Corollary 4.6: *Suppose that Assumptions* (H1) *and* (H2) *hold and the sequence* $\{x_k\}_{k\geq 0}$ *is generated by the NMWPS-N algorithm. Then, we have*

$$\lim_{k\to\infty} \Lambda_k = \lim_{k\to\infty} f_k.$$

Proof: (1) If $k \in \mathcal{I}_1 \cup \mathcal{I}_2$, then the inequality $f_k \leq \Lambda_k \leq f_{l(k)}$ along with Lemma 4.5 implies that

$$\lim_{\substack{k\to\infty\\ k\in\mathcal{I}_1\cup\mathcal{I}_2}} \Lambda_k = \lim_{\substack{k\to\infty\\ k\in\mathcal{I}_1\cup\mathcal{I}_2}} f_k.$$

(2) For $k \in \mathcal{I}_3$, recalling Lemma 4.5 along with the definition of $\hat{\eta}_k$ results in

$$\lim_{\substack{k\to\infty\\ k\in\mathcal{I}_3}} \Lambda_k = \lim_{\substack{k\to\infty\\ k\in\mathcal{I}_3}} f_k. \qquad \blacksquare$$

The following lemmas show that NMWPS-N and NMSPS-N algorithms are well-defined.

Lemma 4.7: *Suppose that Assumption* (H1) *holds and the NMWPS-N algorithm has constructed an infinite sequence* $\{x_k\}_{k\geq 0}$, *then* $\lim_{k\to\infty} \inf \Delta_k = 0$.

Proof: By contradiction, suppose that $\lim_{k\to\infty} \inf \Delta_k = 0$ is not satisfied; hence, we can assume that there exists a constant $\Delta_{\rm LB} > 0$ and a set index $K \subset \mathbb{N}_0$ such that

$$\Delta_k \geq \Delta_{\rm LB}, \quad \forall k \in K.$$

This fact along with Equation (5) results in

$$\tau^{r_k} \geq \frac{\Delta_{\rm LB}}{\Delta_0} > 0, \quad \forall k \in K,$$

which means that the sequence $\{\tau^{r_k}\}_{k\in K}$ is bounded away from zero. Since $\{x_k\}_{k\geq 0} \in L(x_0)$ and $L(x_0)$ is compact, Lemma 3.1 in [31] implies that the sequence $\{\Delta_k\}_{k\geq 0}$ has an upper bounded denoted by $\Delta_{\rm UB}$ and hence the sequence $\{\tau^{r_k}\}_{k\in K}$ is bounded above. In other words, the sequence $\{\tau^{r_k}\}_{k\in K}$ is finite and consequently $\{r_k\}_{k\geq 0}$ has, respectively, a lower and upper bounded, defined by

$$r_{\rm LB} := \min\{r_k \mid 0 \leq k < +\infty\} \quad \text{and} \quad r_{\rm UB} := \max\{r_k \mid 0 \leq k < +\infty\},$$

hence, for any $k \in K$, it can be concluded

$$x_k := x_0 + (\beta^{r_{\rm LB}}\alpha^{-r_{\rm UB}})\Delta_0 B \sum_{k=0}^{k-1} z_k,$$

i.e. it lies on a translated integer lattice generated by x_0 and the columns of $(\beta^{r_{\rm LB}}\alpha^{-r_{\rm UB}})\Delta_0 B$, denoted by K_1. Therefore $x_k \in L(x_0) \cap K_1$ for which $L(x_0) \cap K_1$ is finite and it must has at least a point x_* in $L(x_0) \cap K_1$ such that $x_k := x_*$ for infinitely many k. By steps of NMWPS-N, a lattice point can be revisited finitely many times; hence, the new step d_k is accepted if only if $\Lambda_k > f(x_k + d_k)$. This fact implies that there exists an positive index m such that $x_k := x_*$, for $k \geq m$. This fact together with Corollary 4.6 yields to $\widetilde{\rho}_k \to 0$ and consequently $\Delta_k \to 0$, which is a contradiction since $0 < \Delta_{\rm LB} \leq 0$. $\blacksquare$

Since NMSPS-N uses the relationship (12) to update Δ_k, it ensures that $\lim_{k\to\infty} \Delta_k = 0$.

Corollary 4.8: *Suppose that Assumption* (H1) *holds and the NMSPS-N algorithm has constructed an infinite sequence* $\{x_k\}_{k\geq 0}$. *Then,* $\lim_{k\to\infty} \Delta_k = 0$.

Remark 4.1: Grippo and Sciandrone, in Proposition 2 in [23], showed that if there exist the sequences $\{c_k^i\}_{k\geq 0}$, $i = 1, \ldots, p$, which are bounded and each limit point of the sequence $\{c_k^1, \ldots, c_k^p\}_{k\geq 0}$ is denoted by $(c_*^1, \ldots, c_*^p)$ for which c_*^i, $i = 1, \ldots, p$, is positively span $\mathbb{R}^n$, then

$$\lim_{k\to\infty} \|\nabla f_k\| = 0 \quad \Longleftrightarrow \quad \lim_{k\to\infty} \sum_{i=1}^{p} \min\left\{0, \frac{\nabla f(x_k)^{\mathrm{T}} c_k^i}{\|c_k^i\|}\right\} = 0. \tag{19}$$

Theorem 4.9: *Suppose that Assumptions* (H1) *and* (H2) *hold. Let* $\{x_k\}_{k\geq 0}$ *be the infinite sequence generated by the NMWPS-N. Then,*

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0. \tag{20}$$

Proof: By contradiction, we assume that Equation (20) does not hold. Then, there exists a constant $\delta > 0$ such that $\|\nabla f_k\| \geq \delta$ for all $k \in \mathbb{N}_0$. From Lemma 4.7, there exists an infinite sequence $\mathcal{K}$ such that

$$\liminf_{k\to\infty,\ k\in\mathcal{K}} \Delta_k = 0. \tag{21}$$

By recalling the continuous differentiability of f, it can find, for each $k \in \mathbb{N}_0$ and for $i = 1, \ldots, p$, $\xi_k^i := x_k + \omega_k d_k^i = x_k + \omega_k \Delta_k c_k^i$, in which $\omega_k \in (0, 1)$, such that

$$f(x_k + d_k^i) = f(x_k) + \nabla f(\xi_k^i)^{\mathrm{T}} d_k^i \leq \Lambda_k + \nabla f(\xi_k^i)^{\mathrm{T}} d_k^i. \tag{22}$$

We can get $\{\xi_k\}_{k\in\mathcal{K}} \to x_*$ because Equation (21) gives $\{d_k\}_{k\in\mathcal{K}} \to 0$ $(\lim_{k\to\infty,\ k\in\mathcal{K}}(c_k^i/\|c_k^i\|) = c_*^i)$ and $\{x_k\}_{k\in\mathcal{K}} \to x_*$. Now, these facts along with taking a limit from both sides (22), for $i = 1, \ldots, p$, gives

$$\begin{aligned}
\nabla f(x_*)^{\mathrm{T}} c_*^i &= \lim_{k\to\infty} \frac{\nabla f(x_k)^{\mathrm{T}} c_k^i}{\|c_k^i\|} \\
&= \lim_{k\to\infty} \frac{\nabla f(\xi_k)^{\mathrm{T}} c_k^i}{\|c_k^i\|} \geq 0,
\end{aligned}$$

yielding to

$$\lim_{k\to\infty,\ k\in\mathcal{K}} \sum_{i=1}^{p} \min\left\{0, \frac{\nabla f(x_k)^{\mathrm{T}} c_k^i}{\|c_k^i\|}\right\} = 0.$$

Then, by Equation (19), we get

$$\lim_{k\to\infty,\ k\in\mathcal{K}} \|\nabla f_k\| = 0,$$

leading

$$\liminf_{k\to\infty} \|\nabla f_k\| = 0. \qquad \blacksquare$$

The following lemma helps us to establish the main global theorem.

Lemma 4.10: *Suppose that Assumptions* (H1) *and* (H2) *hold and the columns of the* C_k *are bounded in norm, i.e. there exist two positive constant* γ_1 *and* γ_2 *such that* $\gamma_1 \leq \|c_k^i\| \leq \gamma_2$ *for* $i = 1, \ldots, p$. *Let*

$\{x_k\}_{k\geq 0}$ be the sequence generated by NMSPS-N. If there exists a positive constant δ and a subsequence $K \subseteq \mathbb{N}_0$ such that $\|\nabla f_k\| \geq \delta$ for $k \in K$, then

$$\sum_{k\in K} \Delta_k < \infty.$$

Proof: First, we show that

$$f_{k+1} \leq f_0 - \zeta\delta\gamma_1 \sum_{j=0,j\in K}^{k} \Delta_j.$$

By Lemma 4.4, we get

$$\begin{aligned} f_{k+1} &\leq f_k - \zeta\,\|\nabla f_k\|\|d_k\| \\ &\leq f_k - \zeta\delta\gamma_1\Delta_k \\ &\leq (f_{k-1} - \zeta\delta\gamma_1\Delta_{k-1}) - \zeta\delta\gamma_1\Delta_k \\ &\leq f_0 - \zeta\delta\gamma_1 \sum_{j=0,j\in K}^{k} \Delta_k. \end{aligned}$$

Suppose that there exists a subset index $K' \subset K$ such that $\sum_{k\in K'} \Delta_k = \infty$. Then, we get

$$f_0 \geq f_0 - f_k \geq \zeta\delta\gamma_1 \sum_{j=0,\ j\in K'}^{k-1} \Delta_j \to \infty, \quad \text{as } k \to \infty,$$

yielding to $f_0 \to \infty$, which is a contradiction. Hence, we conclude that

$$\sum_{j=0,\ j\in K}^{k-1} \Delta_j < \infty. \qquad \blacksquare$$

At this point, the global convergence of Algorithm 2 based on the mentioned assumptions of this section can be investigated.

Theorem 4.11: *Suppose that Assumptions* (H1) *and* (H2) *hold and the columns of the C_k are bounded in norm, i.e. there exist two positive constant γ_1 and γ_2 such that $\gamma_1 \leq \|c_k^i\| \leq \gamma_2$ for $i = 1, \ldots, p$. Then, for any $\{x_k\}_{k\geq 0}$ generated by the non-monotone pattern search method* (*NMSPS-N*),

$$\lim_{k\to\infty} \|\nabla f_k\| = 0. \tag{23}$$

Proof: By contradiction, let us assume that the conclusion does not hold. Then, there is a subsequence of successful iterations such that

$$\|\nabla f_k\| \geq \delta > 0, \quad \text{for some } \delta > 0.$$

Theorem 4.9 guarantees that, for each $i = 1, \ldots, p$, there exists a first successful iteration $l(t_i) > t_i$ such that $\|\nabla f_{t_i}\| < \delta$. Denote $l_i := l(t_i)$ and define the index set $\Xi^i := \{k \mid t_i \leq k < l_i\}$, hence there exists another subsequence l_i such that

$$\|\nabla f_k\| \geq \delta, \quad \forall k \in \Xi^i \quad \text{and} \quad \|\nabla f_{l_i}\| < \delta. \tag{24}$$

This fact along with taking $\Xi := \cup_{i=0}^{\infty} \Xi^i$ leads to

$$\liminf_{k\to\infty, k\in\Xi} \|\nabla f_k\| \geq \delta.$$

Then, Lemma 4.10 gives

$$\sum_{j\in\Xi} \Delta_j < \infty,$$

leading to

$$\lim_{i\to\infty} \sum_{j\in\Xi^i} \Delta_j = 0.$$

Hence

$$\|x_{t_i} - x_{l_i}\| \leq \sum_{j\in\Xi^i} \|x_j - x_{j+1}\| \leq \sum_{j\in\Xi^i} \Delta_j \to 0, \quad \text{as } i \to \infty,$$

which deduces from continuity of $\nabla f(x)$ on $L(x_0)$

$$\lim_{i\to\infty} \|\nabla f_{t_i} - \nabla f_{l_i}\| = 0.$$

This is a contradiction since Equation (24) implies $\|\nabla f_{t_i} - \nabla f_{l_i}\| \geq \delta$. ■

5. Numerical experiments

One of the well known pattern search methods is the generalized coordinate search method with fixed step lengths [31]. This section reports some numerical experiments. Our algorithm, NMCS-N, is compared with the following considered algorithms

- GSCS: The generalized strong coordinate search [31]
- NMSCS-G: Algorithm 2 with the non-monotone term of Grippo *et al.* [14]
- NMSCS-A: Algorithm 2 with the non-monotone term of Ahookhosh and Amini [1]
- NMSCS-Z: Algorithm 2 with the non-monotone term of Zhang and Hager [38]

Test problems were selected from a wide range of papers: Problems 1–23 from [25], problems 24–31 from [24] and the problems 32–52 from [18].

All codes are written in MATLAB 9 programming environment on a 2.7 GHz Pentium(R) Dual-core CPU Windows 7 PC with 2G RAM with double precision format in the same subroutine. In our numerical experiments, the algorithms are stopped

$$\Delta_k \leq 10^{-6},$$

or whenever the total number of function evaluations exceeds 100,000. For all algorithms, we take advantages of the parameters $\lambda := 1.5$, $\theta := 0.5$, $\Delta_0 := 1$ and $B := I$. To calculate the non-monotone term $f_{l(k)}$, NMSCS-G, NMSCS-A and NMSCS-N have selected $N := 5$. For NMSCS-A, NMSCS-Z and NMSCS-N, we use $\eta_0 := 0.001$ and for NMSCS-A and NMSCS-N, the parameter η_k is updated by

$$\eta_k := \begin{cases} \eta_0/2, & \text{if } k = 1, \\ (\eta_{k-1} + \eta_{k-2})/2, & \text{if } k \geq 2. \end{cases}$$

For NMSCS-N, we have taken $\beta := 1 + \epsilon_m$, in which ϵ_m is machine ϵ. For all iterations of the coordinate search method, the generating matrix is fixed, i.e. $C_k := C$. Hence, this matrix contains in its columns all possible combinations of $\{-1, 0, 1\}$ and consequently it has $p = 3^n$ columns. In particular, the columns of C contain both I and $-I$, as well as a column of zeros.

The following algorithm briefly summarizes how the exploratory move directions for non-monotone coordinate search are generated, see [31]:

Algorithm 3: NSEMN-N (Non-monotone Strong Exploratory Moves)

Input: An initial point x_k, Δ_k, Λ_k, $f_{\min} := f(x_k)$ and e_i is the unit coordinate vector;
Output: $d_b := d_k$;
begin
    $d_k := 0$; $\tilde{\rho}_k := 0$;
    for $i = 1, 2, \ldots, n$ **do**
        $d_k^i := d_k + \Delta_k e_i$; $x_k^i := x_k + d_k^i$;
        compute $f(x_k^i)$;
        if $f(x_k^i) < f_{\min}$ **then**
            $\tilde{\rho}_k := \Lambda_k - f(x_k^i)$; $f_{\min} := f(x_k^i)$; $d_k := d_k^i$;
        else
            $d_k^i := d_k - \Delta_k e_i$; $x_k^i := x_k + d_k^i$;
            compute $f(x_k^i)$;
            if $f(x_k^i) < f_{\min}$ **then**
                $\tilde{\rho}_k := \Lambda_k - f(x_k^i)$;
                $f_{\min} := f(x_k^i)$;
            end
        end
    end
    $d_b := d_k$;
end

The exploratory moves are executed sequentially in the sense that the selection of the next trial step is based on the success or failure of the previous trial step. Thus, we may compute as few as n trial steps while there are 3^n possible trial steps, but we compute no more than $2n$ at any given iteration, see Figure 1 in [31]. However, in the worst case, the algorithm for coordinate search ensures that all $2n$ steps, defined by $\Delta_k B\Gamma = \Delta_k B[M - M] = \Delta_k[I - I]$, are tried before returning the step $d_k = 0$. In other words, the exploratory moves given in Algorithm 3 examine all $2n$ steps defined by $\Delta_k B\Gamma$ unless a step satisfying $f(x_k + d_k) < \Lambda_k$ is found.

At this point, to have a more reliable comparison and demonstrate the overall behaviour of the present algorithms and get more insight about the performance of considered codes, the performance of all codes, based on both C_t and N_f which are shown in Table 1, have been, respectively, assessed in Figure 1 by applying the performance profile proposed from Dolan and Moré in [6]. Subfigure (a) and (b) of Figure 1 plot the function $P(\tau) : [0, r_{\max}] \to \mathbb{R}^+$, considered as

$$P(\tau) := \frac{\text{card}(p \in \mathcal{P} \mid r_{p,s} \le \tau)}{\text{card}(\mathcal{P})}, \quad \tau \ge 1,$$

where $\mathcal{P}$ denotes the set of test problems, $r_{p,s}$ denotes the ratio of number of function evaluations and CPU-times needed to solve problem p with method s with the least number of function evaluations and CPU-times needed to solve problem p, respectively, and $r_{\max}$ is the maximum value of $r_{p,s}$. Finally, the highest on the plot is describing the best solver.

On one hand, subfigure (a) of Figure 1 compares NMSCS-N in the sense of the total number of function evaluations. It can be easily seen that NMSCS-N is the best algorithm in the sense of the

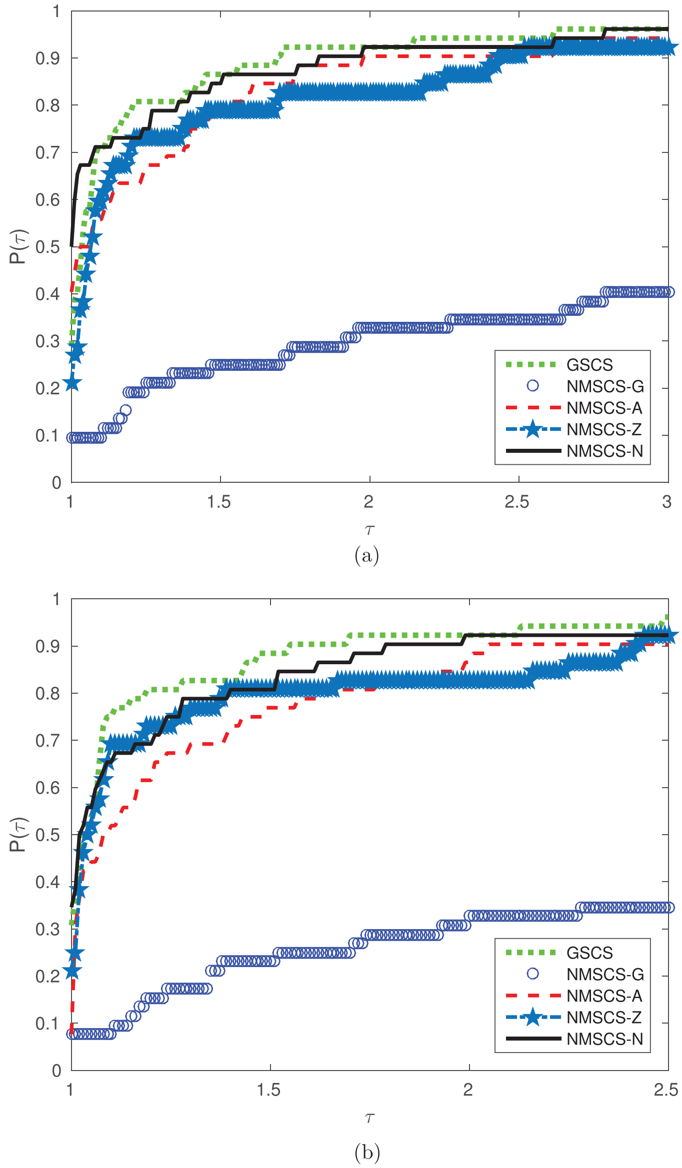

Figure 1. A comparison among proposed algorithms with the performance measures: (a) Number of function evaluations (top), (b) CPU-times (bottom).

most wins on more than 50% of the test functions. On the other hand, to compare the CPU times, because of variation of CPU time, each problem is solved five times and then the average of the CPU times is taken into account. Subfigure (b) of Figure 1 represents a comparison among the considered algorithms regarding CPU times. The results of this subfigure indicate that the performance of NMSCS-N is better than other present algorithms. In details, the new algorithm is the best algorithm on more than 35% of all cases.

6. Concluding remarks

This paper proposes a new non-monotone coordinate search algorithm to solve systems of nonlinear equations. Our method can overcome some disadvantages of the proposed method by Ahookhosh

Table 1. List of test functions.

Problem name	Dim	Problem name	Dim
Extended Powell badly scaled	2	Powell singular	4
Brent	3	Broyden banded	5
Seven-Diagonal System	7	Chebyquad	10
Extended Powell Singular	8	Brown almost linear	10
Triadiagnal exponential	10	Discrete integral equation	20
Generalized Broyden banded	10	Diag. func. premul. by ... matrix	3
Flow in a channel	10	Function 18	3
Swiriling flow	10	Strictly convex 2	5
Thorech	12	Strictly convex 1	5
Trig. exponential system 2	15	Zero Jacobian	5
Countercurrent reactors 1	16	Geometric	5
Countercurrent reactors 2	16	Extended Rosenbrock	6
Porous medium	16	Geometric programming	8
Trigonometric	20	Tridimensional valley	9
Singular Broyden	20	Chandrasekhar's H-equation	10
Broyden tridiagonal	20	Singular	10
Extended Wood	20	Logarithmic	10
Extended Cragg and Levy	24	Variable band 2	10
Trig. exponential system 1	25	Function 15	10
Structured Jacobian	25	Linear function-full rank 1	10
Discrete boundary value	25	Hanbook	10
Possion	25	Variable band 1	15
Possion 2	25	Linear function-full rank 2	20
Rosenbrock	2	Function 27	20
Powell badley scaled	2	Complementary	20
Helical valley	3	Function 21	21

and Amini [1] by presenting a new parameter, defined by using combination of the maximum function value of some preceding successful iterates and the current function value. This parameter can prevent the production of weaker non-monotone strategy whenever iterates are far away from the optimizer and stronger nonmonotone strategy whenever iterates are close to the optimizer. The global convergence properties of the proposed algorithms are established. Preliminary numerical results show the significant efficiency of the new algorithm.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The second author acknowledges the financial support of the Doctoral Program 'Vienna Graduate School on Computational Optimization' funded by Austrian Science Foundation under Project No W1260-N35.

References

[1] M. Ahookhosh and K. Amini, *An efficient nonmonotone trust-region method for unconstrained optimization*, Numer. Algorithms 59 (2012), pp. 523–540.
[2] M. Ahookhosh, K. Amini, and M. Kimiaei, *A globally convergent trust-region method for large-scale symmetric nonlinear systems*, Numer. Funct. Anal. Optim. 36 (2015), pp. 830–855.
[3] M. Ahookhosh, H. Esmaeili, and M. Kimiaei, *An effective trust-region-based approach for symmetric nonlinear systems*, Int. J. Comput. Math. 90(3) (2013), pp. 671–690.
[4] G.E.P. Box, *Evolutionary operation: A method for increasing industrial productivity*, Appl. Stat. 6 (1957), pp. 81–101.
[5] Y.H. Dai, *On the nonmonotone line search*, J. Optim. Theory Appl. 112(2) (2002), pp. 315–330.
[6] E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program. 91 (2002), pp. 201–213.
[7] H. Esmaeili and M. Kimiaei, *An improved adaptive trust-region method for unconstrained optimization*, Math. Model. Anal. 19 (2014), pp. 469–490.

[8] H. Esmaeili and M. Kimiaei, *An efficient adaptive trust-region method for systems of nonlinear equations*, Int. J. Comput. Math. 92 (2015), pp. 151–166.
[9] H. Esmaeili and M. Kimiaei, *A trust-region method with improved adaptive radius for systems of nonlinear equations*, Math. Methods Oper. Res. 83(1) (2016), pp. 109–125.
[10] J.Y. Fan, *Convergence rate of the trust region method for nonlinear equations under local error bound condition*, Comput. Optim. Appl. 34 (2005), pp. 215–227.
[11] J. Fan and J. Pan, *An improved trust region algorithm for nonlinear equations*, Comput. Optim. Appl. 48(1) (2011), pp. 59–70.
[12] M.G. Gasparo, *A nonmonotone hybrid method for nonlinear systems*, Optim. Methods Softw. 13 (2000), pp. 79–94.
[13] M.G. Gasparo, A. Papini, and A. Pasquali, *Nonmonotone algorithms for pattern search methods*, Numer. Algorithms 28 (2001), pp. 171–186.
[14] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal. 23 (1986), pp. 707–716.
[15] L. Grippo, F. Lampariello, and S. Lucidi, *A truncated Newton method with nonmonotone line search for unconstrained optimization*, J. Optim. Theory Appl. 60(3) (1989), pp. 401–419.
[16] L. Grippo, F. Lampariello, and S. Lucidi, *A class of nonmonotone stabilization methods in unconstrained optimization*, Numer. Math. 59 (1991), pp. 779–805.
[17] R. Hooke and T.A Jeeves, *Direct search solution of numerical and statistical problems*, J. ACM 8 (1961), pp. 212–229.
[18] W. LaCruz, C. Venezuela, J.M. Martínez, and M. Raydan, *Spectral residual method without gradient information for solving large-scale nonlinear systems of equations: Theory and experiments*, Technical Report RT–04–08, July 2004.
[19] R.M Lewis and V. Torczon, *Pattern search algorithms for bound constrained minimization*, SIAM. J. Optim. 9 (1999), pp. 1082–1099.
[20] R.M. Lewis and V. Torczon, *Pattern search methods for linearly constrained minimization*, SIAM. J. Optim. 10 (2000), pp. 917–941.
[21] R.M. Lewis, V. Torczon, and M.W. Trosset, *Why pattern search works*, Optima (1988), pp. 1–7.
[22] R.M. Lewis, V. Torczon, and M.W. Trosset, *Direct search methods: Then and now*, J. Comput. Appl. Math. 124 (2000), pp. 191–207.
[23] S. Lucidi and M. Sciandrone, *On the global convergence of derivative free methods for unconstrained optimization*, Technical Report 32–96, DIS, Universita' di Roma 'La Sapienza', 1996.
[24] L. Lukšan and J. Vlček, *Sparse and partially separable test problems for unconstrained and equality constrained optimization*, Techical Report, No. 767, January 1999.
[25] J.J. Moré, B.S. Garbow, and K.E. Hillström, *Testing Unconstrained Optimization Software*, ACM Trans. Math. Softw. 7 (1981), pp. 17–41.
[26] J. Nocedal and J.S. Wright, *Numerical Optimization*, Springer, NewYork, 2006.
[27] Z.J. Shi and S. Wang, *Modified nonmonotone Armijo line search for descent method*, Numer. Algorithms 57(1) (2011), pp. 1–25.
[28] P.L. Toint, *An assessment of nonmonotone linesearch techniques for unconstrained optimization*, SIAM J. Sci. Comput. 17 (1996), pp. 725–739.
[29] V. Torczon, *Multidirectional search: A direct search algorithm for parallel machines*, Ph.D. thesis, Rice University, Houston, TX, 1989.
[30] V. Torczon, *On the convergence of the multidirectional search algorithm*, SIAM J. Optim. 1 (1991), pp. 123–145.
[31] V. Torczon, *On the convergence of pattern search algorithms*, SIAM J. Optim. 7 (1997), pp. 1–25.
[32] G.L. Yuan and X.W. Lu, *A new backtracking inexact BFGS method for symmetric nonlinear equations*, Comput. Math. Appl. 55 (2008), pp. 116–129.
[33] G.L. Yuan and M.J. Zhang, *A three-terms Polak-R*ibière-Polyak conjugate gradient algorithm for large-scale nonlinear equations*, J. Comput. Appl. Math. 286 (2015), pp. 186–195.
[34] G.L. Yuan, S. Lu, and Z. Wei, *A new trust-region method with line search for solving symmetric nonlinear equations*, Int. J. Comput. Math. 88(10) (2011), pp. 2109–2123.
[35] G.L. Yuan, Z.H. Meng, and Y. Li, *A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations*, J. Optim. Theory Appl. 168 (2016), pp. 129–152.
[36] G.L. Yuan, X.W. Lu, and Z.X. Wei, *BFGS trust-region method for symmetric nonlinear equations*, J. Comput. Appl. Math. 230 (2009), pp. 44–58.
[37] G.L. Yuan, Z.X. Wei, and X.W. Lu, *A BFGS trust-region method for nonlinear equations*, Computing 92(4) (2011), pp. 317–333.
[38] H.C Zhang and W.W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM J. Optim. 14(4) (2004), pp. 1043–1056.
[39] J. Zhang and Y. Wang, *A new trust region method for nonlinear equations*, Math. Methods Oper. Res. 58 (2003), pp. 283–298.