Numer Algor (2014) 66:49–78
DOI 10.1007/s11075-013-9723-x

ORIGINAL PAPER

An inexact line search approach using modified nonmonotone strategy for unconstrained optimization

Keyvan Amini · Masoud Ahookhosh · Hadi Nosratipour

Received: 13 May 2012 / Accepted: 14 May 2013 / Published online: 10 June 2013

Abstract This paper concerns with a new nonmonotone strategy and its application to the line search approach for unconstrained optimization. It has been believed that nonmonotone techniques can improve the possibility of finding the global optimum and increase the convergence rate of the algorithms. We first introduce a new nonmonotone strategy which includes a convex combination of the maximum function value of some preceding successful iterates and the current function value. We then incorporate the proposed nonmonotone strategy into an inexact Armijo-type line search approach to construct a more relaxed line search procedure. The global convergence to first-order stationary points is subsequently proved and the *R*-linear convergence rate are established under suitable assumptions. Preliminary numerical results finally show the efficiency and the robustness of the proposed approach for solving unconstrained nonlinear optimization problems.

Keywords Unconstrained optimization · Armijo-type line search · Nonmonotone technique · Theoretical convergence

K. Amini (✉) · H. Nosratipour
Department of Mathematics, Faculty of Science, Razi University, Kermanshah, Iran
e-mail: kamini@razi.ac.ir

H. Nosratipour
e-mail: hadi.nosratipour@gmail.com

M. Ahookhosh
Faculty of Mathematics, University of Vienna, Nordbergstrasse 15, 1090 Vienna, Austria
e-mail: masoud.ahookhosh@univie.ac.at

1 Introduction

Consider the following unconstrained nonlinear optimization problem

$$\text{Minimize } f(x), \quad \text{subject to } x \in \mathbf{R}^n, \tag{1}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ is a twice continuously differentiable function. There are many iterative methods to solve the problem (1). Most of these methods could be divided into two general classes, namely line search and trust-region frameworks (see [18]). Trust-region methods try to find a neighborhood around the current step x_k in which a quadratic model should be agreed with the objective function. On the other hand, line search methods proceed as follows: get a point x_k, find a step direction d_k and search a suitable steplength α_k along this direction. Then the line search procedure generates a new point as $x_{k+1} = x_k + \alpha_k d_k$. In steplength computation process, the ideal choice would be the global minimizer of the following univariate function

$$\phi(\alpha) = f(x_k + \alpha d_k). \tag{2}$$

The traditional monotone line search approaches depart from x_k and then find a steplength α_k along the direction d_k such that descent condition $\phi(\alpha_k) < \phi(0)$ holds. The approach that is used to find a steplength α_k have been called the line search. The exact line search framework, for finding a steplength α_k, can be summarized in solving the following one dimensional optimization problem

$$\varphi(\alpha_k) = \text{Minimize } \phi(\alpha), \quad \text{subject to } \alpha > 0. \tag{3}$$

Firstly, when the iterate x_k is far from the solution of the problem, it is not logical to solve this equation exactly. Secondly, by solving (3) exactly, we get the maximum benefit from the direction d_k, but an exact minimization leads us to solve a nonlinear equations which is expensive and unnecessary, especially for large-scale problems. Finally, in practice, we need that a steplength α_k guarantees a sufficient reduction in function values that induces the global convergence properties of the approach. Therefore, some inexact conditions for determining an acceptable steplength α_k have been proposed, namely the Armijo condition, the Wolfe condition and the Goldstein condition. Among all of these rules, the Armijo rule is the most popular condition that can be stated as follows

$$\phi(\alpha_k) \leq \phi(0) + \delta \alpha_k \phi'(0), \tag{4}$$

where $\delta \in (0, 1)$. For the sake of simplicity, we abbreviate $f(x_k)$ by f_k, $\nabla f(x_k)$ by g_k, $\nabla^2 f(x_k)$ by G_k, and any approximation of Hessian matrix by B_k. Hence, we can rewrite (4) by

$$f(x_k + \alpha_k d_k) \leq f_k + \delta \alpha_k g_k^T d_k. \tag{5}$$

From (5) and the descent condition of the search direction d_k, $g_k^T d_k < 0$, we can deduce that $f(x_{k+1}) < f(x_k)$, so the traditional Armijo rule guarantees the monotonicity of the sequence $\{f_k\}$.

In 1982, Chamberlain et al. in [9] proposed a watchdog technique for constrained optimization, in which some standard line search conditions were relaxed to overcome the Marotos effect. Motivated by this idea, Grippo, Lampariello and Lucidi in [12] presented a nonmonotone Armijo-type line search technique for the Newton method. They also proposed a truncated Newton method with nonmonotone line search for unconstrained optimization, see [13]. In their nonmonotone Armijo-type line search, a steplength α_k is accepted if it satisfies the following condition

$$f(x_k + \alpha_k d_k) \leq f_{l(k)} + \delta \alpha_k g_k^T d_k, \tag{6}$$

where $\delta \in (0, 1)$ and

$$f_{l(k)} = \max_{0 \leq j \leq m(k)} \{f_{k-j}\}, \quad k = 0, 1, 2, \cdots, \tag{7}$$

where $m(0) = 0$ and $0 \leq m(k) \leq min\{m(k-1)+1, N\}$ with $N \geq 0$. Their conclusions are overall favorable, especially when are applied to very nonlinear problems in the presence of a narrow curved valley. Nonmonotone techniques have been distinguished by the fact that they do not enforce strict monotonicity to the objective function values at successive iterates. It has been proved that nonmonotone techniques can improve both the possibility of finding the global optimum and the convergence rate of the sequence generated by these procedures [1, 2, 4, 12, 14, 20, 21, 24]. Inspired by these interesting properties many authors have focused in employing nonmonotone techniques in wide variety of optimization areas.

Although the traditional nonmonotone line search technique (6) has many advantages, this rule also consists of some drawbacks as well (see for example [10, 24]). Some of these can be listed as follows:

- Although the method is generating R-linearly convergent iterates for strongly convex function, the iterates may not satisfy the condition (6) for sufficiently large k, with any fixed bound N on the memory.
- A good function value generated in any iterate is essentially discarded due to the max term in (7).
- In some cases, the numerical results is very dependent on the choice of N.

There are some proposals to overcome these disadvantages, see [3, 15, 22–24], which have introduced a new formula instead of $f_{l(k)}$ in (6). Theoretical and computational results have indicated that these reformation can improve the efficiency of the Armijo-type line search techniques. In the next section, we introduce a new nonmonotone strategy inheriting the outstanding results properties of traditional nonmonotone strategy and improve its computational results due to an appropriate use of current information of function value.

The rest of paper is organized as follows: In Section 2, we introduce a new nonmonotone strategy and then investigate the global convergence along with R-linear convergence rate of the proposed algorithm under some classical assumptions. Preliminarily numerical results in Section 3 indicate that the proposed algorithm is promising. Finally, some conclusions are delivered in Section 4.

2 New algorithms: motivation and theory

We first introduce a new nonmonotone strategy, draw it in algorithmic framework, investigate some convergence properties of the proposed nonmonotone strategy for the Armijo-type line search procedure and then prove the global convergence to first-order stationary points under suitable conditions. Since the new nonmonotone strategy can be regarded as a variant of the nonmonotone strategy of Grippo et al. [12], we expect similar properties and significant similarities in their proof for line search method. We also establish R-linear convergence rate for the proposed algorithm in the sequel.

It is known that the best convergence results can be obtained by a stronger nonmonotone strategy when iterates are far from the optimum, and by a weaker nonmonotone technique whenever iterates are close to the optimum, (see [24]). Furthermore, we believe that the traditional nonmonotone strategy (4) just uses the current function value f_k in the calculation of the max term so that the prominent information f_k is almost ignored. Hence it seems that, near to the optimum, the traditional nonmonotone technique has not shown a suitable behavior, compared with the monotone version. On the other hand, the maximum feature is one of the most important information factors among the recent successful iterates, and we do not attend to lost this valuable information. In order to overcome the above mentioned disadvantages and introduce a more relaxed nonmonotone strategy, we define

$$R_k = \eta_k f_{l(k)} + (1 - \eta_k) f_k, \tag{8}$$

where $0 \leq \eta_{min} \leq \eta_{max} \leq 1$ and $\eta_k \in [\eta_{min}, \eta_{max}]$. It is obvious that one can obtain a stronger nonmonotone strategy whenever η_k is close to 1 and can obtain a weaker nonmonotone strategy whenever η_k is close to 0. Therefore, by choosing an adaptive η_k, the approach not only can increase the effect of $f_{l(k)}$ far from the optimum but also can reduce it close to the optimum. Then the new Armijo-type line search can be defined by

$$f(x_k + \alpha_k d_k) \leq R_k + \delta \alpha_k g_k^T d_k, \tag{9}$$

where d_k is a descent direction.

Now, we can outline our new nonmonotone Armijo-type line search algorithm as follows:

Note that if in Algorithm 1 one sets $N = 0$ or $\eta_k = 0$ for arbitrary $k \in \mathbf{N}$, Algorithm 1 reduces to the traditional Armijo line search algorithm.

Throughout this paper, we consider the following assumptions in order to analyze the convergence results of the proposed algorithm:

(H1) The level set $L(x_0) = \{x \in \mathbf{R}^n \,|\, f(x) \leq f(x_0),\ x_0 \in \mathbf{R}^n\}$ is bounded.

(H2) The gradient $g(x)$ of $f(x)$ is Lipschitz continuous over an open convex S that contains $L(x_0)$; i.e., there exists a positive constant L such that

$$\|g(x) - g(y)\| \leq L\|x - y\|,$$

for all $x, y \in S$.

Algorithm 1: New nonmonotone Armijo-type line search algorithm (NMLS-N)

Input: $x_0 \in \mathbf{R}^n$, $0 < \rho < 1$, $0 < \delta < \frac{1}{2}$, $0 < \gamma_1 < s < \gamma_2$, $0 \le \eta_{min} \le \eta_0 \le \eta_{max} \le 1$, $\epsilon > 0$, $N \ge 0$.
Begin
  $k \leftarrow 0$;
  Compute $R_0 = f(x_0)$;
  While ($\|g_k\| \ge \epsilon$) {**Start of outer loop**}
    {**Determination of search ditrction**}
    Generate a descent direction d_k;
    $\alpha = s$;
    While ($f(x_k + \alpha d_k) > R_k + \delta\alpha^T d_k$) {**Start of backtraking loop**}
      $\alpha \leftarrow \rho\alpha$;
    End While {**End of inner loop**}
    $\alpha_k \leftarrow \alpha$;
    $x_{k+1} \leftarrow x_k + \alpha_k d_k$;
    Choose $\eta_{k+1} \in (\eta_{min}, \eta_{max})$;
    Determine $f_{l(k+1)}$ by (7);
    Compute R_{k+1} by (8);
    $k \leftarrow k + 1$;
  End While {**End of outer loop**}
End

Furthermore, in order to guarantee the global convergence of the iterative scheme $x_{k+1} = x_k + \alpha_k d_k$, we need that a direction d_k satisfies the following sufficient descent conditions

$$g_k^T d_k \le -c_1 \|g_k\|^2 \tag{10}$$

and

$$\|d_k\| \le c_2 \|g_k\|, \tag{11}$$

where c_1 and c_2 are two positive real-valued constants.

Since $f_k \le f_{l(k)}$, we have

$$f_k \le \eta_k f_{l(k)} + (1 - \eta_k) f_k = R_k. \tag{12}$$

Therefore, the right hand side of the proposed line search is is not smaller than the standard Armijo rule. Hence it is possibly permitted to the new algorithm to gain a larger steplength. This fact may reduce the total number of iterates and the total number of function evaluations. In more details,we assume that $\tilde{\alpha}$ and α represent the steplengths satisfying the standard Armijo rule and the new Armijo-type line search procedure respectively. Then from (12), we can get

$$f(x_k + \tilde{\alpha}_k d_k) - R_k \le f(x_k + \tilde{\alpha}_k d_k) - f_k \le \delta\tilde{\alpha}_k g_k^T d_k.$$

This implies that $\tilde{\alpha}$ satisfies the condition (9), so α at least is as large as $\tilde{\alpha}$ suggesting $\tilde{\alpha} \le \alpha$. Using (12), it can be easily seen that

$$\frac{R_k - f(x_k + \alpha d_k) + \delta\alpha g_k^T d_k}{\alpha} \ge \frac{f_k - f(x_k + \alpha d_k) + \delta\alpha g_k^T d_k}{\alpha}.$$

Now, by using the Taylor expansion and recalling that $\|d_k\|$ is bounded we obtain

$$\lim_{\alpha\to 0^+} \frac{f_k - f(x_k+\alpha d_k)+\delta\alpha g_k^T d_k}{\alpha} = \lim_{\alpha\to 0^+} \frac{f_k - \left(f_k+\alpha g_k^T d_k + o(\alpha\|d_k\|)\right)+\delta\alpha g_k^T d_k}{\alpha}$$
$$= -(1-\delta) g_k^T d_k > 0,$$

where the last inequality follows from $0 < \delta < \frac{1}{2}$. Therefore, there exists a steplength $\acute{\alpha}_k > 0$ such that

$$f(x_k+\alpha d_k) \leq R_k + \delta\alpha g_k^T d_k, \quad \text{for all } \alpha \in [0, \acute{\alpha}_k]. \tag{13}$$

Thus, by setting $\acute{\acute{\alpha}}_k = \min\{s, \acute{\alpha}_k\}$, we have

$$f(x_k+\alpha d_k) \leq R_k + \delta\alpha g_k^T d_k, \quad \text{for all } \alpha \in [0, \acute{\acute{\alpha}}_k]. \tag{14}$$

Therefore, the relation (9) and so the backtracking loop of the algorithm are well-defined.

In order to establish the global convergence of the proposed algorithm, the two following results are necessary.

Lemma 1 *Suppose that the condition* (10) *holds and the sequence* $\{x_k\}$ *is generated by Algorithm 1. Then the sequence* $\{f_{l(k)}\}$ *is non-increasing.*

Proof Using the definition R_k and $f_{l(k)}$, we have

$$R_k = \eta_k f_{l(k)} + (1-\eta_k) f_k \leq \eta_k f_{l(k)} + (1-\eta_k) f_{l(k)} = f_{l(k)}. \tag{15}$$

This leads to

$$f(x_k+\alpha_k d_k) \leq R_k + \delta\alpha_k \nabla f(x_k)^T d_k \leq f_{l(k)} + \delta\alpha_k g_k^T d_k. \tag{16}$$

The preceding inequality and the descent condition $g_k^T d_k < 0$ indicate that

$$f_{k+1} \leq f_{l(k)}. \tag{17}$$

On the other hand, from (7), we get

$$\begin{aligned} f_{l(k+1)} &= \max_{0\leq j\leq m(k+1)} \left\{f_{k+1-j}\right\} \\ &\leq \max_{0\leq j\leq m(k)+1} \left\{f_{k+1-j}\right\} = \max\left\{f_{l(k)}, f_{k+1}\right\}. \end{aligned}$$

This fact together with (17) complete the proof. □

Corollary 1 *Suppose that (H1) and* (10) *hold and the sequence* $\{x_k\}$ *be generated by Algorithm 1, then the sequence* $\{f_{l(k)}\}$ *is convergent.*

Proof Lemma 2 and $f_{l(0)} = f_0$ suggest that the sequence $\{x_{l(k)}\}$ remains in level set $L(x_0)$. Since $f(x_k) \leq f(x_{l(k)})$, then the sequence $\{x_k\}$ remains in $L(x_0)$. Now, (H1) together with Lemma 2 imply that the sequence $\{f_{l(k)}\}$ is convergent. □

The subsequent outcome suggests that the sequence $\{f_k\}$ is convergent to an accumulation point of its subsequence $\{f_{l(k)}\}$.

Lemma 2 *Suppose that (H1) and (H2) hold, the direction d_k satisfies* (10) *and* (11) *and the sequence $\{x_k\}$ be generated by Algorithm 1. Then we have*

$$\lim_{k\to\infty} f_{l(k)} = \lim_{k\to\infty} f(x_k). \tag{18}$$

Proof From (7), (9) and (15), for $k > N$, we obtain

$$\begin{aligned} f\left(x_{l(k)}\right) &= f\left(x_{l(k)-1} + \alpha_{l(k)-1}\, d_{l(k)-1}\right) \\ &\le R_{l(k)-1} + \delta\alpha_{l(k)-1}\, g_{l(k)-1}^T d_{l(k)-1} \\ &\le f\left(x_{l(l(k)-1)}\right) + \delta\alpha_{l(k)-1}\, g_{l(k)-1}^T d_{l(k)-1}. \end{aligned}$$

The preceding inequality together with Corollary 1, $\alpha_k > 0$ and $g_k^T d_k < 0$ imply that

$$\lim_{k\to\infty} \alpha_{l(k)-1} g_{l(k)-1}^T d_{l(k)-1} = 0. \tag{19}$$

Using (10) and (11), we have $\alpha_k g_k^T d_k \le -c_1\alpha_k\|g_k\|^2 \le -\left(c_1/c_2{}^2\right)\alpha_k\|d_k\|^2$, for all k. This fact along with $\alpha_k < \gamma_2$ and (19) suggest that

$$\lim_{k\to\infty} \alpha_{l(k)-1}\left\|d_{l(k)-1}\right\| = 0. \tag{20}$$

We now prove that $\lim_{k\to\infty}\alpha_k\|d_k\| = 0$. Let $\hat{l}_k = l(k+N+2)$. First, by induction, we show that, for any $j \ge 1$, we have

$$\lim_{k\to\infty} \alpha_{\hat{l}(k)-j}\left\|d_{\hat{l}(k)-j}\right\| = 0 \tag{21}$$

and

$$\lim_{k\to\infty} f\left(x_{\hat{l}(k)-j}\right) = \lim_{k\to\infty} f\left(x_{l(k)}\right). \tag{22}$$

If $j = 1$, since $\{\hat{l}_k\} \subseteq \{l(k)\}$, the relation (21) directly follows from (20). The condition (21) indicates that $\|x_{\hat{l}(k)} - x_{\hat{l}(k)-1}\| \to 0$. This fact along with the fact that $f(x)$ is uniformly continuous on L_0 imply that (22) holds, for $j = 1$. Now, we assume that (21) and (22) hold, for a given j. Then, using (9) and (15), we obtain

$$\begin{aligned} f\left(x_{\hat{l}(k)-j}\right) &\le R_{\hat{l}(k)-j-1} + \delta\alpha_{\hat{l}(k)-j-1} g_{\hat{l}(k)-j-1}^T d_{\hat{l}(k)-j-1} \\ &\le f\left(x_{l(\hat{l}(k)-j-1)}\right) + \delta\alpha_{\hat{l}(k)-j-1} g_{\hat{l}(k)-j-1}^T d_{\hat{l}(k)-j-1}. \end{aligned}$$

Following the same arguments employed for deriving (20), we deduce

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

This means that

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

This fact together with uniformly continuous property of $f(x)$ on $L(x_0)$ and (22) indicate that

$$\lim_{k\to\infty} f\left(x_{\hat{l}(k)-(j+1)}\right) = \lim_{k\to\infty} f\left(x_{\hat{l}(k)-j}\right) = \lim_{k\to\infty} f\left(x_{l(k)}\right). \tag{23}$$

Thus, we conclude that (21) and (22) hold for any $j \ge 1$.

On the other hand, for any $k \in \mathbf{N}$, we have

$$x_{k+1} = x_{\hat{l}(k)} - \sum_{j=1}^{\hat{l}(k)-k-1} \alpha_{\hat{l}(k)-j} d_{\hat{l}(k)-j}. \tag{24}$$

From definition $l(k)$, we have $\hat{l}(k) - k - 1 = l(k + N + 2) - k - 1 \le N + 1$. Thus, (21) and (24) suggest

$$\lim_{k\to\infty} \left\| x_{k+1} - x_{\hat{l}(k)} \right\| = 0. \tag{25}$$

Since $\{f(x_{l(k)})\}$ admits a limit, it follows from (25) and the uniform continuity of $f(x)$ on $L(x_0)$ that

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

Therefore, (18) holds and the proof is completed. □

Corollary 2 *Suppose that (H1) and (H2) hold, d_k satisfies* (10) *and* (11) *and the sequence $\{x_k\}$ be generated by Algorithm 1. Then we have*

$$\lim_{k\to\infty} R_k = \lim_{k\to\infty} f(x_k). \tag{26}$$

Proof From (12) and (15), we obtain

$$f_k \le R_k \le f_{l(k)}.$$

As a consequence, Lemma 2 completes the proof. □

At this stage, the next result implies that the steplength α_k has a lower bound which is necessary to establish the global convergence of the proposed algorithm.

Lemma 3 *Suppose that (H2) holds and the sequence $\{x_k\}$ be generated by Algorithm 1. Then we have*

$$\alpha_k \ge \min\left\{\gamma_1 \rho, \left(\frac{2(1-\delta)\rho}{L}\right) \frac{\left|g_k^T d_k\right|}{\|d_k\|^2}\right\}. \tag{27}$$

Proof If $\alpha_k/\rho \ge \gamma_1$, then $\alpha_k \ge \gamma_1 \rho$, which gives (27). So we can let $\alpha_k/\rho < \gamma_1$, by the definition α_k and (12), we obtain

$$f(x_k + \alpha_k/\rho\, d_k) > R_k + \delta \frac{\alpha_k}{\rho}\, g_k^T d_k \ge f(x_k) + \delta \frac{\alpha_k}{\rho}\, g_k^T d_k. \tag{28}$$

Using (H2) and Lipschitz continuous property of $g(x)$, we can write

$$\begin{aligned} f(x_k + \alpha d_k) - f(x_k) &= \alpha g_k^T d_k + \int_0^\alpha \left[\nabla f(x_k + t d_k) - \nabla f(x_k)\right]^T d_k\, dt \\ &\le \alpha g_k^T d_k + \int_0^\alpha L\|d_k\|^2\, t\, dt \\ &= \alpha g_k^T d_k + \frac{1}{2} L\alpha^2 \|d_k\|^2. \end{aligned}$$

Setting $\alpha = \alpha_k/\rho$ in the prior inequality and combining it with (28) indicate that (27) holds. This completes the proof. □

Summarizing our theoretical results ensure the global convergence of the algorithm to first-order stationary points that are not local maximum points. More precisely, we wish to prove that, under stated assumptions of this section, all limit points x_* of the generated sequence $\{x_k\}$ by the algorithm are stisfying

$$g(x_*) = 0, \tag{29}$$

irrespective of the position of the starting point x_0.

Theorem 1 *Suppose that (H1) and (H2) hold, the direction d_k satisfies* (10) *and* (11) *and the sequence $\{x_k\}$ is generated by Algorithm 1. Then we have*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{30}$$

Furthermore, there isn't any limit point of the sequence $\{x_k\}$ that be a local maximum of $f(x)$.

Proof We first show

$$f_{k+1} \le R_k - \beta \, \|g_k\|^2, \tag{31}$$

where β is defined by

$$\beta = \min\left\{\delta c_1 \gamma_1 \rho, \frac{2\delta(1-\delta)\rho c_1^2}{L c_2^2}\right\}. \tag{32}$$

If $\alpha_k \ge \rho\gamma_1$, it follows from (9) and (10) that

$$f_{k+1} \le R_k + \delta\alpha_k g_k^T d_k \le R_k - \delta\alpha_k c_1 \|g_k\|^2 \le R_k - \delta c_1 \gamma_1 \rho \|g_k\|^2, \tag{33}$$

which implies that (31) holds.

Now, let $\alpha_k < \rho\gamma_1$. Using (9) and (27), one can obtain

$$f_{k+1} \le R_k - \left(\frac{2\delta(1-\delta)\rho}{L}\right)\left(\frac{g_k^T d_k}{\|d_k\|}\right)^2.$$

Using (10) and (11), we get

$$f_{k+1} \le R_k - \left(\frac{2\delta(1-\delta)\rho c_1^2}{L c_2^2}\right)\|g_k\|^2. \tag{34}$$

This indicates that (31) holds.

By setting β as (32), it follows that $\beta > 0$. Also by (31), we can obtain

$$R_k - f_{k+1} \ge \beta \, \|g_k\|^2 \ge 0.$$

This fact along with Corollary 2 give (30). The proof of this fact that no limit point of $\{x_k\}$ is local maximum of $f(x)$ is similar to proof given by Grippo et al. in [12] so the details are omitted. This completes the proof. □

The primary aim of what follows is to study the convergence rate of the sequence generated by Algorithm 1. Similar to [10], we state the R-linear convergence of the sequence generated by Algorithm 1.

In 2002, Dai in [10] proved the R-linearly convergence rate of the nonmonotone max-based line search scheme (6), when the objective function $f(x)$ is strongly convex [10]. Zhang and Hager in [24] extend this property for their proposed nonmonotone line search algorithm for uniformly convex functions. Motivated by these ideas, similar to Dai in [10], we establish the R-linearly convergence of the sequence generated by Algorithm 1 for strongly convex functions.

Recall that the objective function f is a strongly convex function if there exists a scalar ω such that

$$f(x) \geq f(y) + \nabla f(y)^T(x-y) + \frac{1}{2\omega}\|x-y\|^2, \tag{35}$$

for all $x, y \in \mathbf{R}^n$. In order to establish the R-linearly convergence rate, we need the following lemmas.

Lemma 4 *Suppose that (H1) and (H2) hold, the direction d_k satisfies (10) and (11) and the sequence $\{x_k\}$ be generated by Algorithm 1. Then, for any $l \geq 1$,*

$$\max_{1\leq i\leq N} f(x_{Nl+i}) \leq \max_{1\leq i\leq N} f(x_{N(l-1)+i}) + \delta \max_{0\leq i\leq N-1}\left[\alpha_{Nl+i}\, g_{Nl+i}^T d_{Nl+i}\right]. \tag{36}$$

Proof Using (15), we have

$$\begin{aligned} f(x_{Nl+1}) &\leq R_{Nl} + \delta\alpha_{Nl}\, g_{Nl}^T d_{Nl} \\ &\leq \max_{1\leq i\leq m(Nl)} f(x_{Nl-i}) + \delta\alpha_{Nl}\, g_{Nl}^T d_{Nl}. \end{aligned}$$

The rest of the proof is similar to Lemma 2.1 in [10]. □

Lemma 5 *Suppose that (H1) and (H2) hold, the direction d_k satisfies (10) and (11) and the sequence $\{x_k\}$ be generated by Algorithm 1. Then there exists a constant $c_3 > 1$ such that*

$$\|g_{k+1}\| \leq c_3\|g_k\|. \tag{37}$$

Proof To find a proof, see the Theorem 2.1 in [10]. □

Lemma 6 *Suppose that (H1) and (H2) hold, $f(x)$ be a strongly convex function, the direction d_k satisfies (10) and (11) and the sequence $\{x_k\}$ be generated by Algorithm 1. Then there exist constants $c_4 > 0$ and $c_5 \in (0, 1)$ such that*

$$f(x_k) - f(x_*) \leq c_4 c_5^k\big[f(x_1) - f(x_*)\big], \tag{38}$$

for all $k \in \mathbf{N}$.

Proof Using Lemma 4 and Lemma 5 all conditions of Theorem 3.1 of [10] hold. Therefore, the conclusion can be proved in a similar way. Thus, the details are omitted. □

Theorem 2 *Suppose that (H1), (H2) and* (35) *hold, the direction d_k satisfies* (10) *and* (11) *and the sequence $\{x_k\}$ be generated by Algorithm 1. Then the sequence $\{x_k\}$ converges to the stationary point x_* at least R-linearly.*

Proof Recall that the sequence $\{x_k\}$ converges to x_* R-linearly if there exists a sequence of nonnegative scalars $\{\nu_k\}$ such that, for all $k \in \mathbf{N}$,

$$\|x_k - x_*\| \leq \nu_k, \tag{39}$$

where the sequence $\{\nu_k\}$ converges Q-linearly to zero. We first introduce a sequence $\{\nu_k\}$, then prove its Q-linearly convergence. Lemma 5 together with substituting $y = x_*$ and $x = x_k$ in (35) imply that

$$\|x_k - x_*\|^2 \leq 2\omega(f(x_k) - f(x_*)) \leq \left[2\omega c_4(f(x_1) - f(x_*))\right]{c_5}^k = r{c_5}^k, \tag{40}$$

where $r = [2\omega c_4(f_1 - f_*)]$. By setting $\nu_k = r{c_5}^k$, we get that $\nu^* = 0$. We also have

$$\lim_{k\to\infty} \frac{\nu_{k+1} - \nu_*}{\nu_k - \nu_*} = c_5 < 1. \tag{41}$$

Therefore, the sequence $\{x_k\}$ converges to x_* at least R-linearly. □

It is known that while the the Newton method has the quadratic convergence rate close to the optimum, the quasi-Newton approaches can take the superlinear convergence rate on some suitable conditions, see [18]. It is not hard to show that the present algorithm can reduced to the Newton methods or the quasi-Newton approaches similar to what established in Nocedal and Wright in [18] under some classical assumptions.

3 Preliminary computational experiments

This section reports extensive numerical results obtained by testing the proposed algorithm, NMLS-N, compared with the standard Armijo line search in [7], MLS-A, the nonmonotone line search of Grippo et al. in [12], NMLS-G, and the nonmonotone line search of Hager and Zhang in [24], NMLS-H. We provide three different classes of directions in our comparisons, namely the Barzilai-Borwein, LBFGS and truncated Newton (TN) directions. First part of our comparisons includes using the recently proposed modified two-point stepsize gradient direction of Babaie-Kafaki and Fatemi, [8] for all algorithms. In the second part, the search direction d_k is determined by the well-known limited quasi-Newton approach L-BFGS [16, 19], and in the third part, the search direction d_k is computed by the truncated Newton algorithm proposed in Chapter 6 of [18] with some reformations.

The computational results exploit standard unconstrained test functions from Andrei in [5] and Moré et al. in [17]. The starting points are the standard ones provided by the mentioned papers. We perform our experiments in double precision arithmetic format ine MATLAB 7.4 programming environment. All codes are written in the same subroutine where computes the steplength α_k by the variant Armijo-type conditions with the identical parameters $s = 1$, $\delta = 10^{-4}$ and $\rho = 0.5$ for the modified Barzilai-Borwein and truncated Newton directions and $s = 1$, $\delta = 10^{-3}$ and $\rho = 0.5$ for the LBFGS direction, which are selected respectively the same as what proposed in [8] and [12]. Like NMLS-G, NMLS-N takes advantage of $N = 10$ to calculate the nonmonotone term $f_{l(k)}$. From Algorithm 1, it is clear that the number of iterates and gradient evaluations are the same, so we considered the number of iterates and function evaluations to compare the algorithms.

3.1 Implementations including a modified Barzilai-Borwein direction

This subsection reports the results of the considered directions where the search directions are generated by the modified two-point stepsize gradient algorithm in [8] on a set of 107 unconstrained optimization test problems. We rename the considered algorithms by MLS-A1, NMLS-G1, NMLS-H1 and NMLS-N1, respectively. We here briefly summarize how the search direction, $\mathbf{d_k} = -\lambda_\mathbf{k}\mathbf{g_k}$, is generated by the following procedure:

Procedure 1: Calculation of direction $\mathbf{d_k} = -\lambda_\mathbf{k}\mathbf{g_k}$

Begin
  If $k = 1$
    $\lambda_1 \leftarrow \|g_k\|_\infty^{-1}$;
  Else
    Select $r > 0$, $C > 0$,
    $\vartheta \leftarrow 10^{-5}$; $\varepsilon \leftarrow 10^{-30}$; $s_{k-1} \leftarrow x_k - x_{k-1}$; $y_{k-1} \leftarrow g_k - g_{k-1}$;
    $h_{k-1} \leftarrow C + \max\left\{-\frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2}, 0\right\} \|g_k\|^{-r}$; $\bar{y}_{k-1} \leftarrow y_{k-1} + h_{k-1}\|g_k\|^r s_{k-1}$;
    $\bar{\lambda}_k \leftarrow \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T \bar{y}_{k-1}}$; $\tilde{\lambda}_k \leftarrow \frac{s_{k-1}^T s_{k-1}}{6(f_{k-1}-f_k)+4g_k^T s_{k-1}+2g_{k-1}^T s_{k-1}}$;
    If $s_{k-1}^T y_{k-1} < \vartheta$ **or** $f_k - \left(f_{k-1} + g_{k-1}^T s_{k-1}\right) < \vartheta$ **or** $\tilde{\lambda}_k < 0$ **then**
      $\lambda_k \leftarrow \max\left\{\varepsilon, \min\left\{\frac{1}{\varepsilon}, \tilde{\lambda}_k\right\}\right\}$
    Else
      $\lambda_k \leftarrow \max\left\{\varepsilon, \min\left\{\frac{1}{\varepsilon}, \bar{\lambda}_k\right\}\right\}$
    End
  End
End

Our preliminary numerical experiments have showed that the best convergence results are obtained by η_k close to 1, whenever the iterates are far from the optimum, and by η_k close to 0, whenever the iterates are close to the optimum. It is well-known that, in optimization areas, the best criterion to measure the closeness of the current point x_k to the optimum x_* is to assess the first-order optimality condition, so

$\|g_k\|_\infty \leq 10^{-3}$ can be used as a criteria for the closeness to the optimum. Therefore, NMLS-N exploits the the starting parameter $\eta_0 = 0.95$ and update it by

$$\eta_{k+1} = \begin{cases} \frac{2}{3}\eta_k + 0.01 & \|g_k\|_\infty \leq 10^{-3}, \\ \max\left\{\frac{99}{100}\eta_k, 0.5\right\} & otherwise. \end{cases} \tag{42}$$

We easily can see that the algorithm for problems with the large number iterates, more than 65 iterates, starts with $\eta_0 = 0.95$ and slightly decrease it in about 65 iterates to receive $\eta_k \approx 0.5$ and then preserves $\eta_k = 0.5$ unless the condition $\|g_k\|_\infty \leq 10^{-3}$ holds. After getting this condition, η_k will be decreased quickly by the formula $\eta_k = \frac{2}{3}\eta_{k-1} + 0.01$ to finally fixed about $\eta_k = 0.03$. On the other hand, for problems with the total iterates less than 65, the algorithm begins with $\eta_0 = 0.95$ and slightly decreases it to eventually the condition $\|g_k\|_\infty \leq 10^{-3}$ holds and then decline η_k quickly due to $\eta_k = \frac{2}{3}\eta_{k-1} + 0.01$. Therefore, the algorithm, in both cases, starts with a stronger nonmonotone strategy whenever the iterates are far from the optimum and employs a slightly weaker technique in middle of performance and finally take advantages of a weaker nonmonotone technique close to the optimum, when the condition $\|g_k\|_\infty \leq 10^{-3}$ holds. For the algorithm NMLS-H, we also select $\eta_0 = 0.85$ as proposed by Zhang and Hager in [24]. Furthermore, in our implementations the algorithms stop if

$$\|g_k\|_\infty \leq 10^{-6}(1 + |f_k|)$$

except for problem E. Hiebert, which will stop at k = 0 with this criterion. For this problem, the stopping criterion is

$$\|g_k\|_\infty \leq 10^{-6}\|g_0\|_\infty$$

or the number of iterates exceeds 40000. An “Fail”in the tables means that the corresponding algorithm fails to find the problems optimum because the number of iterations exceeds 40000.

The obtained results are shown in Table 1, where we report the number of iterations (n_i) and the number of function evaluations (n_f).

The results of Table 1 suggest that the proposed algorithm has promising behaviour encountering with medium-scale and large-scale unconstrained optimization problems and it is superior to other considered algorithms in the most cases. For this collection of methods, the obtained results of Table 2 indicate the percentage of the test problems in which a method is the fastest.

In 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 n_i and n_f, have been respectively assessed in Figs. 1 and 2 by applying the performance profile proposed from Dolan and Moré in [11]. In the procedure of Dolan and Moré, the profile of each code is measured considering the ratio of its computational outcome versus the best numerical outcome of all codes. This profile offers a tool for comparing the performance of iterative processes in statistical structure. In particular, let $\mathcal{S}$ is set of all

Table 1 Numerical results with a modified Barzilai-Borwein direction

Prob. name	Dim	MLS-A1 n_i/n_f	NMLS-G1 n_i/n_f	NMLS-H1 n_i/n_f	NMLS-N1 n_i/n_f
Beale	2	66/137	48/76	46/75	47/78
Bro. b. scaled	2	Fail	Fail	Fail	Fail
CUBE	2	237/669	138/566	73/168	101/354
Full Hess. FH1	2	94/227	97/174	91/162	87/140
Full Hess. FH2	2	14/19	15/18	19/22	15/18
Powell b. scal.	2	Fail	1587/21114	4169/63231	1227/10770
Helical valley	3	346/747	150/249	114/199	147/246
Gaussian func.	3	5/15	5/15	5/15	5/15
Box three-dim.	3	878/2224	35/50	176/310	35/50
Gulf r. and dev.	3	12237/40973	1630/5676	1968 6445	5602/18258
Bro. a. Dennis	4	74/133	118/176	74/107	97/143
Wood	4	1679/41885	319/592	714/2442	298/604
Biggs EXP6	6	4677/12697	993/1999	1106/2233	2629/6207
Staircase 1	8	166/382	147/513	169/749	73/110
Staircase 2	8	285/667	105/257	124/369	73/110
HARKERP2	10	3027/7465	1097/1998	1097/1998	1403/2922
Penalty I	15	243/747	40/54	38/67	38/52
Variably dim.	20	1/2	1/2	1/2	1/2
Watson	31	Fail	Fail	Fail	Fail
Penalty II	40	625/1780	269/731	308/1149	191/477
POWER	40	594/1339	873/1521	586/1059	557/1125
DIXON3DQ	100	1029/2366	1389/2609	1269/2397	1164/2508
E. Rosenbrock	100	74/176	69/171	71/178	66/169
GENHUMPS	100	21388/312963	26667/75691	30873/73364	18114/49126
SINQUAD	100	Fail	281/1606	214/1506	165/1581
FLETCHCR	200	2943/6856	3300/5282	3375/5901	2867/5849
ARGLINB	500	Fail	118/1891	146/724	65/378
E. Hiebert	500	Fail	Fail	Fail	39482/406023
G. W. a. Holst	500	38223/83761	35726/70095	34782/66675	33792/65893
BDQRTIC	1000	58/87	56/69	66/82	55/69
BIGGSB1	1000	9047/22706	22189/38256	24403/45306	7455/17346
G. Rosenbrock	1000	29030/64082	25957/50930	24445/46074	24422/46553
Par. per. quad.	1000	394/902	302/439	397/542	403/784
POWER	1000	12622/12643	12622/12643	10933/10955	11258/11374
Trigonometric	1000	79/146	78/102	73/95	66/101
EG2	3000	Fail	2043/24239	Fail	491/5291
E. Penalty	3000	9551/9553	9550/9551	9550/9551	9550/9551
Per. quad.	3000	870/1976	988/1703	758/1366	777/1579
Quad. QF1	3000	1189/2716	1186/2059	1491/2718	1003/1905

Table 1 (continued)

Prob. name	Dim	MLS-A1 n_i/n_f	NMLS-G1 n_i/n_f	NMLS-H1 n_i/n_f	NMLS-N1 n_i/n_f
Alm. per. quad.	5000	1520/3569	1971/3573	1738/3224	1497/2994
EDENSCH	5000	14/18	12/13	12/13	12/13
E. Powell	5000	4185/10285	277/650	295/632	2804/6423
E. Wood	5000	51/66	52/60	52/60	52/60
E. W. a. Holst	5000	361/1009	142/375	132/363	133/339
NONDQUAR	5000	8397/20860	5191/9094	7136/13716	11761/27874
Per. quad. diag.	5000	3140/8888	924/1765	951/1954	1770/5062
Per. trid. quad.	5000	1176/2767	2641/4760	1987/3716	1792/3684
TRIDIA	5000	8726/22021	12671/24084	15282/28908	16918/35958
E. Beale	6000	82/163	74/118	57/92	55/82
DIXMAANA	9000	25/26	25/26	25/26	25/26
DIXMAANB	9000	23/24	23/24	23/24	23/24
DIXMAANC	9000	25/26	25/26	25/26	25/26
DIXMAAND	9000	27/28	27/28	27/28	27/28
DIXMAANE	9000	858/1985	1143/1908	1029/1845	1441/3189
DIXMAANF	9000	939/2221	1185/1925	1217/2089	961/2056
DIXMAANG	9000	724/1595	1056/1686	908/1544	1176/2585
DIXMAANH	9000	754/1727	684/1102	1620/2867	1834/4091
DIXMAANI	9000	4731/11772	10356/17848	7794/14362	9009/21052
DIXMAANJ	9000	454/989	623/1029	726/1258	599/1223
DIXMAANK	9000	478/972	489/781	644/1059	451/871
DIXMAANL	9000	377/805	380/586	426/674	317/626
ARWHEAD	10000	3/4	3/4	3/4	3/4
BDEXP	10000	17/18	17/18	17/18	17/18
Broyden trid.	10000	37/50	98/110	51/55	79/89
COSINE	10000	8/9	8/9	8/9	8/9
Diagonal 2	10000	294/578	203/324	361/694	258/445
Diagonal 3	10000	23/29	20/24	21/24	20/24
Diagonal 4	10000	17/18	17/18	17/18	17/18
Diagonal 5	10000	5/6	5/6	5/6	5/6
Diagonal 7	10000	9/10	9/10	9/10	9/10
Diagonal 8	10000	8/10	8/10	8/10	8/10
DQDRTIC	10000	171/325	85/112	102/141	87/117
ENGVAL1	10000	22/24	21/22	21/22	21/22
E. BD1	10000	20/24	17/18	17/18	17/18
E. Cliff	10000	9/11	11/12	11/12	11/12
E. DENSCHNB	10000	19/20	19/20	19/20	19/20
E. DENSCHNF	10000	20/23	19/20	19/20	19/20
E. EP1	10000	4/10	4/10	4/10	4/10

Table 1 (continued)

Prob. name	Dim	MLS-A1 n_i/n_f	NMLS-G1 n_i/n_f	NMLS-H1 n_i/n_f	NMLS-N1 n_i/n_f
E. Himmelblau	10000	24/26	24/25	24/25	24/25
E. Maratos	10000	205/459	40/77	45/109	40/77
E. PSC1	10000	26/27	26/27	26/27	26/27
E. QP1	10000	1/2	1/2	1/2	1/2
E. QP2	10000	1/2	1/2	1/2	1/2
E. TET	10000	11/15	12/15	12/15	12/15
E. trid. 1	10000	111/265	46/83	53/92	49/96
E. trid. 2	10000	6/11	6/11	6/11	6/11
E. F. a. Roth	10000	390/714	438/831	414/736	378/642
FLETCBV3	10000	516/517	516/517	516/517	516/517
Full Hess. FH3	10000	4/5	4/5	4/5	4/5
G. PSC1	10000	30/31	30/31	30/31	30/31
G. quartic	10000	12/13	12/13	12/13	12/13
G. trid. 1	10000	13/15	13/14	13/14	13/14
G. trid. 2	10000	47/66	77/93	83/95	71/88
Hager	10000	36/37	36/37	36/37	36/37
HIMMELBG	10000	48/49	48/49	48/49	48/49
HIMMELH	10000	13/15	14/15	14/15	14/15
INDEF	10000	185/1410	28/30	65/91	28/30
LIARWHD	10000	1493/4277	371 1225	312/981	311/953
MCCORMCK	10000	6948/8035	9951/10149	9534/9731	9951/10149
NONDIA	10000	6079/21233	16/51	16/53	16/52
NONSCOMP	10000	50/68	91/113	79/94	81/99
QUARTC	10000	1/2	1/2	1/2	1/2
Quad. QF2	10000	1791/4334	2927/5674	2588/4938	2045/3956
Raydan 1	10000	22/29	23/28	23/28	23/28
Raydan 2	10000	1/2	1/2	1/2	1/2
SINCOS	10000	26/27	26/27	26/27	26/27
VARDIM	10000	1/2	1/2	1/2	1/2

Table 2 Comparing the results of Table 1

	MLS-A1	NMLS-G1	NMLS-H1	NMLS-N1
Iterates (n_i)	48.6 %	50.5 %	43.8 %	65.7 %
Function evaluations (n_f)	39 %	61 %	52.4 %	62.8 %

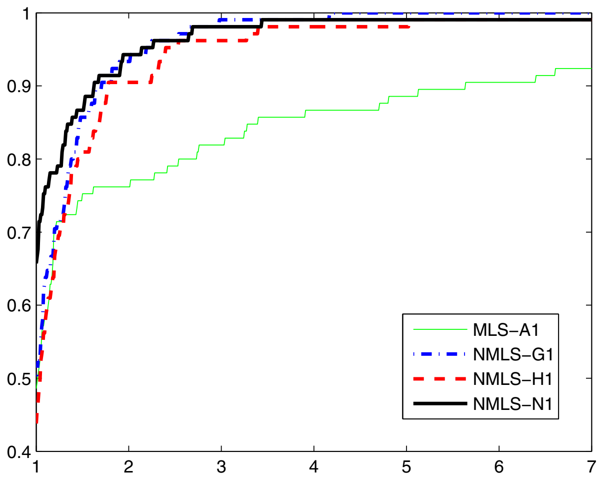

Fig. 1 Iteration performance profiles for a modified Barzilai-Borwein direction

algorithms and $\mathcal{P}$ is a set of test problems, with n_s solvers and n_p problems. For each problem p and solver s, $t_{p,s}$ is the computation result regarding to the performance index. Then, the following performance ratio is defined

$$r_{p,s} = \frac{t_{p,s}}{\min\left\{t_{p,s} : s \in \mathcal{S}\right\}}.$$

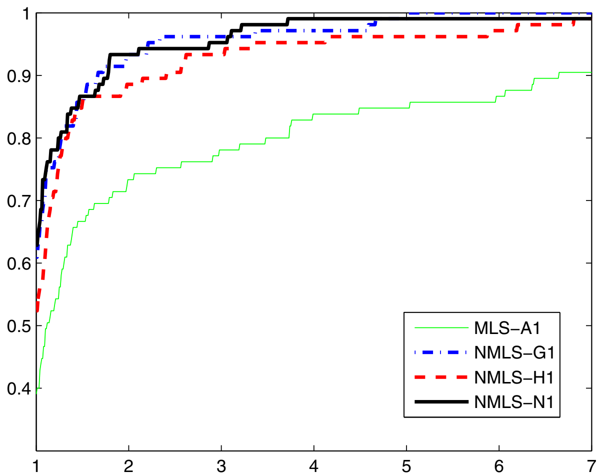

Fig. 2 Function evaluations performance profiles for a modified Barzilai-Borwein direction

If algorithm s is not convergent for a problem p, the procedure sets $r_{p,s} = r_{fail}$, where r_{fail} should be strictly larger than any performance ratio (33). For any factor τ, the overall performance of algorithm s is given by

$$\rho_s(\tau) = \frac{1}{n_p}\text{size}\left\{ p \in \mathcal{P} : r_{p,s} \le \tau \right\}.$$

In fact $\rho_s(\tau)$ is the probability of algorithm $s \in \mathcal{S}$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in \mathbf{R}^n$ of the best possible ratio. The function $\rho_s(\tau)$ is the distribution function for the performance ratio. Especially, $\rho_s(1)$ gives the probability that algorithm s wins over all other algorithms, and $\lim_{\tau \to r_{fail}} \rho_s(\tau)$ gives the probability of that algorithm s solve a problem. Therefore, this performance profile can be considered as a measure of the efficiency and the robustness among the algorithms. In Figs. 1, 2, 3 and 4, the x-axis shows the number τ while the y-axis inhibits $P(r_{p,s} \le \tau : 1 \le s \le n_s)$.

In one hand, Fig. 1 compares the mentioned algorithms in the sense of the total number of iterates. It can be easily seen that NMLS-N1 is the best algorithm in the sense of the most wins on more than 65 % of the test functions. One also can see that NMLS-N1 solves approximately all test functions. On the other hand, Fig. 2 represents a comparison among the considered algorithms regarding the total number of function evaluations. The results of Fig. 2 indicate that the performance of NMLS-N1 is better than other present algorithms. In details, the new algorithm is the best algorithm on more than 62 % of all cases. Further more one can observe that the results of NMLS-G1 and NMLS-N1 is approximately the same regarding the number of function evaluations.

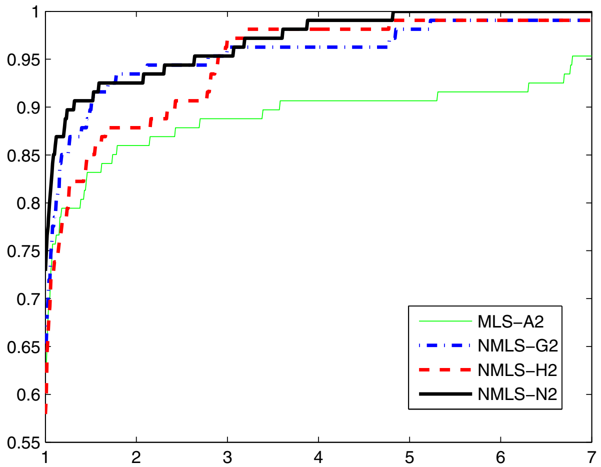

Fig. 3 Iteration performance profiles for the LBFGS direction

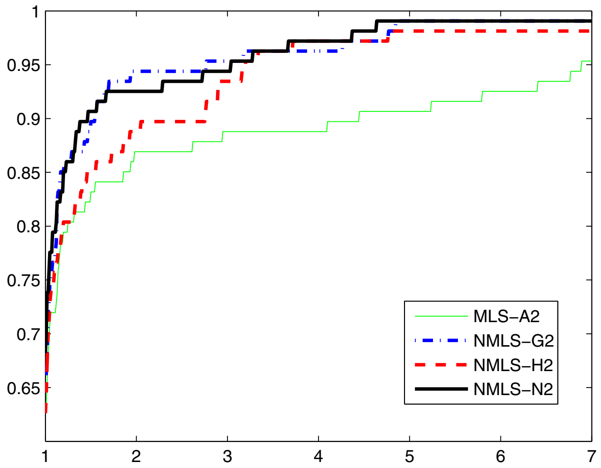

Fig. 4 Function evaluations performance profiles for the LBFGS direction

3.2 Implementations with the LBFGS direction

In this subsection, we implement Algorithm 1 on a set of 107 unconstrained optimization test problems used in previous subsection when the employed direction is a limited memory quasi-Newton direction, namely LBFGS. We rename the considered algorithms by MLS-A2, NMLS-G2, NMLS-H2 and NMLS-N2, respectively. This direction is determined by the following quasi-Newton formula

$$d_k = -H_k g_k,$$

where H_k is a quasi-Newton approximation of the inverse matrix G_k^{-1} generated by the well-known LBFGS approach developed by Nocedal in [19] and Liu and Nocedal in [16]. Let H_0 be a symmetric and positive definite starting matrix and $m = \min\{k, 5\}$. Then the limited memory version of H_k is defined by

$$\begin{aligned} H_{k+1} &= \left(V_k^T \cdots V_{k-m}^T\right) H_0(V_{k-m} \cdots V_k) \\ &\quad + \rho_{k-m}\left(V_k^T \cdots V_{k-m+1}^T\right) s_{k-m} s_{k-m}^T (V_{k-m+1} \cdots V_k) \\ &\quad + \rho_{k-m+1}\left(V_k^T \cdots V_{k-m+2}^T\right) s_{k-m+1} s_{k-m+1}^T (V_{k-m+2} \cdots V_k) \\ &\quad \vdots \\ &\quad + \rho_k s_k s_k^T. \end{aligned}$$

where $\rho_k = 1/y_k^T s_k$ and $V_k = I - \rho_k y_k s_k^T$. The LBFGS code is available from the web page http://www.ece.northwestern.edu/~nocedal/software.html.

We are rewritten this code in MATLAB and exploit it to generate the search direction d_k.

Similar arguments raised in Section 3.1, in the algorithm NMLS-N, parameterη_k is initially set with $\eta_0 = 0.90$ and then be updated as follows

$$\eta_{k+1} = \begin{cases} \frac{2}{3}\eta_k + 0.01 & \|g_k\| \leq 10^{-2}, \\ \max\left\{\frac{99}{100}\eta_k, 0.5\right\} & otherwise. \end{cases}$$

For the algorithm NMLS-H, we also select $\eta_0 = 0.85$ as proposed by Zhang and Hager in [24]. For all algorithms, stopping criterion is

$$\|g_k\|_\infty \leq 10^{-6}.$$

or the algorithm stops when the number of iterates exceeds the maximum number of iterates, 40000.

The results obtained are reported in Table 3. In details, these results clearly suggest that the proposed algorithm has promising behaviour encountering with medium-scale and large-scale unconstrained optimization problems and it is superior to other considered algorithms in the most cases. The percentage of most wins for considered algorithms thanks to Table 3 is reported in Table 4. We also demonstrate the obtained results of Table 3 by performance profiles which can be observed in Figs. 3 and 4.

The Fig. 3 compares the mentioned algorithms in the sense of the total number of iterates. It can be easily seen from the that NMLS-N2 is the best algorithm in the sense of most wins on more than 72 % of the test functions. Meanwhile, NMLS-N2 is competitive with NMLS-G2 and NMLS-H2, but in most cases it grows up faster than these algorithms. It means that in the cases that NMLS-N2 has not the best results, its implementation is close to the performance index of the best algorithm. One also can see that NMLS-N2 solves approximately all test functions. Also, Fig. 4 represents a comparison among the considered algorithms regarding the total number of function evaluations. The results of Fig. 4 indicate that the performance of NMLS-N2 is better than other present algorithms. In details, the new algorithm is the best algorithm on more than 68 % of all cases. Therefore, one can conclude that the behaviour of the proposed Armijo-type algorithm with the LBFGS direction is more efficient and robust than the other considered line search algorithms for solving unconstrained optimization problems.

3.3 Implementations including a truncated Newton direction

This subsection reports some computational experiments with a truncated Newton direction (TN) on a set of some unconstrained optimization test problems, where the algorithms are called MLS-A3, NMLS-G3, NMLS-H3 and NMLS-N3. The algorithms are tested on all of 107 test problems that was used for other directions, but for most of the test problems the results are the same. Then in Table 5, we just report the results that the different outputs obtained by the algorithms.

We here briefly summarize how search directions of the truncated Newton method are generated by the following procedure:

The truncated Newton algorithm requires the computation or estimation of matrix-vector products $G_k p_j$ involving the Hessian matrix of the objective function. An

Table 3 Numerical results with the LBFGS direction

Prob. name	Dim	MLS-A2 n_i/n_f	NMLS-G2 n_i/n_f	NMLS-H2 n_i/n_f	NMLS-N2 n_i/n_f
Beale	2	21/26	21/26	21/26	21/26
Bro. b. scaled	2	18/22	18/22	18/22	18/22
Full Hess. FH1	2	31/36	31/36	31/36	31/36
Full Hess. FH2	2	8/11	8/11	8/11	8/11
Powell b. scal.	2	61/140	170/387	172/448	294/611
Helical valley	3	75/123	35/47	44/55	35/47
Gaussian func.	3	5/8	5/8	5/8	5/8
Box three-dim.	3	35/52	35/52	35/52	35/52
Gulf r. and dev.	3	529/799	79/117	90/163	84/116
Staircase 1	4	14/19	14/19	14/19	14/19
Staircase 2	4	14/19	14/19	14/19	14/19
Bro. a. Dennis	4	41/60	41/60	41/60	41/60
Wood	4	46/59	46/59	133/171	121/161
Biggs EXP6	6	186/250	55/61	65/80	55/61
GENHUMPS	10	23/73	23/73	23/73	23/73
Penalty I	10	11185/11293	11399/11612	11306/11422	11269/11489
Penalty II	10	Fail	664/1260	645/1058	442/887
Variably dim.	10	18/40	18/40	18/40	18/40
Watson	31	19823/19892	13994/15596	12120/13834	11082/14260
HARKERP2	50	16764/16773	Fail	Fail	22614/54947
ARGLINB	100	3/41	3/41	3/41	3/41
Diag. 3	100	69/76	69/76	69/76	72/77
E. Rosenbrock	100	170/277	96/159	63/94	78/123
FLETCBV3	100	Fail	38366/39764	22224/22676	7923/8199
SINQUAD	100	3971/10124	349/2276	564/3305	1111/5211
Trigonometric	100	83/100	68/117	69/80	60/76
Diag. 2	500	120/121	120/121	120/121	114/118
DIXON3DQ	1000	2648/2651	2648/2651	2648/2651	2498/2532
E. Beale	10000	20/23	20/23	20/23	20/23
Fletcher	1000	2553/3237	13350/13807	5963/6228	3892/4320
G. Rosenbrock	1000	Fail	9946/13003	7932/10183	7982/10146
Par. per. quad.	1000	179/193	179/193	179/193	188/208
Hager	1000	48/52	48/52	48/52	48/52
HIMMELH	1000	34/39	14/21	34/39	29/35
BDQRTIC	1000	111/126	111/126	111/126	123/151
EG2	1000	Fail	133/371	91/239	63/117
POWER	1000	12622/12643	12622/12643	10933/10955	11258/11374
P. trid. quad.	5000	610/623	610/623	610/623	588/607
Alm. per. quad.	5000	599/612	599/612	599/612	579/591

Table 3 (continued)

Prob. name	Dim	MLS-A2 n_i/n_f	NMLS-G2 n_i/n_f	NMLS-H2 n_i/n_f	NMLS-N2 n_i/n_f
p. quad. diag.	5000	1087/1101	1044/1212	1229/1454	975/1351
E. Hiebert	5000	Fail	1886/5439	1912/5938	2287/6347
Fletcher	5000	10936/13305	19439/21858	Fail	39435/169519
G. trid. 2	5000	60/81	59/81	91/126	59/81
DENSCHNB	5000	98/99	69/75	149/279	83/103
DENSCHNF	5000	5568/70359	5747/52868	1736/15712	580/5420
NONSCOMP	5000	1205/1781	1116/1549	3324/4890	2577/4698
POWER	5000	30097/30120	90097/30120	30264/30288	31064/31337
FLETCHCR	5000	Fail	15820/81850	11604/76350	9386/91278
LIARWHD	5000	45/61	45/61	47/65	45/61
CUBE	5000	3648/6664	3660/5590	10451/15413	14131/25914
TRIDIA	5000	2004/2017	2004/2017	2004/2017	1969/1997
DIXMAANA	9000	8/12	8/12	8/12	8/12
DIXMAANB	9000	8/12	8/12	8/12	8/12
DIXMAANC	9000	9/14	9/14	9/14	9/14
DIXMAAND	9000	11/17	11/17	11/17	11/17
DIXMAANE	9000	437/441	437/441	437/441	413/419
DIXMAANF	9000	272/276	272/276	272/276	276/288
DIXMAANG	9000	243/248	243/248	243/248	237/250
DIXMAANH	9000	269/275	269/275	269/275	233/242
DIXMAANI	9000	2517/2521	2517/2521	2517/2521	2761/2814
DIXMAANJ	9000	376/380	376/380	376/380	355/363
DIXMAANK	9000	333/338	333/338	333/338	327/333
DIXMAANL	9000	326/332	326/332	326/332	277/285
ARWHEAD	10000	9/26	9/26	9/26	9/26
BDEXP	10000	26/27	26/27	26/27	26/27
Broyden trid.	10000	50/55	50/55	50/55	50/55
BIGGSB1	10000	18/29	18/29	18/29	18/29
COSINE	10000	27/29	27/29	27/29	27/29
Diagonal 4	10000	5/12	5/12	5/12	5/12
Diagonal 5	10000	5/6	5/6	5/6	5/6
Diagonal 7	10000	5/7	5/7	5/7	5/7
Diagonal 8	10000	5/8	5/8	5/8	5/8
DQDRTIC	10000	12/21	12/21	12/21	12/21
ENGVAL1	10000	45/87	45/87	31/45	31/45
EDENSCH	10000	25/30	25/30	25/30	25/30
E. BD1	10000	15/18	15/18	14/16	15/18
E. Cliff	10000	175/2485	49/202	136/1554	49/202
E. DENSCHNB	10000	6/9	6/9	6/9	6/9

Table 3 (continued)

Prob. name	Dim	MLS-A2 n_i/n_f	NMLS-G2 n_i/n_f	NMLS-H2 n_i/n_f	NMLS-N2 n_i/n_f
E. DENSCHNF	10000	13/22	13/22	13/22	13/22
E. EP1	10000	4/13	4/13	4/13	4/13
E. Himmelblau	10000	11/18	11/18	11/18	11/18
E. Maratos	10000	179/314	203/333	178/277	175/367
E. Powell	10000	38/47	38/47	38/47	38/47
E. Penalty	10000	79/120	79/120	79/120	86/108
E. PSC1	10000	11/17	11/17	11/17	11/17
E. QP1	10000	25/41	25/41	25/41	25/41
E. QP2	10000	203/327	130/238	147/211	117/214
E. TET	10000	9/13	9/13	9/13	9/13
E. Wood	10000	46/59	46/59	147/197	141/216
E. W. a. Holst	10000	111/161	114/167	110/150	77/108
E. trid. 1	10000	17/20	17/20	17/20	17/20
E. trid. 2	10000	30/33	30/33	30/33	31/37
Full Hess. FH3	10000	4/19	4/19	4/19	4/19
G. PSC1	10000	34/41	34/41	34/41	34/41
G. quartic	10000	55/77	38/61	34/39	38/61
G. trid. 1	10000	22/26	22/26	22/26	22/26
G. W. a. Holst	10000	Fail	22647/33690	16254/22580	16253/22990
HIMMELBG	10000	2/3	2/3	2/3	2/3
NONDQUAR	10000	2459/2475	2688/2785	4084/4276	3244/3630
Per. quad.	10000	1/4	1/4	1/4	1/4
QUARTC	10000	41/46	41/46	41/46	41/46
Quad. QF1	10000	924/937	924/937	924/937	924/946
Quad. QF2	10000	922/936	922/936	922/936	899/916
Raydan 2	10000	7/8	7/8	7/8	7/8
SINCOS	10000	11/17	11/17	11/17	11/17
TRIDIA	10000	2715/2729	2715/2729	2715/2729	2903/2942
VARDIM	10000	107/198	107/198	107/198	107/198

Table 4 Comparing the results of Table 3

	MLS-A2	NMLS-G2	NMLS-H2	NMLS-N2
Iterates (n_i)	62.6 %	64.5 %	57.9 %	72.9 %
Function evaluations (n_i)	63.5 %	65.5 %	62.6 %	68.2 %

Table 5 Numerical results of truncated Newton algorithm

Prob. name	Dim	MLS-A3			NMLS-G3			NMLS-H3			NMLS-N3		
		n_i	n_g	n_f	n_i	n_g	n_f	n_i	n_g	n_f	n_i	n_g	n_f
BG2	1000	Fail	Fail	Fail	7587	22730	7594	Fail	Fail	Fail	3481	6819	3509
Biggs EXP6	6	15245	30767	15260	17652	35581	17660	17652	35581	17660	17652	35581	17660
CUBE	1000	31216	62433	31219	31065	62131	31066	31065	62131	31066	31065	62131	31066
Diagonal 1	5000	Fail	Fail	Fail	3800	7698	3814	3800	7698	3814	3800	7698	3814
Diagonal 2	1000	1541	3085	1545	3073	6150	3075	3073	6150	3075	3073	6150	3075
	2000	5050	10103	5055	3538	7081	3541	3538	7081	3541	3538	7081	3541
E. Maratos	1000	2812	5664	2824	1135	2306	1143	161	360	171	161	360	171
	5000	2298	4636	2310	1485	3006	1494	151	340	161	151	340	161
	10000	2384	4808	2396	1469	2974	1478	151	340	161	151	340	161
E. QP2	1000	3420	6847	3431	4193	8390	4208	4193	8390	4208	4193	8390	4208
	5000	1225	2451	1231	1223	2447	1228	1223	2447	1228	1223	2447	1228
	10000	827	1655	832	945	1891	947	945	1891	947	827	1655	832
E. trid. 2	5000	Fail	Fail	Fail	131	263	132	131	263	132	101	203	174
E. W. a. Holst	1000	6553	13108	6557	3388	6777	3390	3388	6777	3390	3388	6777	3390
Fletcher	50	3010	6134	3053	1805	3776	1857	2954	6082	3011	2954	6082	3011
	100	7542	15344	7602	4587	9427	4660	6897	14034	6977	6897	14034	6977
G. trid. 1	5000	Fail	Fail	Fail	47	95	48	47	95	48	47	95	48
G. Rosenbrock	100	18782	47316	18799	19230	48198	19246	19230	48198	19246	19230	48198	19246
G. W. a. Holst	100	30186	62492	30341	29264	60328	29330	29911	61688	30027	29904	61684	30016
GENHUMPS	1000	Fail	Fail	Fail	547	1593	549	607	1713	609	543	1585	545

Table 5 (continued)

Prob. name	Dim	MLS-A3			NMLS-G3			NMLS-H3			NMLS-N3		
		n_i	n_g	n_f	n_i	n_g	n_f	n_i	n_g	n_f	n_i	n_g	n_f
Hager	5000	Fail	Fail	Fail	305	611	309	305	611	309	283	567	293
	5000	51	103	54	72	145	73	72	145	73	72	145	73
NONDQUAR	10000	69	139	73	85	171	86	85	171	86	85	171	86
	4	75	151	79	11	23	12	11	23	12	11	23	12
Staircase 1	6	269	539	274	269	539	274	225	451	226	225	451	226
	10	1495	2991	1500	1449	2899	1450	1449	2899	1450	1449	2899	1450
	4	75	151	79	11	23	12	11	23	12	11	23	12
Staircase 2	6	269	539	274	225	451	226	225	451	226	225	451	226
	10	1495	2991	1500	1449	2899	1450	1449	2899	1450	1449	2899	1450
Trigonometric	100	279	598	297	306	650	321	306	650	321	302	642	335

Procedure 2: Truncated Newton direction (TN)

Given initial parameters $z_0 \leftarrow 0$, $r_0 \leftarrow g_k$, $p_0 \leftarrow -g_k$
Begin
  $\epsilon_k \leftarrow \min(0.5/(k+1), \|g_k\|)\|g_k\|$;
  For $j = 0, 1, 2, \ldots$
    If $p_j^T G_k p_j \leq 0$
      If $j \leftarrow 0$
        $d_k \leftarrow -g_k$;
      Else
        $d_k \leftarrow z_j$;
      End
    End
    $\lambda_j \leftarrow r_j^T r_j / p_j^T G_k p_j$;
    $z_{j+1} \leftarrow z_j + \lambda_j p_j$;
    $r_{j+1} \leftarrow r_j + \lambda_j G_k p_j$;
    If $\|r_{j+1}\| \leq \epsilon_k$
        $d_k \leftarrow z_{j+1}$;
    End
    $\beta_{j+1} \leftarrow r_{j+1}^T r_{j+1} / r_j^T r_j$;
    $p_{j+1} \leftarrow -r_{j+1} + \beta_{j+1} p_j$;
  End
End

estimation of this term can be obtained using the following finite difference scheme proposed in [6]:

$$G_k d_j = \frac{\nabla f\left(x_k + \delta d_j\right) - \nabla f(x_k)}{\delta}, \tag{43}$$

where

$$\delta = \frac{2\sqrt{\varepsilon_m}(1 + \|x_k\|)}{\|d_j\|} \tag{44}$$

and ε_m is the machine epsilon. Similar arguments raised in Section 3.1, in the algorithm NMLS-N, the parameter η_k is initially set to $\eta_0 = 0.95$ and then will be updated by the formula (42). For the algorithm NMLS-H, we also select $\eta_0 = 0.85$ as proposed by Zhang and Hager in [24]. For all algorithms, stopping criterion is

$$\|g_k\|_\infty \leq 10^{-6}\|g_0\|_\infty,$$

Table 6 Comparing the results of Table 6

	MLS-A3	NMLS-G3	NMLS-H3	NMLS-N3
Iterates (n_i)	26.7 %	46.7 %	50 %	66.7 %
Gradient evaluations (n_g[a])	26.7 %	43.4 %	50 %	66.7 %
Function evaluations (n_i)	26.7 %	50 %	53.3 %	63.3 %

[a]The number of gradient evaluations

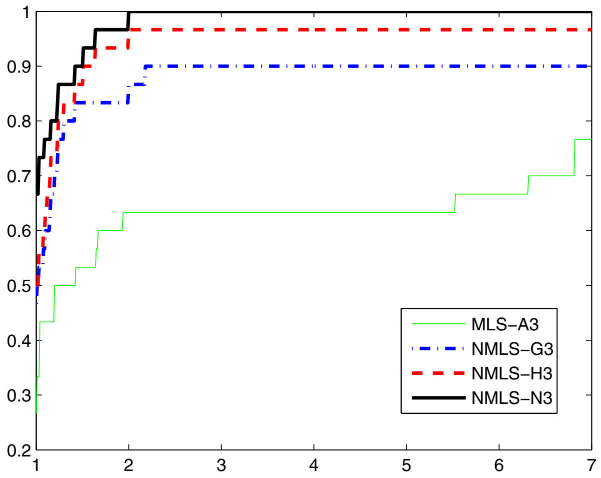

Fig. 5 Iteration performance profiles for the truncated Newton direction

or the algorithm stops when the number of iterates exceeds the maximum number of iterates, 40000.

We now give an overview of the numerical experiments of Table 5. For most of the test problems, the initial step is accepted by the algorithms, which means that the truncated Newton direction satisfies the line search condition with full steplength, $\alpha_k = 1$. Therefore, the results obtained of the algorithms are identical and thus omitted in Table 5. In details, these results suggest that the proposed algorithm has promising behaviour encountering with medium-scale and large-scale unconstrained

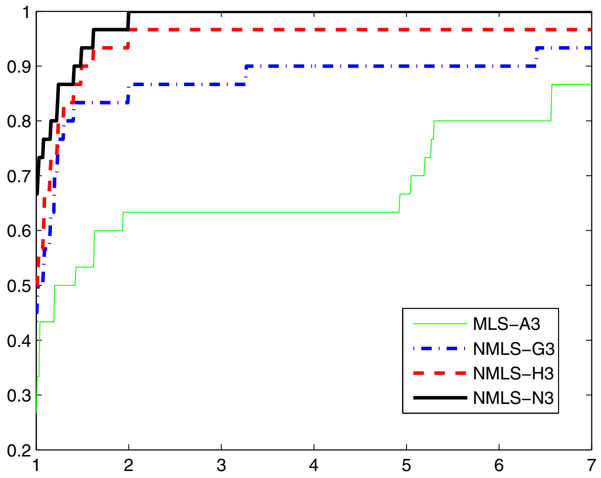

Fig. 6 Gradient evaluations performance profiles for the truncated Newton direction

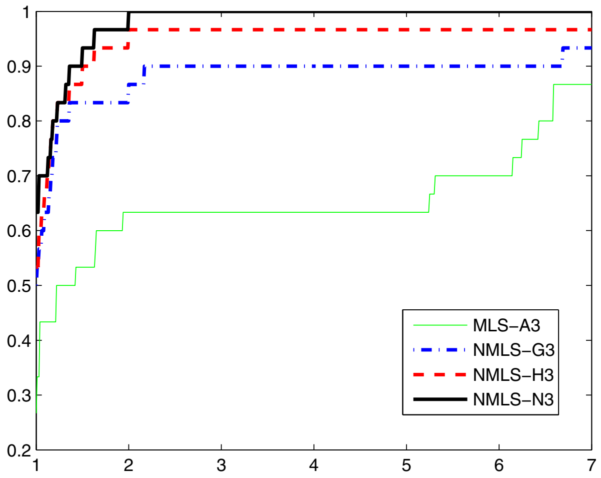

Fig. 7 Function evaluations performance profiles for the truncated Newton direction

optimization problems and it is superior to the other considered algorithms in the most cases. Table 6 shows the percentage of the best results.

The results of this table suggest that NMLS-N3 has better performance in comparison with the other considered algorithms. We also demonstrate the obtained results of Table 6 by performance profiles in Figs. 5, 6 and 7, where respectively compare the number of iterations, gradient evaluations and function evaluations.

Summarizing the results of Figs. 5, 6 and 7 implies that NMLS-N3 is superior to the other presented algorithms respect to the number of iterations, function evaluations and gradient evaluations, However, in many cases the results of all algorithm is the same.

4 Concluding remarks

It is well-known the traditional nonmonotone strategy contains some drawbacks and some efforts in order to overcome theses drawbacks have been done but not enough. Hence, we present a new nonmonotone Armijo-type line search technique for solving unconstrained optimization problems. The introduced nonmonotone strategy takes advantage of a convex combination of the traditional max-term nonmonotone strategy and the current function value to propose a tighter nonmonotone strategy based on effective usage of the current function value. Furthermore, the new line search approach exploits an adaptive technique to make the new nonmonotone strategy stronger far from the optimum and to prepare it weaker close to the optimum. Under some classical assumptions, the approach is convergent to first-order stationary points, irrespective of the chosen starting point. The R-linear convergence rate is also established for strongly convex functions. Preliminary numerical results for the modified Barzilai-Borwein direction, the LBFGS direction and the truncated Newton direction on the large set of standard test functions indicate that the proposed line search technique has efficient performances and promising behavior for solving unconstrained optimization problems.

We believe that there is considerable scope for modifying and adapting the basic ideas presented in this paper. For future works, first of all, other inexact line searches like Wolfe-type or Goldestain-type various can be employed. The next application can be a combination of this strategy with trust-region framework and its variants. Finally, more comprehensive research on finding an adaptive process for the parameter η can be done. It will be a matter of subsequent studies.

Acknowledgments The authors are grateful for the valuable comments and suggestions of two anonymous referees. The authors also would like to thank the research council of University of Vienna and Razi University for supporting this research.

References

1. Ahookhosh, M., Amini, K.: An efficient nonmonotone trust-region method for unconstrained optimization. Numer. Algoritm. **59**(4), 523–540 (2012)
2. Ahookhosh, M., Amini, K.: A nonmonotone trust region method with adaptive radius for unconstrained optimization. Comput. Math. Appl. **60**(3), 411–422 (2010)
3. Ahookhosh, M., Amini, K., Bahrami, S.: A class of nonmonotone Armijo-type line search method for unconstrained optimization. Optimization **61**(4), 387–404 (2012)
4. Ahookhosh, M., Amini, K., Peyghami, M.R.: A nonmonotone trust-region line search method for large-scale unconstrained optimization. Appl. Math. Model. **36**(1), 478–487 (2012)
5. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**(1), 147–161 (2008)
6. Andrei, N.: Accelerated conjugate gradient algorithm with finite difference Hessian/vector product approximation for unconstrained optimization. J. Comput. Appl. Math. **230**, 570–582 (2009)
7. Armijo, L.: Minimization of functions having Lipschitz continuous first partial derivatives. Pac. J. Math. **16**, 1–3 (1966)
8. Babaie-Kafaki, S., Fatemi, M.: A modified two-point stepsize gradient algorithm for unconstrained minimization. Optim. Methods Softw. **1**, 1–11 (2012)
9. Chamberlain, R.M., Powell, M.J.D., Lemarechal, C., Pedersen, H.C.: The watchdog technique for forcing convergence in algorithm for constrained optimization. Math. Program. Stud. **16**, 1–17 (1982)
10. Dai, Y.H.: On the nonmonotone line search. J. Optim. Theory Appl. **112**(2), 315–330 (2002)
11. Dolan, E., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
12. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
13. Grippo, L., Lampariello, F., Lucidi, S.: A truncated Newton method with nonmonotone line search for unconstrained optimization. J. Optim. Theory Appl. **60**(3), 401–419 (1989)
14. Grippo, L., Lampariello, F., Lucidi, S.: A class of nonmonotone stabilization methods in unconstrained optimization. Numer. Math. **59**, 779–805 (1991)
15. Hu, S.L., Huang, Z.H., Lu, N.: A non-monotone line search algorithm for unconstrained optimization. J. Sci. Comput. **42**(1), 38–53 (2010)
16. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Math. Program. **45**, 503–528 (1989)
17. Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Transform. Math. Softw. **7**(1), 17–41 (1981)
18. Nocedal, J., Wright, J.S.: Numerical Optimization, 2nd edn. Springer, New York (2006)
19. Nocedal, J.: Updating quasi-Newton matrices with limited storage. Math. Comput. **35**, 773–782 (1980)

20. Panier, E.R., Tits, A.L.: Avoiding the Maratos effect by means of a nonmonotone linesearch. SIAM J. Numer. Anal. **28**, 1183–1195 (1991)
21. Shi, Z.J., Shen, J.: Convergence of nonmonotone line search method. J. Comput. Appl. Math. **193**, 397–412 (2006)
22. Shi, Z.J., Wang, S.: Modified nonmonotone Armijo line search for descent method. Numer. Algoritm. **57**(1), 1–25 (2011)
23. Toint, Ph.L.: An assessment of nonmonotone linesearch technique for unconstrained optimization. SIAM J. Sci. Comput. **17**, 725–739 (1996)
24. Zhang, H.C., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. **14**(4), 1043–1056 (2004)