Numer Algor (2012) 59:523–540
DOI 10.1007/s11075-011-9502-5

ORIGINAL PAPER

An efficient nonmonotone trust-region method for unconstrained optimization

Masoud Ahookhosh · Keyvan Amini

Received: 8 September 2010 / Accepted: 24 August 2011 /
Published online: 6 September 2011

Abstract The monotone trust-region methods are well-known techniques for solving unconstrained optimization problems. While it is known that the nonmonotone strategies not only can improve the likelihood of finding the global optimum but also can improve the numerical performance of approaches, the traditional nonmonotone strategy contains some disadvantages. In order to overcome to these drawbacks, we introduce a variant nonmonotone strategy and incorporate it into trust-region framework to construct more reliable approach. The new nonmonotone strategy is a convex combination of the maximum of function value of some prior successful iterates and the current function value. It is proved that the proposed algorithm possesses global convergence to first-order and second-order stationary points under some classical assumptions. Preliminary numerical experiments indicate that the new approach is considerably promising for solving unconstrained optimization problems.

Keywords Unconstrained optimization · Trust-region methods · Nonmonotone technique · Global convergence

1 Introduction

Consider the following unconstrained optimization problem

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

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

M. Ahookhosh
e-mail: ahoo.math@gmail.com

where $f : \mathbf{R}^n \to \mathbf{R}$ is a twice continuously differentiable function. Many iterative procedures have been proposed for solving the problem (1), but most of them can be divided to two classes, namely line search methods and trust-region methods (see [13, 14]). Idea of line search algorithms is arising from finding a steplength in a specific direction, but the trust-region methods try to find a neighborhood around the current step x_k in which a quadratic model agrees with objective function. It is famous that trust-region methods have both theoretical convergence properties and the noticeable reputation (see [14, 16–18]). In comparison with quasi-Newton methods, trust-region methods converge to a point which not only is a stationary point but also satisfies in the necessary condition. In trust-region methods, at each iterate the trial step d_k is obtained by solving the following quadratic subproblem

$$\min m_k(d) = f_k + g_k^T d + \frac{1}{2} d^T B_k d, \quad \text{subject to } d \in \mathbf{R}^n \text{ and } \|d\| \leq \delta_k, \tag{2}$$

where $\|.\|$ is the Euclidean norm, $f_k = f(x_k)$, $g_k = \nabla f(x_k)$, B_k is the exact Hessian $G_k = \nabla^2 f(x_k)$ or its symmetric approximation, and δ_k is a trust-region radius.

A crucial point of trust-region methods at each iterate is a strategy for choosing radius δ_k. In the traditional trust-region method, the radius δ_k is determined based on a comparison between the model and the objective function. This leads the traditional trust-region method to define the following ratio

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{m_k(0) - m_k(d_k)}, \tag{3}$$

where the numerator is called the actual reduction and the denominator is called the predicted reduction. It is clear that there is an appropriate agreement between the model and the objective function over the current region whenever ρ_k be close to 1 , so it is safe to expand the trust-region radius δ_k in the next iterate. On the other hand, if ρ_k be a so small positive number or a negative number, the agreement is not appropriate and so the trust-region δ_k should be shrunk.

In 1982, Chamberlain et al. in [3] proposed the watchdog technique for constrained optimization, in which the standard line search condition is relaxed to overcome the Maratos effect. Inspired by this idea, Grippo, Lamparillo and Lucidi introduced a nonmonotone line search technique for Newton's method in [9]. They also proposed a truncated Newton method with nonmonotone line search for unconstrained optimization in [10]. In their proposal, the steplength α_k is accepted whenever

$$f(x_k + \alpha_k d_k) \leq f_{l(k)} + \beta \alpha_k \nabla f(x_k)^T d_k, \tag{4}$$

in which $\beta \in (0, 1)$ and

$$f_{l(k)} = \max_{0 \leq j \leq m(k)} \{f_{k-j}\}, \quad \forall k \in \mathbf{N} \cup \{0\}, \tag{5}$$

where $m(0) = 0$, $0 \le m(k) \le \min\{m(k-1)+1, N\}$ and $N \ge 0$. Their conclusions were overall approachable for the nonmonotone method, especially when applied to highly nonlinear problems and in presence of narrow curved valley. Nonmonotone methods are distinguished by the fact that they do not enforce strict monotonicity of the objective function values at successive iterates. It has been proved that using nonmonotone techniques can improve both the possibility of finding a global optimum and the rate of convergence of algorithm (see [9, 23]). Due to highly efficient behavior of nonmonotone techniques, many authors have been fascinated to work on employing nonmonotone strategies in various branches of optimization procedures.

The first exploitation of nonmonotone strategies in a trust-region framework was proposed by Deng et al. in [6] by changing the ratio (3) assessing an agreement between the quadratic model and the objective function over a trust region area. This idea was developed further by Zhou and Xiao in [21, 24], Xiao and Chu in [22], Toint in [19, 20], Dai in [5], Panier and Tits in [15] and so on. Anyway, the most common nonmonotone ratio is as follows

$$\tilde{\rho}_k = \frac{f_{l(k)} - f(x_k + d_k)}{m_k(0) - m_k(d_k)}, \tag{6}$$

where $f_{l(k)}$ is defined by (5). At the glance in (6), one can observe that the new point is compared with the worst point in previous iterates which means that this ratio is more relaxed in comparison with (3). Numerical experiments have been suggested that nonmonotone trust-region methods are more efficient than the monotone versions, especially in presence of narrow curved valley.

It is frequently discussed that in spite of many advantages of the traditional nonmonotone technique (6), it contains some drawbacks (see [4, 17, 23]), that some of them can be listed as follows:

- Although an iterative method is generating R-linearly convergent iterates for strongly convex functions, the iterates may not satisfy the condition (4) for k sufficiently large, for any fixed bound N on the memory.
- A good function value generated in any iterate is essentially discard due to the max in (4).
- In some cases, the numerical performances are very dependent on the choice of parameter N.

This paper introduces a modified nonmonotone strategy and employs it in a trust-region framework. The new algorithm solves the subproblem (2) in order to compute the trial step d_k , then set $x_{k+1} = x_k + d_k$ whenever the trial step d_k is accepted by the proposed nonmonotone trust-region approach. The analysis of the new algorithm shows that it inherits both stability of trust-region methods and effectiveness of the nonmonotone strategy. In addition, we investigate the global convergence to first-order and second-order stationary points of the proposed algorithm and establish the superlinear and the quadratic convergence properties. To illustrate the efficiency and robustness of the proposed algorithm in practice, we report some numerical experiments.

The rest of this paper organized as follows. In Section 2, we describe a new nonmonotone trust-region algorithm and show some of its properties. In Section 3, we prove that the proposed algorithm is globally convergent. Preliminarily numerical results are reported in Section 4. Finally, some conclusions are expressed in Section 5.

2 Algorithmic framework

This section devotes to describe a new nonmonotone trust-region algorithm and some of its properties. Since it can be as a variant of the nonmonotone strategy of Grippo et al. in [9], we expect similar properties and significant similarities in their proof for trust-region procedure.

It is well-known that the best convergence results are obtained by stronger nonmonotone strategy whenever iterates are far from the optimum, and by weaker nonmonotone strategy whenever iterates are close to the optimum (see [23]). In addition, we believe that the traditional nonmonotone strategy (5) almost ignores useful properties of the current objective function value f_k, where use it just in calculation of the maximum term. As a result, it seems that close to the optimum the traditional nonmonotone technique doesn't show an appropriate behavior. On the other hand, the maximum is one of the most information factors in the recent successful iterates and we do not want to lost it. In order to overcome to this disadvantage and introduce a more relaxed nonmonotone strategy, we define

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

in which $\eta_{\min} \in [0, 1)$, $\eta_{\max} \in [\eta_{\min}, 1]$ and $\eta_{k-1} \in [\eta_{\min}, \eta_{\max}]$. Obviously ,one obtains a stronger nonmonotone strategy whenever η_k is close to 1, and obtains a weaker nonmonotone strategy whenever η_k is close to 0. Hence, by choosing an adaptive η_k, one can increase the affection of $f_{l(k)}$ far from the optimum and can reduce it in close the optimum.

In the basis of considered discussion, we now can outline new nonmonotone trust-region algorithm as Algorithm 1.

Obviously, if in Algorithm 1 one sets $N = 0$ or $\eta_k = 1$, it reduces to the traditional trust-region algorithm. In addition, note that in Algorithm 1 iterates for which $\hat{\rho}_k \geq \mu_1$ are called successful iterates, and iterates for which $\hat{\rho}_k \geq \mu_2$ are called very successful iterates.

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

- **(H1)** The objective function f is continuously differentiable and has a lower bound on the level set $L(x_0) = \{x \in R^n | f(x) \leq f(x_0)\}$.
- **(H2)** The matrix B_k is an uniformly bounded matrix, i.e. there exists $M > 0$ such that $\|B_k\| \leq M$ for all $k \in \mathbf{N}$.

Algorithm 1 New Nonmonotone Trust-Region Algorithm (NMTR-N)

Step 0. *Initialization.* An initial point $x_0 \in \mathbf{R}^n$, symmetric matrix $B_0 \in \mathbf{R}^{n\times n}$ and initial trust-region radius $\delta_0 > 0$ are given. The constants $0 < \mu_1 \le \mu_2 < 1$, $0 < \gamma_1 \le \gamma_2 < 1$, $0 \le \eta_{\min} \le \eta_{\max} < 1$, $N \ge 0$ and $\epsilon > 0$ are also given. Set $k = 0$ and compute $f(x_0)$.

Step 1. Compute $g(x_k)$. If $\|g(x_k)\| \le \epsilon$, stop.

Step 2. Solve the subproblem (2) to determine a trial step d_k that $\|d_k\| \le \delta_k$.

Step 3. Compute $m(k)$, $f_{l(k)}$, and R_k and define

$$\hat{\rho}_k = \frac{R_k - f(x_k + d_k)}{m_k(0) - m_k(d_k)}. \tag{8}$$

If $\hat{\rho}_k \ge \mu_1$, then set $x_{k+1} = x_k + d_k$.

Step 4. Set

$$\delta_{k+1} \in \begin{cases} [\delta_k, \infty), & \text{if } \hat{\rho}_k \ge \mu_2; \\ [\gamma_2\delta_k, \delta_k), & \text{if } \mu_1 \le \hat{\rho}_k < \mu_2; \\ [\gamma_1\delta_k, \gamma_2\delta_k], & \text{if } \hat{\rho}_k < \mu_1. \end{cases} \tag{9}$$

Update B_{k+1} by a quasi-Newton formula, k=k+1 and go to Step 1.

Remark 1 If $f(x)$ is a twice continuously differentiable function and the level set $L(x_0)$ is bounded, then (H1) implies that $\|\nabla^2 f(x)\|$ is uniformly continuous and bounded on the open bounded convex set Ω which contains $L(x_0)$. Hence, there exists a constant $L > 0$ such that $\|\nabla^2 f(x)\| \le L$, and by using the mean value theorem we have

$$\|g(x) - g(y)\| \le L\|x - y\|, \quad \forall\, x, y \in \Omega.$$

Remark 2 Similar to [14], we can solve (2) inaccurately such that the decrease on the model m_k is at least as much as a fraction of that obtained by Cauchy point, i.e. there exists a constant $\beta \in (0, 1)$ such that, for all k,

$$m_k(0) - m_k(d_k) \ge \beta\|g_k\| \min\left\{\delta_k, \frac{\|g_k\|}{\|B_k\|}\right\}. \tag{10}$$

This condition have been called the sufficient reduction condition. Inequality (10) implies that $d_k \ne 0$ whenever $g_k \ne 0$.

We recall [4] as follows which we will need in later.

Lemma 3 *Suppose that the sequence* $\{x_k\}$ *be generated by Algorithm* 1*, then we have*

$$|f(x_k) - f(x_k + d_k) - (m_k(0) - m_k(d_k))| \le O\left(\|d_k\|^2\right).$$

Lemma 4 *Suppose that the sequence* $\{x_k\}$ *be generated by Algorithm* 1*, then the sequence* $\{f_{l(k)}\}$ *is a decreasing sequence.*

Proof Using definition of R_k and $f_{l(k)}$, we observe that

$$R_k = \eta_k f_{l(k)} + (1-\eta_k) f_k \le \eta_k f_{l(k)} + (1-\eta_k) f_{l(k)} = f_{l(k)}. \tag{11}$$

Assume that x_{k+1} is accepted by Algorithm 1. This fact along with (11) imply that

$$\frac{f_{l(k)} - f(x_k + d_k)}{m_k(0) - m_k(d_k)} \ge \frac{R_k - f(x_k + d_k)}{m_k(0) - m_k(d_k)} \ge \mu_1.$$

This follows

$$f_{l(k)} - f(x_k + d_k) \ge \mu_1(m_k(0) - m_k(d_k)) \ge 0, \quad \forall k \in \mathbf{N}.$$

Therefore, we have

$$f_{l(k)} \ge f_{k+1}, \quad \forall k \in \mathbf{N}. \tag{12}$$

Now, if $k \ge N$, from (12) taking into account that $m(k+1) \le m(k) + 1$, we obtain

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

For $k < N$, it is clear that $m(k) = k$. Since, for any k, $f_k \le f_0$, we see that $f_{l(k)} = f_0$.

Therefore, in both cases, the sequence $\{f_{l(k)}\}$ is a decreasing sequence. This completes the proof. □

Lemma 5 *Suppose that the sequence* $\{x_k\}$ *be generated by Algorithm* 1*, then we have*

$$f_{k+1} \le R_{k+1}, \quad \forall k \in \mathbf{N}. \tag{13}$$

Furthermore, if (H1) *holds, then the sequence* $\{x_k\}$ *is contained in* $L(x_0)$*.*

Proof From the definition of $f_{l(k+1)}$, we have $f_{k+1} \le f_{l(k+1)}$, for any $k \in \mathbf{N}$. Hence, (13) holds by the following inequality

$$\begin{aligned} f_{k+1} &= \eta_{k+1} f_{k+1} + (1-\eta_{k+1}) f_{k+1} \\ &\le \eta_{k+1} f_{l(k+1)} + (1-\eta_{k+1}) f_{k+1} = R_{k+1}, \quad \forall k \in \mathbf{N}. \end{aligned}$$

Obviously, the definition of R_k indicates that $R_0 = f_0$. By induction, assuming $x_i \in L(x_0)$ for all $i = 1, 2, \cdots, k$, we prove $x_{k+1} \in L(x_0)$. From (5), (7) and Lemma 4, we obtain

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

This means that the sequence $\{x_k\}$ is contained in $L(x_0)$ and the proof is completed. □

Corollary 6 *Suppose that* (H1) *holds and the sequence* $\{x_k\}$ *be generated by Algorithm* 1*. Then the sequence* $\{f_{l(k)}\}$ *is convergent.*

Proof Lemma 4 follows that the sequence $\{f_{l(k)}\}$ is non-increasing sequence. This fact together with (H1) imply that

$$\exists\, \lambda \; s.t. \; \forall n \in \mathbf{N} \;:\; \lambda \le f_{k+n} \le f_{l(k+n)} \le \cdots \le f_{l(k+1)} \le f_{l(k)}.$$

This inequality and Lemma 5 indicate the convergence of the sequence $\{f_{l(k)}\}$. □

3 Convergence analysis

We now wish to prove that Algorithm 1 is globally convergent to first-order and second-order stationary points. More precisely, we intend to prove that any limit point x^* of the sequence $\{x_k\}$ generated by Algorithm 1 satisfying $g(x^*) = 0$. We first discuss some convergence properties of the proposed algorithm then prove its global convergence in the sequel. Moreover, we show that the proposed algorithm is convergent superlinearly and quadratically under some suitable conditions.

In order to attain the global convergence, we require that $\|d_k\| \le c\|g_k\|$. Hence, we need to make an additional assumption as follows:

(H3) There exists a constant $c > 0$ such that the trial step d_k satisfies $\|d_k\| \le c\|g_k\|$.

Lemma 7 *Suppose that* (H1)–(H3) *hold and the sequence* $\{x_k\}$ *be generated by Algorithm* 1*, then we have*

$$\lim_{k\to\infty} f(x_{l(k)}) = \lim_{k\to\infty} f(x_k). \tag{14}$$

Proof It follows from the definition of x_{k+1} and (11) that

$$\frac{f_{l(k)} - f(x_k + d_k)}{m_k(0) - m_k(d_k)} \ge \frac{R_k - f(x_k + d_k)}{m_k(0) - m_k(d_k)} \ge \mu_1.$$

Thus,

$$f_{l(k)} - f(x_k + d_k) \ge \mu_1(m_k(0) - m_k(d_k)). \tag{15}$$

By substituting the index k with $l(k) - 1$, we get

$$f_{l(l(k)-1)} - f_{l(k)} \ge \mu_1 \left[m_k(x_{l(k)-1}) - m_k(x_{l(k)})\right] \ge 0.$$

Using of Corollary 6, it is easy to see that

$$\lim_{k\to\infty} \left[m_k(x_{l(k)-1}) - m_k(x_{l(k)})\right] = 0. \tag{16}$$

On the other hand, according to (H2), (H3) and (10) we obtain

$$\begin{aligned} m_k(x_{l(k)-1}) - m_k(x_{l(k)}) &\geq \beta \, \|g_{l(k)-1}\| \, \min\left\{\delta_{l(k)-1}, \frac{\|g_{l(k)-1}\|}{\|B_{l(k)-1}\|}\right\} \\ &\geq \beta \, \|g_{l(k)-1}\| \, \min\left\{\|d_{l(k)-1}\|, \frac{\|d_{l(k)-1}\|}{cM}\right\} \\ &\geq \frac{\beta}{c} \, \min\left\{1, \frac{1}{cM}\right\} \|d_{l(k)-1}\|^2 = \kappa \, \|d_{l(k)-1}\|^2 \geq 0, \end{aligned}$$

where $\kappa = \frac{\beta}{c} \min\left\{1, \frac{1}{cM}\right\}$. Thus, from (16), we have

$$\lim_{k\to\infty} \|d_{l(k)-1}\| = 0. \tag{17}$$

Uniform continuity of $f(x)$ along with (17) conclude that

$$\lim_{k\to\infty} f(x_{l(k)}) = \lim_{k\to\infty} f(x_{l(k)-1}). \tag{18}$$

Similar to literature [9], we define $\hat{l}(k) = l(k + N + 2)$. By induction, for all $j \geq 1$, we show

$$\lim_{k\to\infty} \|d_{\hat{l}(k)-j}\| = 0. \tag{19}$$

Note that for $j = 1$, since $\{\hat{l}(k)\} \subset \{l(k)\}$, (19) follows from (17). We now assume that j is given and (19) holds for j, so it is sufficient to show that (19) holds for $j + 1$. Let k be as large as so that $\hat{l}(k) - (j + 1) > 0$. Using (14) and substituting k with $\hat{l}(k) - j - 1$, we have

$$f(x_{\hat{l}(k)-j-1}) - f(x_{\hat{l}(k)-j}) \geq \mu \, \left[m_k(x_{\hat{l}(k)-j-1}) - m_k(x_{\hat{l}(k)-j})\right].$$

Following the same arguments to derive (17), we deduce

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

This means that the inductive is completed, therefore (19) holds for any given $j \geq 1$. Similar to (18), for any given $j \geq 1$, we have that $\lim_{k\to\infty} f(x_{\hat{l}(k)-j}) = \lim_{k\to\infty} f(x_{l(k)})$.

On the other hand, for any k, we know that

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

From (19), bearing the fact $\hat{l}(k) - j - 1 \leq N + 1$ in mind, we have

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

Therefore, from the uniform continuity of $f(x)$, we observe

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

Hence the proof is complete. □

Corollary 8 *Suppose* (H1)–(H3) *hold and the sequence* $\{x_k\}$ *be generated by Algorithm* 1*, then we have*

$$\lim_{k\to\infty} R_k = \lim_{k\to\infty} f(x_k). \tag{20}$$

Proof From (11) and (13), we have

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

This fact, along with Lemma 7, leads us to have the conclusion. □

Lemma 9 *Suppose that* (H1) *and* (H2) *hold, the sequence* $\{x_k\}$ *be generated by Algorithm* 1 *and* $\|g_k\| \ge \epsilon > 0$*. Then, for any* k*, there exists a nonnegative integer* p *such that* x_{k+p+1} *is a very successful iterate.*

Proof We assume that there exists an integer constant k such that for any arbitrary p the point x_{k+p+1} is not a very successful point. Hence, for any constant $p = 0, 1, 2, \cdots$, we have that $\hat{\rho}_{k+p} < \mu_2$. Using Step 4 of Algorithm 1, we get

$$\lim_{p\to\infty} \delta_{k+p} = 0. \tag{21}$$

From $\|g_k\| \ge \epsilon > 0$, (H2) and Remark 2 we obtain

$$m_k(0) - m_k(d_k) \ge \beta\|g_k\| \min\left\{\delta_k, \frac{\|g_k\|}{\|B_k\|}\right\} \ge \beta\epsilon \min\left\{\delta_k, \frac{\epsilon}{M}\right\}. \tag{22}$$

Thus, from Lemma 3, (21), and (22), we have

$$\begin{aligned}
|\rho_{k+p} - 1| &= \left|\frac{f(x_{k+p}) - f(x_{k+p} + d_{k+p})}{m_{k+p}(0) - m_{k+p}(d_{k+p})} - 1\right| \\
&= \left|\frac{f(x_{k+p}) - f(x_{k+p} + d_{k+p}) - (m_{k+p}(0) - m_{k+p}(d_{k+p}))}{m_{k+p}(0) - m_{k+p}(d_{k+p})}\right| \\
&\le \frac{O\left(\|d_{k+p}\|^2\right)}{\beta\epsilon \min\left\{\delta_{k+p}, \frac{\epsilon}{M}\right\}} \le \frac{O\left(\delta_{k+p}^2\right)}{\beta\epsilon \min\left\{\delta_{k+p}, \frac{\epsilon}{M}\right\}} \to 0 \quad as \quad p \to \infty.
\end{aligned}$$

This implies that for p sufficiently large, (3) holds. Hence, Lemma 5 follows that

$$\frac{R_{k+p} - f(x_{k+p} + d_{k+p})}{m_{k+p}(0) - m_{k+p}(d_{k+p})} \ge \frac{f(x_{k+p}) - f(x_{k+p} + d_{k+p})}{m_{k+p}(0) - m_{k+p}(d_{k+p})} \ge \mu_2.$$

Therefore, when p is sufficiently large, $\hat{\rho}_{k+p} \ge \mu_2$. This contradicts with the assumption $\hat{\rho}_{k+p} < \mu_2$. Hence,the proof is completed. □

Lemma 9 indicates that if the current iterate is not a first-order stationary point, then at least one can find a very successful iterate point, i.e. the trust-region radius δ_k can grow up. Now, we are in position to establish the global convergence of the new algorithm.

Theorem 10 *Suppose that* (H1) *and* (H2) *hold, and the sequence* $\{x_k\}$ *be generated by Algorithm* 1. *Then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{23}$$

Proof The proof is divided to the two following cases:

Case 1 If Algorithm 1 has finitely very successful iterates, then, for sufficiently large k, the iterate is unsuccessful. Suppose that k_0 is the index of the last successful iterate. If $\|g_{k_0+1}\| > 0$, then it will follow from Lemma 3.3 that there is a very successful iterate of index larger than k_0. This is a contradiction to the assumption.

Case 2 If Algorithm 1 has infinitely very successful iterates. By contradiction, we assume that there exist constants $\epsilon > 0$ and $K > 0$ such that

$$\|g_k\| \geq \epsilon \quad \forall k \geq K. \tag{24}$$

If x_{k+1} is a successful iterate and $k \geq K$, then, from (H2), Lemma 3 and (24), we have

$$\begin{aligned} R_k - f(x_k + d_k) &\geq \mu_1(m_k(0) - m_k(d_k)) \\ &\geq \beta\|g_k\| \ \min\left\{\delta_k, \frac{\|g_k\|}{\|B_k\|}\right\} \geq \beta\epsilon \ \min\left\{\delta_k, \frac{\epsilon}{M}\right\}. \end{aligned} \tag{25}$$

This inequality together with Corollary 8 imply that

$$\lim_{k\to\infty} \delta_k = 0. \tag{26}$$

In addition, Algorithm 1 has infinitely very successful iterates, so from Lemma 9 and (24) we have that the sequence $\{x_k\}$ contains infinitely very successful iterates in which trust-region radius is grown up. This fact contradicts with (26). Therefore, the assumption (24) is false, which yields (23). □

Theorem 11 *Suppose that* (H1) *and* (H2) *hold, then the sequence* $\{x_k\}$ *generated by Algorithm 1 satisfies*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{27}$$

Proof By contradiction, we assume that $\lim_{k\to\infty} \|g_k\| \neq 0$. Hence, there exist $\epsilon > 0$ and an infinite subsequence of $\{x_k\}$, indexed by $\{t_i\}$, such that

$$\|g_{t_i}\| \geq 2\epsilon > 0 \quad \forall i \in \mathbf{N}. \tag{28}$$

Theorem 10 ensures the existence, for each t_i, a first successful iterate $r(t_i) > t_i$ such that $\|g_{r(t_i)}\| < \epsilon$. We denote $r_i = r(t_i)$. Thus, there exists another subsequence, indexed by $\{r_i\}$, such that

$$\|g_k\| \geq \epsilon \text{ for } t_i \leq k < r_i \text{ and } \|g_{r_i}\| < \epsilon. \tag{29}$$

We now restrict our attention to the sequence of successful iterates whose indices are in the set $\kappa = \{k \in \mathbf{N} \mid t_i \leq k < r_i\}$. Using (29), for every $k \in \kappa$, we have that (25) holds. This fact along with Corollary 8 imply that

$$\lim_{k \to \infty} \delta_k = 0, \quad k \in \kappa. \tag{30}$$

Now, using (H2), Remark 2 and $\|g_k\| \geq \epsilon$, we deduce that (22) holds for $k \in \kappa$. So, using Lemma 3 and (30), it follows that

$$\begin{aligned} |\rho_k - 1| &= \left| \frac{f(x_k) - f(x_k + d_k)}{m_k(0) - m_k(d_k)} - 1 \right| \\ &= \left| \frac{f(x_k) - f(x_k + d_k) - (m_k(0) - m_k(d_k))}{m_k(0) - m_k(d_k)} \right| \\ &\leq \frac{O\left(\|d_k\|^2\right)}{\beta\epsilon \; \min\left\{\delta_k, \frac{\epsilon}{M}\right\}} \leq \frac{O\left(\delta_k^2\right)}{\beta\epsilon\delta_k} \to 0, \quad \text{as } k \to \infty \text{ and } k \in \kappa. \end{aligned}$$

If x_{k+1} is a successful iterate and $k+1 \in \kappa$, then, using (H2), Lemma 3 and $\|g_k\| \geq \epsilon$, we obtain

$$\begin{aligned} f_k - f(x_k + d_k) &\geq \mu_1(m_k(0) - m_k(d_k)) \\ &\geq \beta\|g_k\| \min\left\{\delta_k, \frac{\|g_k\|}{\|B_k\|}\right\} \geq \beta\epsilon \; \min\left\{\delta_k, \frac{\epsilon}{M}\right\}. \end{aligned} \tag{31}$$

As a consequence of the last inequality, the second term dominates in the minimum of (25). Hence, for $k \in \kappa$ sufficiently large, we get

$$\delta_k \leq \frac{1}{\beta\mu_1}(f_k - f_{k+1}). \tag{32}$$

Based on this bound and Lemma 5, we can write

$$\|x_{t_i} - x_{r_i}\| \leq \sum_{j \in \kappa,\, j = t_i}^{r_i - 1} \|x_j - x_{j+1}\| \leq \sum_{j \in \kappa,\, j = t_i}^{r_i - 1} \delta_j \leq \frac{1}{\beta\mu_1}(f_{t_i} - f_{r_i}) \leq \frac{1}{\beta\mu_1}(R_{t_i} - f_{r_i}) \tag{33}$$

for sufficiently large i. Corollary 8 suggests that the right hand side of the above inequality must converge to zero. Therefore, we obtain that $\|x_{t_i} - x_{r_i}\|$ must tend to zero, when i tends to infinity. From the continuity of gradient, we can deduce

$$\lim_{i \to \infty} \|g_{t_i} - g_{r_i}\| = 0. \tag{34}$$

By the definitions of $\{t_i\}$ and $\{r_i\}$, this is impossible which implies that $\|g_{t_i} - g_{r_i}\| \geq \epsilon$. Therefore, there is not any subsequence satisfying (28). This completes the proof. □

Theorem 12 *Suppose that all assumptions of Theorem* 11 *hold, then there is no limit point of the sequence* $\{x_k\}$ *being a local maximum of* $f(x)$.

Proof By the same way as in the proof of Theorem in [9], we have the result. Hence we omit the details. □

To establish the global convergence to second-order stationary points, similar to [6], we require to make an additional assumption as follows:

(H4) If $\lambda_{\min}(B_k)$ represents the smallest eigenvalue of the symmetric matrix B_k, then there exists a positive scaler c_3 such that

$$m_k(0) - m_k(d_k) \geq c_3 \lambda_{\min}(B_k)\delta^2.$$

Theorem 13 *Suppose that $f(x)$ be a twice continuously differentiable function and the assumptions* (H1)–(H4) *hold. Then there exists a limit point x^* of the sequence $\{x_k\}$ such that $\nabla^2 f(x^*)$ is positive semidefinite.*

Proof The conclusion can be proved by a similar way to Theorem 3.4 in [6]. The details are omitted. □

In the rest of this section, we establish both the superlinear and the quadratic convergence rate of the proposed algorithm under some reasonable conditions.

Theorem 14 *Suppose that the assumptions* (H1)–(H3) *hold. Also assume that the sequence $\{x_k\}$ be generated by Algorithm* 1 *converges to x^*, $d_k = -B_k^{-1} g_k$, $G(x) = \nabla^2 f(x)$ be a continuous matrix in a neighborhood $N(x^*, \epsilon)$ of x^*, and B_k satisfies in the following condition*

$$\lim_{k\to\infty} \frac{\|[B_k - \nabla^2 f(x^*)]d_k\|}{\|d_k\|} = 0.$$

Then the sequence $\{x_k\}$ converges to x^* superlinearly.

Proof By the same way as in the proof of Theorem 4.1 in [1], we have the conclusion. □

Theorem 15 *Assume that all conditions of Theorem* 14 *hold. Furthermore, suppose that $B_k = G(x_k)$ and $G(x)$ is a Lipschitz continuous matrix, then the sequence $\{x_k\}$ converges to x^* quadratically.*

Proof In order to observe a proof, see Theorem 4.2 in [1]. □

4 Preliminary numerical experiments

Here, we report some numerical results to show the performance of the new algorithm. We compare the proposed algorithm with the traditional

Table 1 Numerical results

Problem name	Dimension	NMTR-T	NMTR-M	NMTR-N1	NMTR-N2
Powell badly scal.	2	54/62	54/62	54/62	54/62
Brown badly scal.	2	39/39	41/42	41/42	47/48
Full Hessian FH1	2	30/32	30/32	19/21	29/31
Full Hessian FH2	2	6/7	6/7	5/6	6/7
Beale	2	14/16	14/16	12/14	14/16
Ext. Hiebert	2	1906/2061	1478/1549	1622/1681	1962/2039
MCCORMCK	2	10/11	10/11	10/11	10/11
Helical valley	3	41/46	27/31	29/32	30/34
Box three-dim.	3	28/30	27/29	26/28	27/29
Gulf res. and dev.	3	50/60	64/78	58/72	36/41
Gaussian	3	5/6	5/6	5/6	5/6
Brown and Dennis	4	36/42	27/34	30/37	29/35
Wood	4	80/86	69/77	65/75	47/52
Biggs EXP6	6	160/180	139/151	163/191	266/322
GENHUMPS	20	1021/1305	309/458	174/254	95/173
SINQUAD	20	97/118	81/96	115/145	387/440
FLETCBV3	20	6628/8120	2811/3359	2189/2589	2610/3063
Watson	31	77/86	73/80	43/49	67/77
Gen. tridiag. 2	40	188/264	181/263	188/267	143/228
Diag. 3	40	128/161	132/182	121/157	126/180
Penalty function II	100	333/408	231/291	99/136	135/188
Diag. 1	100	261/343	258/356	251/340	131/198
Trigonometric	500	78/88	100/110	70/79	100/110
Gen. Rosenbrock	500	4659/6844	5212/7654	5112/7521	6072/8048
Broyden tridiag.	500	2418/3404	1077/1738	2408/3440	1311/1862
Diag. 9	500	970/1382	561/866	248/381	699/913
CUBE	500	675/837	721/910	148/198	969/1341
NONSCOMP	500	2084/2652	2335/3035	916/1355	2938/3720
POWER	500	6982/10184	7983/11472	5398/7762	9833/13314
BDQRTIC	500	443/621	174/251	116/170	108/160
TRIDIA	500	3280/4886	3401/4811	2459/3563	3786/5187
FLETCHCR	500	4005/5759	4505/6431	4515/6555	5473/7272
Ext. tridiag. 2	1000	123/124	123/124	89/90	123/124
Gen. tridiag. 1	1000	914/1318	280/412	235/341	147/228
Ext. Powell sing.	1000	1765/2474	2059/3051	292/435	3010/4480
Ext. Rosenbrock	1000	423/576	444/654	404/604	714/1080
Partial per. quad.	1000	770/1104	235/351	225/331	123/190
Almost per. quad.	1000	1044/1546	1206/1983	581/924	1374/2081
Ext. block diag. 1	1000	12/13	12/13	12/13	13/15
per. quad. diag.	1000	411/509	186/227	127/155	106/144
Ext. quad. pen. QP2	1000	62/68	67/74	63/70	62/69
Ext. White and Holst	1000	856/1186	169/221	876/1322	1398/2116
Gen. White and Holst	1000	8964/12879	13428/19229	6018/8334	15807/22068
Per. tridiag. Quad.	1000	1082/1582	1206/1982	596/920	1372/1981
Variably dim.	1000	21/21	21/21	15/15	21/21
Gen. PSC1	1000	198/212	51/54	85/97	51/54
Ext. PSC1	1000	15/15	15/15	13/13	15/15
Quad. QF1	1000	1063/1582	1215/1995	579/912	1431/1982
Quad. QF2	1000	1386/2047	1462/2384	847/1427	1665/2477
Raydan 1	1000	958/1411	1101/1791	499/806	1216/1584
Diag. 2	1000	285/285	285/285	195/195	285/285
Diag. 4	1000	5/5	5/5	4/4	5/5
Per. quad.	1000	1070/1563	1197/1975	562/894	1350/2000
Ext. Beale	1000	16/17	16/17	13/14	16/17

Table 1 (continued)

Problem name	Dimension	NMTR-T	NMTR-M	NMTR-N1	NMTR-N2
Staircase 1	1000	395/478	239/335	106/150	296/404
Staircase 2	1000	366/484	213/293	161/223	318/419
Ext. Wood	1000	5310/6816	1217/1827	1222/1635	1529/2165
Ext. Maratos	1000	619/879	474/695	465/707	560/831
Gen. Quartic	1000	19/19	19/19	14/14	19/19
NONDQUAR	1000	671/678	612/618	620/631	606/610
DQDRTIC	1000	13/14	13/14	13/14	13/14
LIARWHD	1000	13/13	20/21	12/12	20/20
VARDIM	1000	21/21	21/21	15/15	21/21
QUARTC	1000	22/22	22/22	17/17	22/22
ENGVAL1	1000	559/762	242/329	150/207	120/168
EDENSCH	1000	558/768	254/365	202/285	126/186
EG2	1000	94/99	72/73	81/87	119/132
DIXON3DQ	1000	1357/1487	1322/1531	1288/1476	1636/1985
BIGGSB1	1000	1345/1476	1301/1490	1290/1476	1626/1872
HIMMELBG	1000	28/28	28/28	21/21	28/28
Hager	1000	741/1062	269/378	245/360	150/218
DIXMAANI	1200	2398/2415	2311/2320	2243/2246	2258/2259
DIXMAANJ	1200	1345/1345	1345/1345	1345/1345	1345/1345
DIXMAANK	1200	1162/1162	1162/1162	1162/1162	1162/1162
DIXMAANL	1200	956/958	1279/1280	1279/1280	1279/1280
Ext. tridiag. 1	2000	26/26	26/26	17/17	26/26
Ext. DENSCHNF	2000	22/24	18/19	19/21	15/16
Penalty function I	2000	29/29	29/29	29/29	29/29
Ext. Himmelblau	2000	17/17	13/15	13/13	14/16
ARWHEAD	2000	8/10	8/10	7/9	8/10
BDEXP	2000	26/26	26/26	26/26	26/26
SINCOS	2000	15/15	15/15	13/13	15/15
Ext. Cliff	2000	16/16	16/16	16/16	16/16
HIMMELH	2000	7/7	7/7	6/6	7/7
Ext. quad. pen. QP1	3000	19/19	19/19	16/16	19/19
Diag. 7	3000	7/7	7/7	6/6	7/7
DIXMAANA	3000	13/13	13/13	12/12	13/13
DIXMAANB	3000	43/51	28/34	67/77	32/36
DIXMAANC	3000	52/61	53/61	46/55	22/25
DIXMAAND	3000	50/58	54/64	48/58	32/37
DIXMAANE	3000	278/279	278/279	151/152	278/279
DIXMAANF	3000	225/228	224/226	145/148	197/199
DIXMAANG	3000	435/441	390/393	251/256	263/272
DIXMAANH	3000	220/236	265/270	350/565	543/559
Ext. DENSCHNB	5000	9/9	9/9	9/9	9/9
Full Hessian FH3	5000	5/5	5/5	5/5	5/5
Ext. quad. exp. EP1	5000	4/6	4/6	4/6	4/6
Ext. Fre. and Roth	5000	15/15	15/15	15/15	15/15
Ext. TET	5000	8/8	8/8	8/8	8/8
Raydan 2	5000	8/8	8/8	8/8	8/8
Diag. 5	5000	7/7	7/7	7/7	7/7
Diag. 8	5000	6/6	6/6	6/6	6/6
ARGLINB	5000	3/3	3/3	3/3	3/3
ARGLINC	5000	3/3	3/3	3/3	/3/3

nonmonotone trust-region algorithm, (NMTR-T), and the nonmonotone trust-region algorithm of Mo et al. in [11], (NMTR-M). We test all algorithms on 104 standard unconstrained test problems in [2] and [12]. The starting points are the standard ones provided by these literatures. We performed our codes in double precision arithmetic format in MATLAB 7.4 programming environment on a 3.0 GHz Intel single-core processor computer with 1GB of RAM. For proper comparison, we provide all codes in the same subroutine and solved the trust-region subproblems by Steihaug–Toint procedure (see p. 205 in [4]).

In all algorithms, we set $\mu_1 = 0.05$, $\mu_2 = 0.9$ and $\delta_0 = 10$. In addition, the stopping criterion is

$$\|\nabla f(x_k)\| \leq 10^{-6}\|\nabla f(x_0)\|.$$

We choose $N = 10$ for the new algorithm and the NMTR-T. Based on our preliminary numerical experiments, we decided to update η_k by

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

We name our algorithm as NMTR-N1 and NMTR-N2 whenever $\eta_0 = 0.85$ and $\eta_0 = 0.2$, respectively. For NMTR-M algorithm, we select $\eta_0 = 0.85$ as

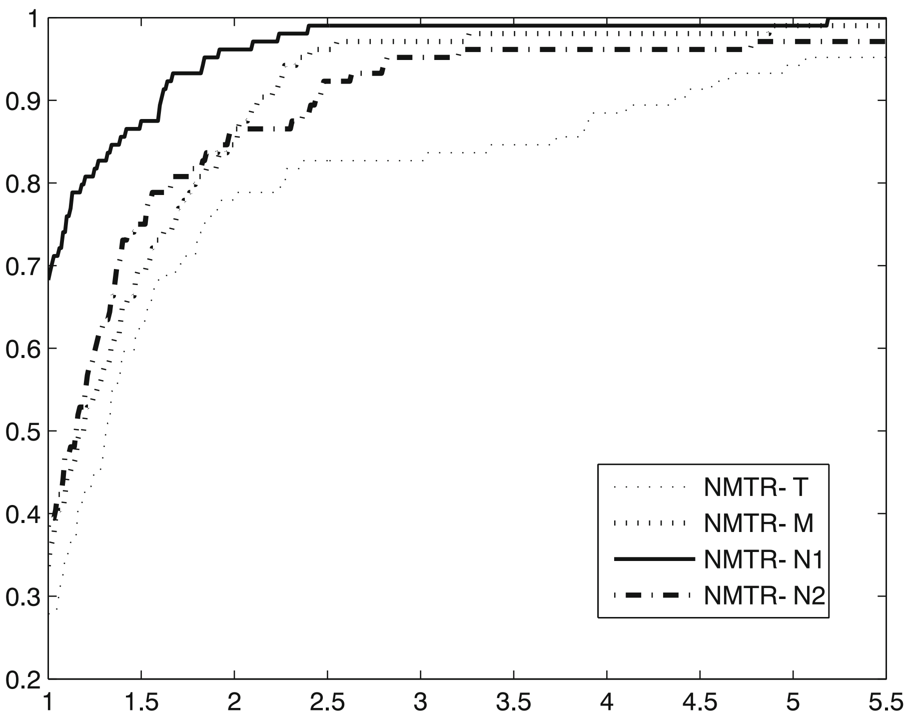

Fig. 1 Performance profile for the number of iterates

considered in [8, 11]. For all algorithms, the trust-region radius update is implemented using

$$\delta_{k+1} = \begin{cases} c_1\|d_k\|, & \text{if } \hat{r}_k < \mu_1; \\ \delta_k, & \text{if } \mu_1 \le \hat{r}_k < \mu_2; \\ \max[\delta_k, c_2\|d_k\|], & \text{if } \hat{r}_k \ge \mu_2, \end{cases}$$

where $c_1 = 0.25$ and $c_2 = 2.5$. Table 1 shows the total number of iterates (n_i) and the total number of function evaluations (n_f) that each algorithm need to solve an arbitrary problem. In Table 1, we have only kept the problems for which all algorithms converge to the same local solution.

In order to compare iterative algorithms, Dolan and Moré in [7] proposed a new technique employing a statistical process by demonstration of a performance profile. In this technique, one can choose a performance index as measure of comparison among the considered algorithms and can illustrate the results with a performance profile. Figures 1 and 2 show this process with index of the number of iterates and the number of function evaluations, respectively.

In Fig. 1, firstly, we observe that NMTR-N1 have the most wins while it solves about 70% of the tests with the greatest efficiency. Moreover, if we focus our attention on the ability of completing a run successfully, we have again that NMTR-N1 is the best among considered algorithms. Secondly,

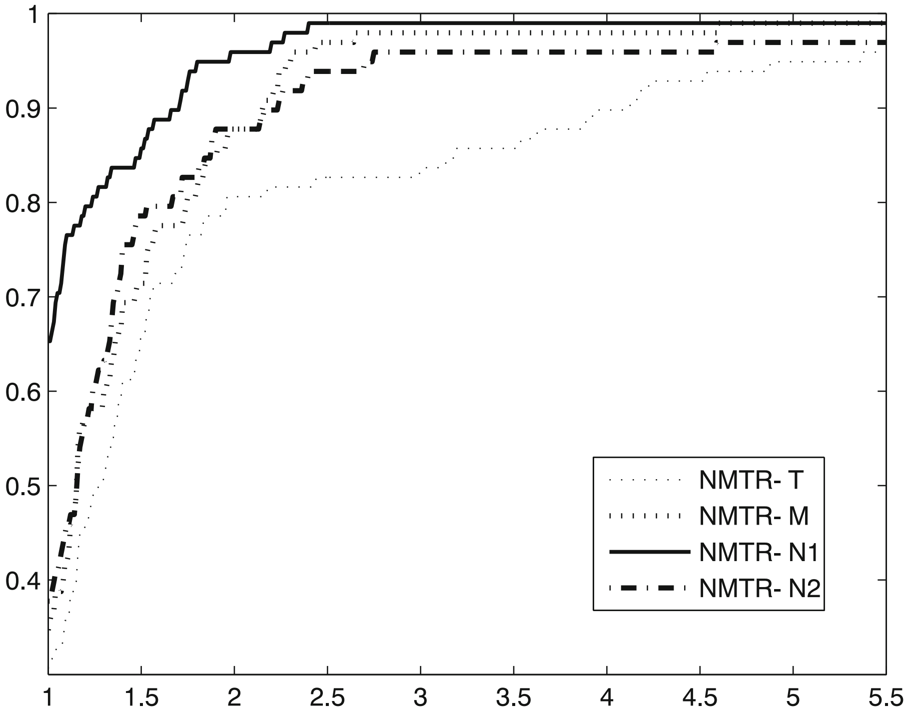

Fig. 2 Performance profile for the number of function evaluations

NMTR-N2 is competitive with NMTR-M, but in most cases it grows up much faster than NMTR-M. Finally, one can see that NMTR-N1 increase noticeably in comparison with the other considered algorithms. It means that in the cases that NMTR-N1 is not the best algorithm which performance index is close to the performance index of the best algorithm. Therefore, we can deduce that the new algorithm is more efficient and robustness than the other considered trust-region algorithms in the sense of the total number of iterates.

Results of Fig. 2 are remarkably similar to the mentioned results of Fig. 1, in the sense of the total number of function evaluations.

5 Conclusions

The present paper proposes a new nonmonotone strategy and exploits it in trust-region framework to introduce an efficient procedure to solve unconstrained optimization. The new nonmonotone strategy has been constructed based on appropriate using of the function value in current iterate to overcome some disadvantages of traditional nonmonotone strategy. In addition, an adaptive process for increasing the effects of traditional nonmonotone term far from the optimum and decreasing its effects close to the optimum is suggested. The global convergence to first-order and second-order stationary points for the new algorithm, similar to those stated for common trust-region algorithm, have been established. Preliminary numerical results show the significant efficiency of the new algorithm.

Acknowledgements The authors would like to thank the referees for their valuable suggestions and comments.

References

1. Ahookhosh, M., Amini, K.: A nonmonotone trust region method with adaptive radius for unconstrained optimization. Comput. Math. Appl. **60**, 411–422 (2010)
2. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**(1), 147–161 (2008)
3. 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)
4. Conn, A.R., Gould, N.I.M., Toint, Ph.L.: Trust-Region Methods. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2000)
5. Dai, Y.H.: On the nonmonotone line search. J. Optim. Theory Appl. **112**(2), 315–330 (2002)
6. Deng, N.Y., Xiao, Y., Zhou, F.J.: Nonmonotonic trust region algorithm. J. Optim. Theory Appl. **76**, 259–285 (1993)
7. Dolan, E., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
8. Gould, N.I.M, Orban, D., Sartenaer, A., Toint, Ph.L.: Sentesivity of trust-region algorithms to their parameters. Q. J. Oper. Res. **3**, 227–241 (2005)
9. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)

10. Grippo, L., Lampariello, F., Lucidi, S.: A truncated Newton method with nonmonotone linesearch for unconstrained optimization. J. Optim. Theory Appl. **60**, 401–419 (1989)
11. Mo, J., Liu, C., Yan, S.: A nonmonotone trust region method based on nonincreasing technique of weighted average of the succesive function value. J. Comput. Appl. Math. **209**, 97–108 (2007)
12. Moré, J.J., Garbow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**, 17–41 (1981)
13. Nocedal, J., Yuan, Y.: Combining trust region and line search techniques. In: Yuan, Y. (ed.) Advanced in Nonlinear Programming, pp. 153–175. Kluwer Academic, Dordrecht (1996)
14. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, New York (2006)
15. Panier, E.R., Tits, A.L.: Avoiding the Maratos effect by means of a nonmonotone linesearch. SIAM J. Numer. Anal. **28**, 1183–1195 (1991)
16. Powell, M.J.D.: Convergence properties of a class minimization algorithms. In: Mangasarian, O.L., Meyer, R.R., Robinson, S.M. (eds.) Nonlinear Programming, vol. 2, pp. l–27. Academic, New York (1975)
17. Powell, M.J.D.: On the global convergence of trust region algorithms for unconstrained optimization. Math. Program. **29**, 297–303 (1984)
18. Schultz, G.A., Schnabel, R.B., Byrd, R.H.: A family of trust-region-based algorithms for unconstrained minimization with strong global convergence. SIAM J. Numer. Anal. **22**, 47–67 (1985)
19. Toint, Ph.L.: An assessment of nonmonotone linesearch technique for unconstrained optimization. SIAM J. Sci. Comput. **17**, 725–739 (1996)
20. Toint, Ph.L.: Non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints. Math. Program. **77**, 69–94 (1997)
21. Xiao, Y., Zhou, F.J.: Nonmonotone trust region methods with curvilinear path in unconstrained optimization. Computing **48**, 303–317 (1992)
22. Xiao, Y., Chu, E.K.W.: Nonmonotone trust region methods. Technical Report 95/17, Monash University, Clayton, Australia (1995)
23. Zhang, H.C., Hager, W.W.: A nonmonotone line search technique for unconstrained optimization. SIAM J. Optim. **14**(4), 1043–1056 (2004)
24. Zhou, F., Xiao, Y.: A class of nonmonotone stabilization trust region methods. Computing **53**(2), 119–136 (1994)