Available online at www.sciencedirect.com

ScienceDirect

Computers and Mathematics with Applications 55 (2008) 2158–2172

An International Journal
computers & mathematics with applications

www.elsevier.com/locate/camwa

Incorporating nonmonotone strategies into the trust region method for unconstrained optimization[☆]

Neng-zhu Gu[a], Jiang-tao Mo[b,*]

[a] *School of Business, University of Shanghai for Science and Technology, Shanghai 200093, China*
[b] *College of Mathematics and Information Science, Guangxi University, Nanning 530004, China*

Received 7 July 2005; received in revised form 13 August 2007; accepted 30 August 2007

Abstract

This paper concerns a nonmonotone line search technique and its application to the trust region method for unconstrained optimization problems. In our line search technique, the current nonmonotone term is a convex combination of the previous nonmonotone term and the current objective function value, instead of an average of the successive objective function values that was introduced by Zhang and Hager [H. Zhang, W.W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM J. Optim. 14 (4) (2004) 1043–1056]. We incorporate this nonmonotone scheme into the traditional trust region method such that the new algorithm possesses nonmonotonicity. Unlike the traditional trust region method, our algorithm performs a nonmonotone line search to find a new iteration point if a trial step is not accepted, instead of resolving the subproblem. Under mild conditions, we prove that the algorithm is global and superlinear convergence holds. Primary numerical results are reported.
© 2008 Published by Elsevier Ltd

Keywords: Unconstrained optimization; Nonmonotone trust region method; Nonmonotone line search; Global convergence; Superlinear convergence

1. Introduction

We consider the following unconstrained optimization problem

$$\min\{f(x) | x \in \Re^n\}, \tag{1.1}$$

where $f : \Re^n \to \Re$ is twice continuously differentiable.

Since it was proposed by Levenberg [1] and Marquardt [2] for nonlinear least-square problems and then developed by Goldfeld [3] for unconstrained optimization, trust region methods have been studied extensively by many researchers. For example, Powell [4] established the convergence result of the trust region method for unconstrained

☆ This work was supported by the Science Foundation of Guangxi, No: 0542043.

* Corresponding author.
E-mail addresses: gnzemail@hotmail.com (N.-z. Gu), mjt@gxu.edu.cn (J.-t. Mo).

0898-1221/$ - see front matter © 2008 Published by Elsevier Ltd
doi:10.1016/j.camwa.2007.08.038

optimization, Fletcher [5], Yuan [6], Powell and Yuan [7] proposed various trust region algorithms for constrained optimization problems. For a more complete introduction, we refer the reader to [8].

Trust region methods solve an optimization problem iteratively. At each iteration, a trial step d_k is generated by solving the subproblem

$$\min_{d\in\Re^n} \quad g_k^T d + \frac{1}{2} d^T B_k d = \phi_k(d) \qquad (1.2)$$
$$\text{s.t.} \quad \|d\| \le \Delta_k,$$

where $g_k = \nabla f(x_k)$, $B_k \in \Re^{n\times n}$ is an approximate Hessian matrix of f at x_k, $\Delta_k > 0$ is a trust region radius. Some criteria are used to decide whether a trial step d_k is accepted. If a trial step is not accepted, traditional trust region methods resolve the subproblem (1.2) by reducing the trust region radius until an acceptable step is found. Therefore, the subproblem may be solved several times at an iteration before an acceptable step is found, and these repetitious processes are likely to increase the total cost of computation for large scale problems.

To improve the computational efficiency of trust region methods, new type trust region algorithms that combine line search techniques have been developed (e.g., Necedal and Yuan [9] and Gertz [10]). The prominent virtue of these algorithms is to perform a line search to find an successful iterative point when a trial step is not accepted; this mechanism implies that the subproblem is needs to be solved once for each iteration.

Most of the proposed methods for optimization problems require objective function values monotonically decreasing at each iteration. However, Grippo, Lampariello and Lucidi [11] shown that this scheme can considerably slow the rate of convergence in the intermediate stages of the minimization process, especially in the presence of the narrow curved valley. To overcome this shortcoming, they introduced a nonmonotone line search technique of the form

$$f(x_k + \alpha d_k) \le \max_{0\le j\le m(k)} f(x_{k-j}) + \delta\alpha\nabla f(x_k)^T d_k, \qquad (1.3)$$

where $\delta \in (0, 1)$ is a constant, N is an integer and

$$m(k) = \begin{cases} k, & k \le N; \\ \min\{m(k-1)+1, N\}, & k > N. \end{cases} \qquad (1.4)$$

This technique leads to a breakthrough in nonmonotonic algorithms for nonlinear optimization. For example, based on (1.3), Deng, Xiao and Zhou [12] first proposed a nonmonotone trust region method for unconstrained optimization, Ke and Han [13], Sun [14], Fu and Sun [15] presented various nonmonotone trust region methods.

Recently, Zhang and Hager [16] found that there still exist some drawbacks with the nonmonotone technique (1.3). First, a good function value generated in any iteration might be excluded due to the max in (1.3). Second, in many cases, the numerical performance is dependent on the choice of N (see [11,17]). Moreover, for any given bound N on the memory, even an iterative method is generating R-linearly convergence for a strongly convex function, the iterates may not satisfy the condition (1.3) for k sufficiently large (see [18]). To improve these limitations, Zhang and Hager [16] proposed a nonmonotone gradient method for unconstrained optimization. Their method requires an average of the successive function values that is decreasing – that is to say, the steplength α is computed to satisfy the following line search condition

$$f(x_k + \alpha d_k) \le C_k + \delta\alpha\nabla f(x_k)^T d_k, \qquad (1.5)$$

where

$$C_k = \begin{cases} f(x_k), & k = 1; \\ (\eta_{k-1} Q_{k-1} C_{k-1} + f(x_k))/Q_k, & k \ge 2, \end{cases} \qquad (1.6)$$

$$Q_k = \begin{cases} 1, & k = 1; \\ \eta_{k-1} Q_{k-1} + 1, & k \ge 2, \end{cases} \qquad (1.7)$$

and $\eta_{k-1} \in [\eta_{\min}, \eta_{\max}]$. $\eta_{\min} \in (0, \eta_{\max})$ and $\eta_{\max} \in (0, 1)$ are two chosen parameters. The numerical results reported in [16] show that the nonmonotone technique (1.5) is particularly efficient for unconstrained problems.

Inspired by this nonmonotone technique, Mo, Liu and Yan [19] introduced it into trust region method and developed a nonmonotone algorithm. The numerical results indicate that the algorithm is robust and encouraging.

As we see, the definition of mean values C_k implies that each C_k is a convex combination of the previous C_{k-1} and f_k, including the complex η_k and Q_k. In theory, the best convergence results were obtained by dynamically varying η_k, using values closer to 1 when the iterates were far from the optimum, and using values closer to 0 when the iterates were near the optimum (see [16] for details). In practice, however, it becomes an encumbrance to update η_k and Q_k at each iteration. Therefore, in this study, we introduce another nonmonotone line search

$$f(x_k + \alpha d_k) \le D_k + \delta\alpha \nabla f(x_k)^T d_k, \tag{1.8}$$

where D_k is a simple convex combination of the previous D_{k-1} and f_k, say

$$D_k = \begin{cases} f(x_k), & k = 1; \\ \eta D_{k-1} + (1-\eta) f(x_k) & k \ge 2 \end{cases} \tag{1.9}$$

for some fixed $\eta \in (0, 1)$, or a variable η_k. In the rest of this paper, our purpose is to develop an algorithm that combines the nonmonotone technique (1.8) and trust region method for unconstrained optimization problems. The algorithm doesn't restrict the object function values to be monotonically decreasing. Furthermore, if a trial step is not accepted, the algorithm performs the nonmonotone line search (1.8) to find an iterative point instead of resolving the subproblem. The new algorithm can viewed as a generalization of the Nocedal and Yuan algorithm [9] from monotone to non-monotone, or the Mo et al. algorithm [19] from non-linesearch to linesearch.

The paper is organized as follows. In Section 2, we describe the algorithm. In Section 3, we establish the global convergence and superlinear convergence for the algorithm. Primary numerical results are presented in Section 4. Finally, our conclusions are given in Section 5.

2. Algorithm

In this section, we state our method in the form of an algorithm. Throughout this paper, we use $\|\cdot\|$ to represent the Euclidean norm and denote $f(x_k)$ by f_k, $\nabla f(x_k)$ by g_k, etc. Vectors are column vectors unless a transpose is used.

The method we present is also an iterative method. At each iteration, a trial step d_k is generated by solving the trust region subproblem (1.2). Similarly to [9], we solve (1.2) inaccurately such that $\|d_k\| \le \Delta_k$,

$$\phi_k(0) - \phi_k(d_k) \ge \tau \|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\} \tag{2.1}$$

and

$$d_k^T g_k \le -\tau \|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\}, \tag{2.2}$$

where $\tau \in (0, 1)$ is a constant.

To determine whether a trial step will be accepted, we compute ρ_k, the ratio between the actual reduction and the predicted reduction. Instead of using $f_k - f(x_k + d_k)$ as the actual reduction, we choose $D_k - f(x_k + d_k)$ as the actual reduction, such that the algorithm does not require object function values to be monotonically decreasing. ρ_k is given as

$$\rho_k = \frac{D_k - f(x_k + d_k)}{\phi_k(0) - \phi_k(d_k)}, \tag{2.3}$$

where D_k is defined by (1.9). If $\rho_k \ge \mu$, where μ is a constant, we accept d_k as a successful step and let $x_{k+1} = x_k + d_k$. Otherwise, we generate a new iterative point by $x_{k+1} = x_k + \alpha_k d_k$, where α_k is a steplength satisfying the line search condition (1.8).

More precisely, we describe the nonmonotone trust region algorithm as follows:

Algorithm 1 (*Nonmonotone Trust Region Method With line Search*).

Step 1. Give $x_1 \in \Re^n$, $\Delta_1 > 0$. Choose constants $c_1, c_2, \mu, \epsilon, \lambda, \delta, \eta$, such that $0 < c_1 < 1 < c_2$, $\mu \in (0, 1)$, $\epsilon = 10^{-6}$, $\lambda \in (0, 1)$, $\delta \in (0, 1/2)$ and $\eta \in (0, 1)$; a symmetric positive definite matrix $B_1 \in \Re^{n\times n}$. Set $k := 1$.

Step 2. Compute g_k, if $\|g_k\| < \epsilon$, stop.
Step 3. Solve (1.2) inaccurately, such that $\|d_k\| \le \Delta_k$, (2.1) and (2.2) are satisfied.
Step 4. Compute D_k by (1.9) and ρ_k by (2.3).
Step 5. If $\rho_k \ge \mu$, go to Step 6. Otherwise, compute i_k, the minimum nonnegative integer i satisfying

$$f(x_k + \lambda^i d_k) \le D_k + \delta\lambda^i g_k^T d_k. \tag{2.4}$$

Set $\alpha_k = \lambda^{i_k}$,

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2.5}$$

and

$$\Delta_{k+1} \in [\|x_{k+1} - x_k\|, c_1\Delta_k], \tag{2.6}$$

go to Step 7.
Step 6. Set

$$x_{k+1} = x_k + d_k, \tag{2.7}$$

and

$$\Delta_{k+1} \begin{cases} = \Delta_k, & \text{if } \|d_k\| < \Delta_k, \\ \in [\Delta_k, c_2\Delta_k], & \text{if } \|d_k\| = \Delta_k. \end{cases} \tag{2.8}$$

Step 7. Update the symmetric matrix B_k, set $k := k + 1$, go to Step 2.

Remark 2.1. Applying the Algorithm 2.6 in [9], we can find an inaccurate solution of (1.2), so that $\|d_k\| \le \Delta_k$, (2.1) and (2.2) are satisfied.

3. Global convergence

We now turn to analyze the convergence behaviour of Algorithm 1 when it is applied to problem (1.1). To this end, the following assumptions are required.

Assumption A. The level set $L = \{x \mid f(x) \le f(x_1)\} \subset \Omega$, where Ω is a close, bound set of $\Re^n$.

Assumption B. There exists a positive constant m such that

$$d^T B_k d \ge m\|d\|^2, \quad \forall d \in \Re^n \text{ and } k = 1, 2, \ldots.$$

Remark 3.1. Assumption A together with f is twice continuously differentiable, suggesting that there exists a constant $M > m$, such that

$$\|\nabla^2 f(x)\| \le M, \quad \forall\, x \in \Omega. \tag{3.1}$$

For simplicity, we define two index sets as follows

$$I = \{k : \rho_k \ge \mu\} \quad \text{and} \quad J = \{k : \rho_k < \mu\}.$$

We first analyze the nonmonotone line search conditions (1.8). Obviously, if parameter η is chosen to be zero, this line search reduces to a monotone line search. The following lemma show that if $\eta \in (0, 1)$, $D_k \ge f_k$, this implies that the nonmonotonicity of (1.8) can be guaranteed.

Lemma 3.1. *Let $\{x_k\}$ be the sequence generated by Algorithm 1. Then*

$$f_{k+1} \le D_{k+1} \tag{3.2}$$

holds for all k.

Proof. Due to (1.9), the definition of D_k, we obtain $D_{k+1} - f_{k+1} = \eta(D_k - f_{k+1})$. Now, we consider two cases dependent on $k \in I$ and $k \in J$.

Case 1. $k \in I$, i.e., $\rho_k \geq \mu$. It follows from (2.1) and (2.3) that

$$D_k - f_{k+1} > \mu\tau\|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\} \geq 0.$$

Therefore we have $D_{k+1} - f_{k+1} \geq 0$.

Case 2. $k \in J$. In this case, the trial step cannot be accepted, and a line search is used in the algorithm. Note that (2.2) implies that $g_k^T d_k \leq 0$ for all k. This inequality together with (2.4) yield

$$D_k - f_{k+1} \geq -\delta\alpha_k g_k^T d_k \geq 0.$$

Therefore we also have $D_{k+1} - f_{k+1} \geq 0$ for $k \in J$. □

Next, we show that Algorithm 1 is well defined. It is enough to prove that there exists an integer i_k such that line search (2.4) holds.

Lemma 3.2. *For any $k \in J$, the line search in Step 5 of Algorithm 1 terminates in a finite number of steps, i.e., there exists an integer i_k such that (2.4) holds.*

Proof. Suppose first, for the purpose of deriving a contradiction, that there exists $k \in J$ such that

$$f(x_k + \lambda^i d_k) > D_k + \delta\lambda^i g_k^T d_k, \quad \forall i.$$

Using $D_k \geq f_k$, we obtain

$$\frac{f(x_k + \lambda^i d_k) - f_k}{\lambda^i} > \delta g_k^T d_k.$$

Recall that f is differentiable, and taking the limit with $i \to \infty$, we have

$$g_k^T d_k \geq \delta g_k^T d_k. \tag{3.3}$$

Since $\delta \in (0, 1/2)$, inequality (3.3) suggests that $g_k^T d_k \geq 0$. However, from Step 2 of Algorithm 1, we have $\|g_k\| \geq \epsilon$; thus (2.2) implies that $g_k^T d_k < 0$. Therefore, (3.3) contradicts (2.2). This indicate that for any $k \in J$, there exists an $i_k > 0$, such that (2.4) holds. □

Under suitable assumptions, we can establish a lower bound for stepsize α_k for $k \in J$.

Lemma 3.3. *Assume that Assumption B holds, a sequence $\{x_k\}$ is generated by Algorithm 1. Then stepsize α_k satisfies*

$$\alpha_k > \frac{(1-\delta)\lambda m}{M} \tag{3.4}$$

for all $k \in J$.

Proof. From Step 5 of Algorithm 1, we have

$$f(x_k + \lambda^{-1}\alpha_k d_k) > D_k + \delta\lambda^{-1}\alpha_k g_k^T d_k. \tag{3.5}$$

By Taylor's expansion, we obtain

$$f(x_k + \lambda^{-1}\alpha_k d_k) = f_k + \lambda^{-1}\alpha_k g_k^T d_k + \frac{1}{2}\lambda^{-2}\alpha_k^2 d_k^T \nabla^2 f(\xi_k) d_k, \tag{3.6}$$

where $\xi_k \in (x_k, x_k + \lambda^{-1}\alpha_k d_k)$. Using (3.2), (3.5), (3.6) and (3.1), we get

$$\delta\lambda^{-1}\alpha_k g_k^T d_k < \lambda^{-1}\alpha_k g_k^T d_k + \frac{1}{2}\lambda^{-2}\alpha_k^2 M\|d_k\|^2.$$

This inequality leads to

$$-(1-\delta)g_k^T d_k < \frac{1}{2}\lambda^{-1}\alpha_k M\|d_k\|^2. \tag{3.7}$$

Using (1.2) and (2.1), we have

$$-g_k^T d_k \geq \frac{1}{2} d_k^T B_k d_k. \tag{3.8}$$

Combining (3.7) and (3.8), we obtain

$$(1-\delta) d_k^T B_k d_k < \lambda^{-1} \alpha_k M \|d_k\|^2. \tag{3.9}$$

Inequality (3.9), together with Assumption B, imply that (3.4) holds. □

To establish the convergent results for the algorithm, we need to investigate some properties of the line search condition, the following lemma indicates that sequence $\{D_k\}$ is monotone decreasing. Here we need to define a sequence

$$M_k = 1 + \max_{1 \leq i \leq k} \|B_i\| \tag{3.10}$$

for all k.

Lemma 3.4. *Let $\{x_k\}$ be the sequence generated by Algorithm 1. Then sequence $\{D_k\}$ is monotonically decreasing. Furthermore, if there exists a constant $\epsilon > 0$ such that*

$$\|g_k\| \geq \epsilon, \quad \forall k. \tag{3.11}$$

Then

$$D_{k+1} - D_k \leq -(1-\eta)\psi\epsilon \min\{\Delta_k, \epsilon/M_k\} \tag{3.12}$$

is satisfied for all k, where $\psi = \min\{\mu\tau, \frac{\delta\tau(1-\delta)\lambda m}{M}\}$.

Proof. We first show that sequence $\{D_k\}$ is monotonically decreasing.

If $k \in I$, we have from (2.3) and (2.1) that

$$D_k - f(x_{k+1}) \geq \mu(\phi_k(0) - \phi_k(d_k)) \geq \mu\tau\|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\}.$$

This leads to

$$f(x_{k+1}) \leq D_k - \mu\tau\|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\}. \tag{3.13}$$

If $k \in J$, it follows from (2.2), (2.4) and (3.4), that we obtain

$$\begin{aligned} f(x_{k+1}) &\leq D_k - \delta\tau\alpha_k\|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\} \\ &\leq D_k - \frac{\delta\tau(1-\delta)\lambda m}{M}\|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\}. \end{aligned} \tag{3.14}$$

Let $\psi = \min\{\mu\tau, \frac{\delta\tau(1-\delta)\lambda m}{M}\}$, combining (3.13) and (3.14), we have

$$f(x_{k+1}) \leq D_k - \psi\|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\}. \tag{3.15}$$

From (1.9) and (3.15), we obtain for all k

$$\begin{aligned} D_{k+1} &= \eta D_k + (1-\eta) f_{k+1} \\ &\leq \eta D_k + (1-\eta)(D_k - \psi\|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\}) \\ &= D_k - (1-\eta)\psi\|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\}. \end{aligned} \tag{3.16}$$

It is clear from (3.16) that sequence $\{D_k\}$ is monotonically decreasing.

Now, using (3.16), (3.11) and (3.10), we obtain

$$D_{k+1} - D_k \leq -(1-\eta)\psi\epsilon \min\{\Delta_k, \epsilon/M_k\}.$$

Thus (3.12) holds for all k. □

The following important lemma is required when we establish a lower bound for all Δ_k.

Lemma 3.5. *Suppose that Assumption A holds if sequence $\{x_k\}$ does not Converge – that is, there exists a constant $\epsilon > 0$ such that (3.11) holds. Then*

$$\lim_{k\to\infty} \min\{\Delta_k, \epsilon/M_k\} = 0. \tag{3.17}$$

Proof. Due to the definition of D_k, we have $D_1 = f_1$. Lemmas 3.1 and 3.4 give $f_{k+1} \le D_{k+1} \le D_k \le f_1$. Since Assumption A implies that $\{f(x_k)\}$ is bounded below, we know that $\{D_k\}$ is also bounded below. Using (3.12), we have that (3.17) holds immediately. □

In the following lemma, we show that there exists a lower bound for Δ_k, for $k \in J$.

Lemma 3.6. *Let $\{x_k\}$ be the sequence generated by Algorithm 1, and suppose that sequence $\{x_k\}$ does not converge, i.e., that (3.11) holds. Then the following inequalities*

$$\|x_{k+1} - x_k\| \ge \sqrt{\frac{2(1-\mu)\tau\epsilon \min\{\Delta_k, \epsilon/M_k\}}{M - \mu m}} \quad \textit{if } \alpha_k = 1, \tag{3.18}$$

$$\|x_{k+1} - x_k\| > \frac{2(1-\delta)\tau\epsilon\lambda}{M} \min\{1, \epsilon/(\Delta_k M_k)\} \quad \textit{if } \alpha_k < 1 \tag{3.19}$$

and

$$\Delta_k > \epsilon/M_k \tag{3.20}$$

are satisfied for $k \in J$ sufficiently large.

Proof. We first show that the inequalities (3.18) and (3.19) hold for $k \in J$ sufficiently large.

(I) If $i_k = 0$, $\alpha_k = 1$. By Step 5 of Algorithm 1, we have

$$\|x_{k+1} - x_k\| = \|d_k\| \le \Delta_k, \quad \forall k \in J. \tag{3.21}$$

Using $\rho_k < \mu$, (3.2) and (2.3), we obtain

$$f(x_k) - f(x_k + d_k) \le D_k - f(x_k + d_k) \le \mu\left(-g_k^T d_k - \frac{1}{2} d_k^T B_k d_k\right). \tag{3.22}$$

By Taylor's expansion and (3.1), we have

$$f(x_k + d_k) - f(x_k) \le g_k^T d_k + \frac{1}{2} M \|d_k\|^2. \tag{3.23}$$

Combining (3.22) with (3.23), we get

$$-g_k^T d_k - \frac{1}{2} M \|d_k\|^2 \le \mu\left(-g_k^T d_k - \frac{1}{2} d_k^T B_k d_k\right). \tag{3.24}$$

It is clear from (3.24) and Assumption B that

$$-(1-\mu) g_k^T d_k \le \frac{1}{2}(M - \mu m)\|d_k\|^2. \tag{3.25}$$

We have from (3.25) and (2.2) that

$$(1-\mu)\tau \|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\} \le \frac{1}{2}(M - \mu m)\|d_k\|^2. \tag{3.26}$$

Using (3.21), (3.26), (3.11) and (3.10), we obtain

$$(1-\mu)\tau\epsilon \min\{\Delta_k, \epsilon/M_k\} \le \frac{1}{2}(M - \mu m)\|x_{k+1} - x_k\|^2.$$

Thus (3.18) holds.

(II) If $i_k > 0$, i.e., $\alpha_k < 1$. Using (3.5), (3.2), Taylor's expansion, (3.1), (2.2), (3.11) and (3.10), we obtain

$$\begin{aligned}
0 &> D_k - f(x_k + \lambda^{-1}\alpha_k d_k) + \delta\lambda^{-1}\alpha_k g_k^T d_k \\
&\geq f_k - f(x_k + \lambda^{-1}\alpha_k d_k) + \delta\lambda^{-1}\alpha_k g_k^T d_k \\
&\geq -\lambda^{-1}(1-\delta)\alpha_k g_k^T d_k - \frac{1}{2}M\lambda^{-2}\alpha_k^2\|d_k\|^2 \\
&\geq \lambda^{-1}(1-\delta)\tau\epsilon\alpha_k \min\{\Delta_k, \epsilon/M_k\} - \frac{1}{2}M\lambda^{-2}\alpha_k\|d_k\|\|x_{k+1}-x_k\| \\
&\geq \lambda^{-1}\alpha_k\left[(1-\delta)\tau\epsilon\min\{\Delta_k, \epsilon/M_k\} - \frac{1}{2}M\lambda^{-1}\Delta_k\|x_{k+1}-x_k\|\right] \\
&= \lambda^{-1}\alpha_k\Delta_k\left[(1-\delta)\tau\epsilon\min\{1, \epsilon/(\Delta_k M_k)\} - \frac{1}{2}M\lambda^{-1}\|x_{k+1}-x_k\|\right].
\end{aligned}$$

Thus (3.19) holds.

We now prove that (3.20) holds for $k \in J$ sufficiently large in two cases.

Case 1. $\alpha_k = 1$. Assume that (3.20) does not hold, i.e.,

$$\Delta_k \leq \epsilon/M_k. \tag{3.27}$$

It follows from (3.18) that

$$\Delta_k \geq \|x_{k+1}-x_k\| \geq \sqrt{\frac{2(1-\mu)\tau\epsilon\Delta_k}{M-\mu m}}.$$

The above inequality yields

$$\Delta_k \geq \frac{2(1-\mu)\tau\epsilon}{M-\mu m}. \tag{3.28}$$

Inequality (3.27), together with (3.28), imply that

$$\epsilon/M_k \geq \Delta_k \geq \frac{2(1-\mu)\tau\epsilon}{M-\mu m}.$$

This contradicts (3.17).

Case 2. $\alpha_k < 1$. By Steps 3 and 5 of Algorithm 1, we have

$$\|x_{k+1}-x_k\| = \alpha_k\|d_k\| \leq \Delta_k, \quad \forall k \in J. \tag{3.29}$$

Suppose (3.20) does not hold, i.e., $\Delta_k \leq \epsilon/M_k$. Inequality (3.19) gives

$$\|x_{k+1}-x_k\| > 2(1-\delta)\tau\epsilon\lambda/M. \tag{3.30}$$

Then it follows from (3.27), (3.29) and (3.30) that

$$\epsilon/M_k \geq \Delta_k > 2(1-\delta)\tau\epsilon\lambda/M,$$

and this contradicts (3.17). Thus (3.20) holds. □

Based on Lemmas 3.5 and 3.6, we now show that there exists a lower bound for the trust region radius Δ_k, for all sufficiently large k.

Lemma 3.7. *Let $\{x_k\}$ be the sequence generated by Algorithm 1, suppose that sequence $\{x_k\}$ does not converge, i.e., (3.11) holds. Then*

$$\Delta_k > \epsilon/M_k \tag{3.31}$$

for all sufficiently large k.

Proof. If there are only finitely many ks such that $\Delta_{k+1} \leq \Delta_k$, namely, J is a finite set, then there exists a positive constant η^* such that $\Delta_k > \eta^*$ for all k. Note that (3.17) implies that $\lim_{k\to\infty} \epsilon/M_k = 0$. Hence (3.31) holds for all large k.

Now, we assume that J is an infinite set. Lemma 3.6 suggests that there exists a $\bar{k} \in J$ such that (3.20) holds for $k \in J$ and $k \geq \bar{k}$. For any $k \in I$ and $k \geq \bar{k}$, let $\check{k} = \max\{i : i \in J \text{ and } i \leq k\}$. The definition of $\check{k}$ suggests that

$$\Delta_{\check{k}} > \epsilon/M_{\check{k}} \tag{3.32}$$

and

$$\check{k} + s \in I \tag{3.33}$$

for all $s = 1, 2, \ldots k - \check{k}$. Using (3.32), (3.33), (2.6) and (2.8), we have

$$\Delta_{\check{k}} \leq \Delta_{\check{k}+1} \leq \Delta_{\check{k}+2} \leq \cdots \leq \Delta_k. \tag{3.34}$$

(3.34) together with (3.32) imply that

$$\Delta_k > \epsilon/M_{\check{k}}.$$

Due to the monotonicity of $\{M_k\}$, we have

$$M_k \geq M_{\check{k}}.$$

Thus (3.31) holds. □

Now, we establish the global convergence for Algorithm 1. Similarly to [9], we assume that $\|B_k\|$ does not grow too rapidly.

Theorem 3.1. *Suppose that Assumptions A and B hold, and $\{B_k\}$ satisfies*

$$\sum_{k=1}^{\infty} 1/M_k = \infty. \tag{3.35}$$

Then sequence $\{x_k\}$ generated by Algorithm 1 satisfies

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.36}$$

Proof. Assume that (3.36) does not hold, then there exists a constant ϵ such that (3.11) holds. It follows from (3.31) that

$$\Delta_k > \epsilon/M_k \tag{3.37}$$

for sufficiently large k. From (3.12), we obtain

$$(1-\eta)\psi\epsilon \min\{\Delta_k, \epsilon/M_k\} \leq D_k - D_{k+1}. \tag{3.38}$$

Using (3.37) and (3.38), we have

$$\sum_{k=1}^{\infty}(1-\eta)\psi\epsilon^2/M_k \leq \sum_{k=1}^{\infty}(D_k - D_{k+1}).$$

Thus

$$\sum_{k=1}^{\infty} 1/M_k < \infty, \tag{3.39}$$

This contradicts (3.35). Therefore (3.36) holds. □

In general, the superlinear convergence result of Algorithm 1 requires further assumptions; we state the assumptions and main result in the following theorem.

Theorem 3.2. *Suppose that subproblem (1.2) is solved accurately, i.e.,* $d_k = -B_k^{-1} g_k$. *Suppose also that*

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x_*))d_k\|}{\|d_k\|} = 0. \tag{3.40}$$

Then if $\{x_k\}$ *converges to a point* x_* *such that* $\nabla^2 f(x_*)$ *is positive definite, the rate of convergence is superlinear, i.e.,*

$$\|x_{k+1} - x_*\| = o(\|x_k - x_*\|).$$

Proof. Let $x_{k+1} = x_k + t_k d_k$, where

$$t_k = \begin{cases} 1 & \text{if } \rho_k \ge \mu; \\ \alpha_k & \text{if } \rho_k < \mu. \end{cases} \tag{3.41}$$

We will verify that $t_k \equiv 1$ for k sufficiently large. Since $\{x_k\}$ is convergent and $\alpha_k > (1-\delta)\lambda m/M$, we have $d_k \to 0$ as $k \to \infty$. $d_k = -B_k^{-1} g_k$ and (3.40) imply that

$$\lim_{k\to\infty} \frac{\|g_k - \nabla^2 f(x_*) B_k^{-1} g_k\|}{\|B_k^{-1} g_k\|} = 0.$$

Thus

$$g_k^T B_k^{-1} g_k = (B_k^{-1} g_k)^T \nabla^2 f(x_*)(B_k^{-1} g_k) + o(\|B_k^{-1} g_k\|^2). \tag{3.42}$$

By Lemma 3.1, Taylor's expansion and (3.42), we obtain

$$\begin{aligned} f(x_k - B_k^{-1} g_k) - D_k + \delta g_k^T B_k^{-1} g_k &\le f(x_k - B_k^{-1} g_k) - f_k + \delta g_k^T B_k^{-1} g_k \\ &= -(1-\delta) g_k^T B_k^{-1} g_k + \frac{1}{2} B_k^{-1} g_k^T \nabla^2 f(\omega_k) B_k^{-1} g_k \\ &= -\left(\frac{1}{2} - \delta\right)(B_k^{-1} g_k)^T \nabla^2 f(x_*)(B_k^{-1} g_k) + o(\|B_k^{-1} g_k\|^2), \end{aligned}$$

where $\omega_k \in (x_k, x_k - B_k^{-1} g_k)$. Since $\nabla^2 f(x_*)$ is positive definite, we have

$$f(x_k - B_k^{-1} g_k) - D_k + \delta g_k^T B_k^{-1} g_k \le 0,$$

This inequality leads to

$$f(x_k + d_k) \le D_k + \delta g_k^T d_k$$

for $k \in J$ sufficiently large. In other words, $\alpha_k = 1$ satisfies the line search condition (2.4) for $k \in J$ sufficiently large. $\square$

By the definition of t_k, we see that $x_{k+1} = x_k + d_k$ holds for all sufficiently large k. Therefore, Algorithm 1 becomes the Newton method or quasi-Newton method for sufficiently large k. The superlinear convergence result can be established as Proposition 1.3.2 in [20].

4. Numerical results

In this section, we present preliminary computational results to illustrate the performance of the nonmonotone strategy (1.8). We choose test functions from Moré, Garbow and Hillstrom [21]. All our experiments are performed in MATLAB Version 6.5. Some fixed parameter values are given as follows

$$\Delta_1 = 0.5, \qquad \lambda = 0.5, \qquad \delta = 0.4, \qquad k_{\max} = 500.$$

Table 1
Comparisons of Mtrls and Ntrls2

Problem	Dim	Mtrls	Ntrls2
		$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$
Freudenstein and Roth	2	15/16/16/0	15/16/16/0
Beale	2	16/17/17/0	16/17/17/0
Helical valley	3	33/36/36/2	33/36/36/2
Bard	3	25/26/26/0	25/26/26/0
Gulf research and development	3	40/43/43/2	42/43/43/0
Box three-dimensional	3	54/55/55/0	54/55/55/0
Powell singular	4	36/38/38/1	51/52/52/0
Wood	4	41/42/42/0	41/42/42/0
Osborne 2	11	61/67/67/5	60/65/65/4
Extended Rosenbrock	1000	52/56/56/3	52/54/54/1
	1500	50/55/55/4	49/52/52/2
	2000	56/60/60/3	51/54/54/2
Extended Powell Singular	1000	77/78/78/0	75/76/76/0
	1500	94/96/96/1	79/80/80/0
	2000	102/104/104/1	84/86/86/1
Discrete integral equation	1000	13/14/14/0	13/14/14/0
	2000	14/15/15/0	14/15/15/0
Extended Beale	1000	28/29/29/0	24/25/25/0
	2000	35/36/36/0	31/32/32/0
Broyden tridiagonal	1000	109/110/110/0	105/106/106/0
	2000	115/116/116/0	108/109/109/0
Broyden banded	1000	110/111/111/0	110/111/111/0
	2000	125/126/126/0	114/115/115/0
Linear function — full rank	1000	128/129/129/0	101/102/102/0

B_k is updated by BFGS formula

$$B_{k+1} = B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k},$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$, we let $B_{k+1} = B_k$ as $s_k^T y_k \le 0$. In all tests, the initial matrix B_1 is chosen as $|f_1|I$, where I is the identify matrix. The stopping condition is

$$\max\{\|g(x_k)\|, \|f(x_k) - f(x_*)\|\} \le 10^{-6}.$$

Firstly, to illustrate that the nonmonotone strategy (1.8) is not inferior to the corresponding monotone strategy, especially for larger-dimensional test problems. We give test results in Table 1. Secondly, to show that the nonmonotone strategy (1.8) is as effective as the nonmonotone strategy (1.5), we provide results in Tables 2 and 3. In Table 2, parameter η in both nonmonotone strategies is of fixed value. In Table 3, parameters η_k are varied. Finally, the numerical results of Ntrls2 under different parameter values μ are presented in Table 4.

Notations listed in tables are defined as:

- Problem: the names of the test problems;
- Dim: the dimensions of the problems;
- n_i: the number of iterations;
- n_f: the number of function evaluations;
- n_g: the number of gradient evaluations;
- n_s: the number of line searches;
- Mtrls: the Algorithm 3.1 given by Nocedal and Yuan [9];
- Ntrls1: the nonmonotone strategy (1.5) is used in Algorithm 1;
- Ntrls2: the nonmonotone strategy (1.8) is used in Algorithm 1.

Table 2
Comparisons of Ntrls1 and Ntrls2 with fixed values η_k

Problem	Dim	Ntrls1(0.15)	Ntrls1(0.85)	Ntrls2(0.15)	Ntrls2(0.85)
		$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$
FROTH	2	15/16/16/0	15/16/16/0	15/16/16/0	15/16/16/0
BEALE	2	16/17/17/0	16/17/17/0	16/17/17/0	16/17/17/0
HELIX	3	33/36/36/2	31/32/32/0	33/36/36/2	31/32/32/0
BARD	3	25/26/26/0	25/26/26/0	25/26/26/0	25/26/26/0
GULF	3	42/43/43/0	42/43/43/0	42/43/43/0	42/43/43/0
BOX	3	54/55/55/0	55/56/56/0	54/55/55/0	55/56/56/0
SING	4	51/52/52/0	51/52/52/0	51/52/52/0	51/52/52/0
WOOD	4	41/42/42/0	41/42/42/0	41/42/42/0	41/42/42/0
OSB2	11	60/65/65/4	69/72/72/2	60/65/65/4	65/69/69/3
ROSEX	100	45/48/48/2	54/56/56/1	45/48/48/2	54/56/56/1
	300	46/52/52/5	60/61/61/0	46/52/52/5	60/61/61/0
	500	53/57/57/3	63/66/66/2	53/57/57/3	73/76/76/2
	1000	54/57/57/2	66/67/67/0	54/57/57/2	60/63/63/2
SINGX	200	75/77/77/1	75/76/76/0	75/77/77/1	75/76/76/0
	500	79/81/81/1	85/86/86/0	79/81/81/1	85/86/86/0
	1000	75/76/76/0	75/76/76/0	75/76/76/0	75/76/76/0
PEN1	10	103/107/107/3	183/184/184/0	103/107/107/3	190/191/191/0
	20	170/175/175/4	227/228/228/0	170/175/175/4	227/228/228/0
	40	361/365/365/3	393/394/394/0	361/365/365/3	393/394/394/0
PEN2	10	285/311/311/25	402/415/415/12	285/311/311/25	395/407/407/11
	20	369/383/383/13	446/451/451/4	369/383/383/13	446/451/451/4
	30	426/431/431/4	474/475/475/0	426/431/431/4	474/475/475/0
VARDIM	100	40/41/41/0	39/40/40/0	40/41/41/0	40/41/41/0
IE	500	8/9/9/0	10/11/11/0	8/9/9/0	8/9/9/0
	1000	13/14/14/0	13/14/14/0	13/14/14/0	13/14/14/0
	2000	14/15/15/0	14/15/15/0	14/15/15/0	14/15/15/0
TRID	200	47/48/48/0	47/48/48/0	47/48/48/0	47/48/48/0
	500	79/80/80/0	79/80/80/0	79/80/80/0	79/80/80/0
BAND	200	84/85/85/0	84/85/85/0	84/85/85/0	84/85/85/0
	500	97/98/98/0	97/98/98/0	97/98/98/0	97/98/98/0
	1000	110/111/111/0	110/111/111/0	110/111/111/0	110/111/111/0
LIN	500	91/92/92/0	91/92/92/0	91/92/92/0	91/92/92/0

To investigate whether the nonmonotone trust region strategy is superior to monotone trust region strategy, we compare our algorithm with the Algorithm 3.1 proposed by Nocedal and Yuan [9], which is a pioneering work incorporating line search into the trust region method. In this test, both parameters μ and η are set to be 0.25.

Table 1 suggests that nonmonotone linesearch (1.8) is quite promising. We can see in Table 1 that for most larger-dimensional test problems, Ntrls2 takes fewer iterations and function evaluations than Mtrls.

In the second experiment, parameter η, both in (1.5) and (1.8), is fixed value. We consider two cases, say, $\eta = 0.15$ and $\eta = 0.85$. Parameter μ is given as 0.25. Other parameter values are declared as above.

In the third experiment, we consider various parameters η_k. In theory, the best convergence results were obtained by dynamically varying η_k if the iterates were far away from the optimum, using values closer to 1, and if not, using values closer to 0. Therefore we set $\eta_k = 0.95$ for $k \in [1, 20]$, $\eta_k = 0.5$ for $k \in [21, 40]$, $\eta_k = 0.25$ for $k \in [41, 60]$ and $\eta_k = 0.01$ for $k \geq 61$. This test is done under two different values of μ, namely, $\mu = 0.25$ and $\mu = 0.75$. Other parameter values are similar to the above experiment.

Tables 2 and 3 show that the results of nonmonotone strategies (1.8) and (1.5) are almost identical for the given problems.

Finally, we give an experiment to observe the impacts of μ. As we shall see, the numbers for using line search depend on the value of μ. If μ tends to zero, it means that the trial step can be accepted even if the ratios between the actual reductions and the predicted reductions are very small. In this case, the line search is used infrequently in the algorithm. On the other hand, if μ tends to 1, $\rho_k \geq \mu$ is not necessarily satisfied for many k's. Consequently, the line

Table 3
Comparisons of Ntrls1 and Ntrls2 with varied values η_k

Problem	Dim	Ntrls1 ($\mu = 0.25$)	Ntrls1 ($\mu = 0.75$)	Ntrls2 ($\mu = 0.25$)	Ntrls2 ($\mu = 0.75$)
		$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$
FROTH	2	15/16/16/0	15/16/16/0	15/16/16/0	16/18/18/1
BEALE	2	16/17/17/0	16/17/17/0	16/17/17/0	15/17/17/1
HELIX	3	31/32/32/0	31/32/32/0	31/32/32/0	31/32/32/0
BARD	3	25/26/26/0	25/26/26/0	25/26/26/0	24/26/26/1
GULF	3	42/43/43/0	42/43/43/0	42/43/43/0	56/59/59/2
BOX	3	55/56/56/0	55/56/56/0	55/56/56/0	55/56/56/0
SING	4	51/52/52/0	51/52/52/0	51/52/52/0	50/52/52/1
WOOD	4	41/42/42/0	41/42/42/0	41/42/42/0	41/42/42/0
OSB2	11	63/67/67/3	92/96/96/3	62/67/67/4	62/68/68/5
ROSEX	100	51/53/53/1	51/53/53/1	51/53/53/1	51/53/53/1
	300	54/56/56/1	54/56/56/1	54/56/56/1	54/56/56/1
	500	52/56/56/3	51/56/56/4	52/56/56/3	51/56/56/4
	1000	51/55/55/3	51/55/55/3	51/55/55/3	51/55/55/3
SINGX	200	75/77/77/1	75/77/77/1	75/77/77/1	75/77/77/1
	500	93/94/94/0	86/88/88/1	93/94/94/0	86/88/88/1
	1000	75/76/76/0	75/76/76/0	75/76/76/0	75/76/76/0
PEN1	10	110/113/113/2	101/105/105/3	110/113/113/2	101/105/105/3
	20	172/176/176/3	172/176/176/3	175/177/177/1	175/177/177/1
	40	363/367/367/3	364/368/368/3	364/367/367/2	365/369/369/3
PEN2	10	296/321/321/24	279/306/306/26	286/309/309/22	259/286/286/26
	20	406/421/421/14	376/393/393/16	406/421/421/14	376/393/393/16
	30	443/447/447/3	435/439/439/3	449/454/454/4	444/451/451/6
VARDIM	100	40/41/41/0	40/41/41/0	39/40/40/0	40/41/41/0
IE	500	8/9/9/0	8/9/9/0	10/11/11/0	8/9/9/0
	1000	13/14/14/0	13/14/14/0	13/14/14/0	13/14/14/0
	2000	14/15/15/0	14/15/15/0	14/15/15/0	14/15/15/0
TRID	200	47/48/48/0	47/48/48/0	47/48/48/0	47/48/48/0
	500	79/80/80/0	79/80/80/0	79/80/80/0	79/80/80/0
BAND	200	84/85/85/0	84/85/85/0	84/85/85/0	84/85/85/0
	500	97/98/98/0	97/98/98/0	97/98/98/0	97/98/98/0
	1000	110/111/111/0	110/111/111/0	110/111/111/0	110/111/111/0
LIN	500	91/92/92/0	91/92/92/0	91/92/92/0	91/92/92/0

search would be used frequently in the algorithm. In terms of Tables 2 and 3, we find that the varieties of η_k have little effect on performance for the given problems. Therefore in this test, we let η_k be a fixed value of 0.5.

Table 4 implies that, the parameter values of μ have little effect on computation. This observation implies that the ratios between the actual reduction $D_k - f(x_k + d_k)$ and the predicted reduction $\phi_k(0) - \phi_k(d_k)$ are large enough in most of the cases.

5. Concluding remarks

This paper presents a nonmonotone technique based on the nonmonotone technique proposed by Zhang and Hager [16]. The main distinction of techniques (1.8) and (1.5) is that the nonmonotone term of the former is only a simple convex combination of its previous one and f_k. However, the latter induces the complex η_k and Q_k. Based on (1.8), we developed a nonmonotone trust region algorithm, and the actual reduction and line search in our algorithm are nonmonotonic.

We have established global and superlinear properties and demonstrated how this technique can be efficiently implemented. Finally, we have provided the computational results of our preliminary numerical experiments. The theoretical and the computational results indicate that this nonmonotone technique has considerable practical utility for solving unconstrained optimization problems.

Table 4
Numerical results of Ntrls2 with $\eta = 0.5$ under different values of μ

Problem	Dim	Ntrls2 ($\mu = 0.25$)	Ntrls2 ($\mu = 0.5$)	Ntrls2 ($\mu = 0.75$)	Ntrls2 ($\mu = 1$)
		$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$	$n_i/n_f/n_g/n_s$
FROTH	2	15/16/16/0	15/16/16/0	16/18/18/1	16/18/18/1
BEALE	2	16/17/17/0	16/17/17/0	15/17/17/1	15/17/17/1
HELIX	3	31/32/32/0	31/32/32/0	31/32/32/0	31/32/32/0
BARD	3	25/26/26/0	25/26/26/0	24/26/26/1	24/26/26/1
GULF	3	42/43/43/0	42/43/43/0	51/54/54/2	49/53/53/3
BOX	3	55/56/56/0	55/56/56/0	55/56/56/0	55/56/56/0
SING	4	51/52/52/0	51/52/52/0	50/52/52/1	50/52/52/1
WOOD	4	41/42/42/0	41/42/42/0	41/42/42/0	41/42/42/0
OSB2	11	63/68/68/4	63/68/68/4	60/67/67/6	59/66/66/6
ROSEX	100	49/52/52/2	49/53/53/3	49/53/53/3	42/49/49/6
	300	48/53/53/4	48/53/53/4	48/53/53/4	46/52/52/5
	500	53/57/57/3	53/57/57/3	53/58/58/4	52/57/57/4
	1000	51/55/55/3	51/55/55/3	51/55/55/3	50/54/54/3
SINGX	200	75/77/77/1	75/77/77/1	75/77/77/1	75/77/77/1
	500	93/94/94/0	86/88/88/1	86/88/88/1	86/88/88/1
	1000	75/76/76/0	75/76/76/0	75/76/76/0	75/76/76/0
PEN1	10	110/113/113/2	107/111/111/3	101/105/105/3	101/105/105/3
	20	175/177/177/1	175/177/177/1	175/177/177/1	169/174/174/4
	40	364/367/367/2	364/367/367/2	365/369/369/3	364/369/369/4
PEN2	10	281/302/302/20	300/323/323/22	300/334/334/33	282/318/318/35
	20	420/433/433/12	411/430/430/18	394/406/406/11	400/416/416/15
	30	431/433/433/1	434/437/437/2	434/437/437/2	431/435/435/3
VARDIM	100	40/41/41/0	40/41/41/0	40/41/41/0	40/41/41/0
IE	500	8/9/9/0	8/9/9/0	8/9/9/0	9/11/11/1
	1000	13/14/14/0	13/14/14/0	13/14/14/0	13/14/14/0
	2000	14/15/15/0	14/15/15/0	14/15/15/0	14/15/15/0
TRID	200	47/48/48/0	47/48/48/0	47/48/48/0	47/48/48/0
	500	79/80/80/0	79/80/80/0	79/80/80/0	79/80/80/0
BAND	200	84/85/85/0	84/85/85/0	84/85/85/0	84/85/85/0
	500	97/98/98/0	97/98/98/0	97/98/98/0	97/98/98/0
	1000	110/111/111/0	110/111/111/0	110/111/111/0	110/111/111/0
LIN	500	91/92/92/0	91/92/92/0	91/92/92/0	91/92/92/0

Acknowledgments

We are grateful to the anonymous referees for their careful reading of the manuscript and many helpful comments, which improved the presentation of this paper.

References

[1] K. Levenberg, A method for the solution of certain nonlinear problems in least squares, Quart. Appl. Math. 2 (1994) 164–168.
[2] D.W. Marquardt, An algorithm for least squares estimation with nonlinear parameters, SIAM J. Appl. Math. 11 (1963) 431–441.
[3] S. Goldfeld, R. Quandt, H. Trotter, Maximization by quadratic hill climbing, Econometrica 34 (1966) 541–551.
[4] M.J.D. Powell, Convergence properties of a class of minimization algorithm, in: O.L. Mangasarian, R.R. Meyer, S.M. Robinson (Eds.), Nonlinear Programming II, Academic Press, New York, 1975, pp. 1–27.
[5] R. Fletcher, An algorithm for solving linearly constrained optimization problems, Math. Program. 2 (1972) 133–165.
[6] Y. Yuan, On a subproblem of trust region algorithms for constrained optimization, Math. Program. 47 (1990) 53–63.
[7] M.J.D. Powell, Y. Yuan, A trust region algorithm for equality constrained optimization, Math. Program. 49 (1990) 189–213.
[8] A.R. Conn, N.I.M. Gould, Ph.L. Toint, Trust-Region Methods, SIAM Publications, Philadelphia, PA, 2000.
[9] J. Nocedal, Y. Yuan, Combining trust region and line search techniques, in: Y. Yuan (Ed.), Advances in Nonlinear Programming, Kluwer Academic Publishers, Dordrecht, 1996, pp. 153–175.
[10] E.M. Gertz, Combination trust-region line search methods for unconstrained optimization, University of California, San Diego, 1999.
[11] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's method, SIAM J. Numer. Anal. 23 (4) (1986) 707–716.
[12] N.Y. Deng, Y. Xiao, F.J. Zhou, Nonmonotone trust region algorithm, J. Optim. Theory Appl. 76 (2) (1993) 259–285.

[13] X. Ke, J. Han, A class of nonmonotone trust region algorithms for unconstrained optimization, Sci. China Ser. A 41 (9) (1998) 927–932.
[14] W. Sun, Nonmonotone trust region method for solving optimization problems, Appl. Math. Comput. 156 (1) (2004) 159–174.
[15] J. Fu, W. Sun, Nonmonotone adaptive trust-region method for unconstrained optimization problems, Appl. Math. Comput. 163 (1) (2005) 489–504.
[16] H. Zhang, W.W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM J. Optim. 14 (4) (2004) 1043–1056.
[17] P.L. Toint, An assessment of non-monotone line search techniques for unconstrained optimization, SIAM J. Sci. Comput. 17 (1996) 725–739.
[18] Y.H. Dai, On the nonmonotone line search, J. Optim. Theory Appl. 112 (2) (2002) 315–330.
[19] J. Mo, C. Liu, S. Yan, A nonmonotone trust region method based on nonincreasing technique of weighted average of the successive function values, J. Comput. Appl. Math. (2006) doi:10.1016/j.cam.2006.10.070.
[20] D.P. Bertsekas, Nonlinear Programming, second ed., Athena Scientific, Belmont, MA, 1999.
[21] J.J. Moré, B.S. Garbow, K.E. Hillstrome, Testing unconstrained optimization software, ACM Trans. Math. Software 7 (1) (1981) 17–41.