Journal of Mathematics in Industry
a SpringerOpen Journal

RESEARCH Open Access

A new nonmonotone adaptive trust region line search method for unconstrained optimization

Xinyi Wang[1], Xianfeng Ding[1*] and Quan Qu[1]

*Correspondence: fxxd@163.com
[1] School of Science, Southwest Petroleum University, Chengdu, China

Abstract

This paper proposes a new nonmonotone adaptive trust region line search method for solving unconstrained optimization problems, and presents a modified trust region ratio, which obtained more reasonable consistency between the accurate model and the approximate model. The approximation of Hessian matrix is updated by the modified BFGS formula. Trust region radius adopts a new adaptive strategy to overcome additional computational costs at each iteration. The global convergence and superlinear convergence of the method are preserved under suitable conditions. Finally, the numerical results show that the proposed method is very efficient.

Keywords: Unconstrained optimization; Trust region method; Nonmonotone adaptive; Convergence

1 Introduction

Consider the following unconstrained optimization problem

$$\min_{x\in R^n} f(x), \tag{1}$$

where $f: R^n \to R$ is a twice continuously differentiable function. Trust region method is one of prominent class of iterative methods. The basic idea of trust region methods as follows: at the current step x_k, the trial step d_k is obtained by solving the subproblem:

$$\min_{d\in R^n} m_k(d) = g_k^T d + \frac{1}{2} d^T B_k d, \tag{2}$$

$$\|d\| \leq \Delta_k,$$

where $f_k = f(x_k)$, $g_k = \nabla f(x_k)$, $G_k = \nabla^2 f(x_k)$, B_k be a symmetric approximation of G_k, Δ_k is trust region radius, and $\|\cdot\|$ is the Euclidean norm.

To evaluate an agreement between the model and the objective function, the most ordinary ratio is defined as follows:

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

where the numerator is called the actual reduction and the denominator is called the predicted reduction. The ratio ρ_k is used to determine whether the trial step d_k is accepted. Given $\mu \in [0,1]$, if $\rho_k < \mu$, the trial step d_k is not successful and the subproblem (2) should be resolved with a smaller radius. Otherwise, d_k is acceptable and the radius should be increased.

It is well-known that monotone techniques may slow down the rate of convergence, especially in the presence of the narrow curved valley. The monotone techniques that require the objective function to be decreased at each iteration. In order to overcome these disadvantages, Grippo et al. [1] proposed a nonmonotone technique for Newton's method in 1986. In 1998, Nocedal and Yuan [2] proposed a nonmonotone trust region method with line search techniques, the step size α_k satisfies the following inequality:

$$f(x_k + \alpha_k d_k) \le f_{l(k)} + \sigma \alpha_k g_k^T d_k, \tag{4}$$

where $\sigma \in (0,1)$. The general nonmonotone term $f_{l(k)}$ is defined by $f_{l(k)} = \max_{0 \le j \le m(k)}\{f_{k-j}\}$, in which $m(0) = 0$, $0 \le m(k) \le \min\{m(k-1)+1, N\}$ and $N \ge 0$ is an integer constant.

However, the general nonmonotone strategy does not sufficiently employ the current value of the objective function f. It seems that the nonmonotone term has well performance far from the optimum. In order to introduce a more relaxed nonmonotone strategy, Ahookhosh et al. [3] introduced a modified nonmonotone term in 2002. More precisely, for $\sigma \in (0,1)$, the step size α_k satisfies the following inequality:

$$f(x_k + \alpha_k d_k) \le R_k + \sigma \alpha_k g_k^T d_k, \tag{5}$$

where the nonmonotone term R_k is defined by

$$R_k = \eta_k f_{l(k)} + (1 - \eta_k) f_k, \tag{6}$$

in which $\eta_k \in [\eta_{\min}, \eta_{\max}]$, with $\eta_{\min} \in [0,1)$, and $\eta_{\max} \in [\eta_{\min}, 1]$.

One knows that an adaptive radius avoid the blindness of updating the initial trust region radius, and may cause the decrease in the total number of iterations. In 1997, Sartenear [4] proposed a new strategy for automatically determining the initial trust region radius. In 2002, Zhang et al. [5] proposed a new scheme to determine trust region radius as follows: $\Delta_k = c^p \|\widehat{B}_k^{-1}\| \|g_k\|$. To avoid calculating the inverse of the matrix B_k and an estimation of $\widehat{B}_k^{-1}$ in each iteration, Li [6] proposed an adaptive trust region radius as follows: $\Delta_k = \frac{\|d_{k-1}\|}{\|y_{k-1}\|}\|g_k\|$, where $y_{k-1} = g_k - g_{k-1}$. Inspired by these facts, some modified versions of adaptive trust region methods have been proposed in [7–14].

This paper is organized as follows. In Sect. 2, we describe the new algorithm. The global and superlinear convergence of the algorithm are established in Sect. 3. In Sect. 4, numerical results are reported, which show that the new method is effective. Finally, conclusions are drawn in Sect. 5.

2 New algorithm

In this section, a new adaptive nonmonotone trust region line search algorithm is proposed. Here, based on the method of Li [6], we proposed a adaptive trust region radius as follows:

$$d_k \leq \Delta_k := c_k \frac{\|d_{k-1}\|}{\|y_{k-1}\|} \|g_k\|, \tag{7}$$

c_k is an adjustment parameter. Prompted by the adaptive technique, the proposed method has the following well properties: it is convenient to adjust the radius by using the adjustment parameter c_k, and the algorithm also reduces the related workload and calculation time.

On the basis of considered discussion, at each iteration, a trial step d_k is obtained by solving the following trust region subproblem:

$$\begin{aligned} &\min_{d \in R^n} m_k(d) = g_k^T d + \frac{1}{2} d^T B_k d, \\ &\|d\| \leq \Delta_k := c_k \frac{\|d_{k-1}\|}{\|y_{k-1}\|} \|g_k\|, \end{aligned} \tag{8}$$

where $y_{k-1} = g_k - g_{k-1}$. The matrix B_k is updated by a modified BFGS formula [11],

$$B_{k+1} = \begin{cases} B_k + \frac{z_k z_k^T}{z_k^T d_k} - \frac{B_k d_k d_k^T B_k}{d_k^T B_k d_k}, & y_k^T d_k > 0, \\ B_k, & y_k^T d_k \leq 0, \end{cases} \tag{9}$$

where $d_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$, $z_k = y_k + t_k \|g_k\| d_k$, $t_k = 1 + \max\{-\frac{y_k^T d_k}{\|g_k\| \|d_k\|}, 0\}$.

Considering advantage of the Ahookhosh's nonmonotone term, the best convergence behavior can be obtained by adopting a stronger nonmonotone strategy away from the solution and a weaker monotone strategy closer to the solution. We defined a modified form of trust region ratio as follows:

$$\widehat{\rho}_k = \frac{R_k - f(x_k + d_k)}{f_{l(k)} - f_k - m_k(d_k)}, \tag{10}$$

As seen, the effect of nonmonotonicity can be controlled in (10) by numerator and denominator.

Now, we list the new adaptive nonmonotone trust region line search algorithm as follows:

Algorithm 2.1 (New nonmonotone adaptive trust region algorithm)

Step 0. Given initial point $x_0 \in R^n$, a symmetric matrix $B_0 \in R^n \times R^n$. The constants $0 < \mu_1 < \mu_2 < 1$, $0 < \eta_{\min} \leq \eta_{\max} < 1$, $0 < \beta_1 < 1 < \beta_2$, $0 < \delta_1 < 1 < \delta_2$, $N > 0$ and $\varepsilon > 0$ are also given. Set $k = 0$, $c_0 = 1$.

Step 1. If $\|g_k\| \leq \varepsilon$, then stop. Otherwise, go to Step 2.

Step 2. Solve the subproblem (8) to obtain d_k.

Step 3. Compute R_k and $\widehat{\rho}_k$ respectively.

Step 4.

$$c_{k+1} := \begin{cases} \beta_1 c_k, & \text{if } \widehat{\rho}_k < \mu_1, \\ c_k, & \text{if} \mu_1 \leq \widehat{\rho}_k < \mu_2, \\ \beta_2 c_k, & \text{if } \widehat{\rho}_k \geq \mu_2. \end{cases}$$

Step 5. If $\widehat{\rho}_k \geq \mu_1$, set $x_{k+1} = x_k + d_k$ and go to Step 6. Otherwise, find the step size α_k satisfying (5). Set $x_{k+1} = x_k + \alpha_k d_k$, go to Step 6.

Step 6. Update the trust region radius by $\Delta_{k+1} = c_{k+1} \frac{\|x_{k+1}-x_k\|}{\|g_{k+1}-g_k\|} \|g_{k+1}\|$ and go to Step 7.

Step 7. Compute the new Hessian approximation B_{k+1} by a modified BFGS formula (9). Set $k = k + 1$ and go to Step 1.

Assumption 2.1

H1. The level set $L(x_0) = \{x \in R^n | f(x) \leq f(x_0)\} \subset \Omega$, where $\Omega \in R^n$ is bounded. $f(x)$ is continuously differentiable on the level set $L(x_0)$.

H2. The matrix B_k is uniformly bounded, i.e., there exists a constant $M_1 > 0$ such that $\|B_k\| \leq M_1$, $\forall k \in N \cup \{0\}$.

Remark 2.1 If f is a twice continuously differentiable function, then H1 implies that ∇f is continuous and uniformly bounded on Ω. Hence, there exists a constant L such that

$$\bigl\| \nabla f(x) - \nabla f(y) \bigr\| \leq L \|x - y\|, \quad \forall x, y \in \Omega \tag{11}$$

3 Convergence analysis

Lemma 3.1 *There is a constant $\tau \in (0,1)$, the trial step d_k satisfies the following inequalities*:

$$m_k(0) - m_k(d_k) \geq \tau \|g_k\| \min\left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\}, \tag{12}$$

$$g_k^T d_k \leq -\tau \|g_k\| \min\left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\}. \tag{13}$$

Proof The proof is exactly similar to the proof of Lemma 6 and Lemma 7 of [15] and here is omitted. □

Lemma 3.2 *Suppose that Assumption 2.1 holds, then we have*,

$$f_{l(k)} - f_k - m_k(d_k) \geq \frac{\beta_1^{p_k} \|g_k\|^2}{2M_1}, \tag{14}$$

where p_k is the iteration of the solution to subproblem from the previous trial step d_{k-1} to the currently acceptable trial step d_k.

Proof According to Step 4 of Algorithm 2.1, the trust region radius satisfies $\Delta_k \geq c_k \frac{\|g_k\|}{\|B_k\|} \geq \frac{\beta_1^{p_k} \|g_k\|}{\|B_k\|} \geq \frac{\beta_1^{p_k} \|g_k\|}{M_1}$. Thus, according to $\|d_k\| \leq \Delta_k$, we assume that $\overline{d}_k = \frac{\beta_1^{p_k} \|g_k\|}{M_1}$ is a feasible solution to trust region subproblem. Therefore, we obtain,

$$\begin{aligned} f_{l(k)} - f_k - m_k(d_k) &\geq m_k(0) - m_k(d_k) \\ &\geq -\left(g_k^T d_k + \frac{1}{2} d_k^T B_k d_k \right) \\ &\geq -\left(g_k^T \overline{d}_k + \frac{1}{2} \overline{d}_k^T B_k \overline{d}_k \right) \end{aligned}$$

$$\begin{aligned} &= \frac{\beta_1^{p_k}\|g_k\|^2}{M_1} - \frac{\beta_1^{p_k}\|g_k\|^2}{2M_1} \\ &= \frac{\beta_1^{p_k}\|g_k\|^2}{2M_1}. \end{aligned} \tag{15}$$

□

Lemma 3.3 *Suppose that the sequence* $\{x_k\}$ *is generated by Algorithm* 2.1. *Then we have,*

$$R_k \le f_{l(k)}. \tag{16}$$

Proof Using $R_k = \eta_k f_{l(k)} + (1-\eta_k)f_k$ and $f_k \le f_{l(k)}$, we have

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

□

Lemma 3.4 *Suppose that Assumption* 2.1 *holds. Step* 4 *and Step* 5 *of Algorithm* 2.1 *are well-defined.*

Proof Set $\overline{d}_k = \frac{\beta_1^{p_k}\|g_k\|}{M_1}$ is a solution of subproblem (8) corresponding to $p_k = p$.

Firstly, we prove that $\widehat{\rho}_k \ge \mu_1$, for sufficiently large p. Using Lemma 3.1, Lemma 3.2 and Taylor's formula, we have

$$\begin{aligned} |\widehat{\rho}_k - 1| &= \left|\frac{R_k - f(x_k+d_k)}{f_{l(k)} - f_k - m_k(d_k)} - 1\right| \\ &= \frac{|R_k - f(x_k+d_k) - f_{l(k)} + f_k + m_k(d_k)|}{f_{l(k)} - f_k - m_k(d_k)} \\ &\le \frac{|f_k - f(x_k+d_k) + m_k(d_k)|}{f_{l(k)} - f_k - m_k(d_k)} \\ &\le \frac{o(\|d_k\|^2)}{\frac{\beta_1^p}{2M_1}\|g_k\|^2} \to 0 \quad (p\to\infty). \end{aligned}$$

Therefore, we have $\widehat{\rho}_k \ge \mu_1$, for sufficiently large p. This implies that Steps 4 and 5 of Algorithm 2.1 are well-defined. □

Lemma 3.5 *Suppose that Assumption* 2.1 *holds and the sequence* $\{x_k\}$ *is generated by Algorithm* 2.1. *The sequence* $\{f_{l(k)}\}$ *is* (*not monotonically increasing*) *convergent.*

Proof The proof is exactly similar to the proof of Lemma 2.1 and Corollary 2.1 in [3] and here is omitted. □

Lemma 3.6 *Suppose that the sequence* $\{x_k\}$ *is generated by Algorithm* 2.1. *Using* $\|d_k\| \le \Delta_k$, *there exists a constant* κ *such that* $\|d_k\| \le \kappa\|g_k\|$.

Proof From (7) and $\|d_k\| \le \Delta_k$, we observe that

$$\|d_k\| \le c_k \frac{\|d_{k-1}\|}{\|y_{k-1}\|}\|g_k\|. \tag{17}$$

Thus, setting $\kappa = c_k\frac{\|d_{k-1}\|}{\|y_{k-1}\|}$. □

Lemma 3.7 *Suppose that Assumptions 2.1 holds, and the sequence $\{x_k\}$ is generated by Algorithm 2.1. For $\rho_k < \mu_1$, the step size α_k satisfies the following inequality*:

$$\alpha_k \geq \frac{2\rho\tau(\sigma-1)}{M_1\kappa}\min\left\{1, \frac{1}{\kappa M_1}\right\}. \tag{18}$$

Proof Set $\alpha = \frac{\alpha_k}{\rho}$, where $\rho \in (0,1)$. According to Step 5 of Algorithm 2.1 and (5), it is easy to show that

$$R_k + \sigma\alpha g_k^T d_k < f(x_k + \alpha d_k). \tag{19}$$

Using the definition of R_k and Taylor expansion, we have

$$\begin{aligned} f_k + \sigma\alpha g_k^T d_k &\leq R_k + \sigma\alpha g_k^T d_k \\ &\leq f(x_k + \alpha d_k) \\ &\leq f_k + \alpha g_k^T d_k + \frac{1}{2}\alpha^2 d_k^T \nabla^2 f(\xi) d_k \\ &\leq f_k + \alpha g_k^T d_k + \frac{1}{2}\alpha^2 M_1 \|d_k\|^2, \end{aligned}$$

where $\xi \in (x_k, x_{k+1})$. Thus,we get

$$-(1-\sigma)g_k^T d_k \leq \frac{1}{2}\alpha M_1 \|d_k\|^2, \tag{20}$$

On the other hand, form $\|d_k\| \leq \kappa\|g_k\|$ and (13), we can write

$$\begin{aligned} g_k^T d_k &\leq -\tau\|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right\} \\ &\leq -\tau\frac{\|d_k\|}{\kappa}\min\left\{\|d_k\|, \frac{\|d_k\|}{\kappa M_1}\right\} \\ &\leq -\tau\frac{1}{\kappa}\min\left\{1, \frac{1}{\kappa M_1}\right\}\|d_k\|^2. \end{aligned} \tag{21}$$

Hence, combining above inequality and (20), we have

$$-(1-\sigma)\frac{\tau}{\kappa}\min\left\{1, \frac{1}{\kappa M_1}\right\}\|d_k\|^2 \leq \frac{M_1}{2\rho}\alpha_k\|d_k\|^2. \tag{22}$$

Thus, we can obtain (18). □

Lemma 3.8 *Suppose that Assumption 2.1 holds and the sequence $\{x_k\}$ is generated by Algorithm 2.1, then we have,*

$$\lim_{k\to\infty} f(x_{l(k)}) = \lim_{k\to\infty} f(x_k). \tag{23}$$

Proof From Lemma 3.3, we know that Algorithm 2.1 generates an infinite sequence $\{x_k\}$ satisfying $\widehat{\rho}_k \geq \mu_1$, we obtain,

$$\frac{f_{l(k)} - f(x_k + d_k)}{f_{l(k)} - f_k - m_k(d_k)} > \frac{R_k - f(x_k + d_k)}{f_{l(k)} - f_k - m_k(d_k)} \geq \mu_1.$$

Then,

$$\begin{aligned} f_{l(k)} - f(x_k + d_k) &\ge \mu_1\bigl(f_{l(k)} - f_k - m_k(d_k)\bigr) \\ &\ge \mu_1\bigl(m_k(0) - m_k(d_k)\bigr). \end{aligned} \tag{24}$$

Replacing k by $l(k) - 1$, we can write

$$f_{l(l(k)-1)} - f_{l(k)} \ge \mu_1\bigl(m_{l(k)-1}(0) - m_{l(k)-1}(d_{l(k)-1})\bigr).$$

Combine Lemma 3.8 with the above inequality, we get

$$\lim_{k\to\infty} \bigl(m_{l(k)-1}(0) - m_{l(k)-1)}(d_{l(k)-1)})\bigr) = 0. \tag{25}$$

According to Assumption 2.1 and (12), we have

$$\begin{aligned} m_{l(k)-1}(0) - m_{l(k)-1}(d_{l(k)-1}) &\ge \tau\|g_{l(k)-1}\|\min\left\{\Delta_{l(k)-1}, \frac{\|g_{l(k)-1}\|}{\|B_{l(k)-1}\|}\right\} \\ &\ge \tau\|g_{l(k)-1}\|\min\left\{\|d_{l(k)-1}\|, \frac{\|d_{l(k)-1}\|}{\kappa M_1}\right\} \\ &\ge \frac{\tau}{\kappa}\min\left\{1, \frac{1}{\kappa M_1}\right\}\|d_{l(k)-1}\|^2 \\ &= \omega\|d_{l(k)-1}\|^2 \ge 0, \end{aligned}$$

where $\omega = \frac{\tau}{\kappa}\min\{1, \frac{1}{\kappa M_1}\}$. It follows from (25) that

$$\lim_{k\to\infty} \|d_{l(k)-1}\| = 0 \tag{26}$$

The reminder of the proof is similar to a theorem of [1] and here is omitted. □

On the basis of the above lemmas and analysis, we can obtain the global convergence result of Algorithm 2.1 as follows:

Theorem 3.1 (Global convergence) *Suppose that Assumption 2.1 holds and the sequence $\{x_k\}$ is generated by Algorithm 2.1. Then we have,*

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

Proof We assume that $\overline{d}_k$ be the solution of subproblem (8) corresponding to $p_k = p$, and we have an infinite sequence $\{x_k\}$ satisfying $\widehat{\rho}_k \ge \mu_1$.

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

According to Lemma 3.2, we have,

$$f_{l(k)} - f(x_k + d_k) \ge \mu_1\bigl(f_{l(k)} - f_k - m_k(d_k)\bigr) \ge \mu_1\frac{\beta_1^p}{2M_1}\|g_k\|^2.$$

This above inequality and Lemma 3.8 indicate that (27) holds. □

We will prove the superlinear convergence of Algorithm 2.1 under suitable conditions.

Theorem 3.2 (Superlinear convergence) *Suppose that Assumption 2.1 holds and Algorithm 2.1 generated the sequence $\{x_k\}$ converges to x^*. Moreover, assume that $\nabla^2 f(x^*)$ is positive definite matrix and $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of x^*. If $\|d_k\| \leq \Delta_k$, where $d_k = -B_k^{-1} g_k$, and*

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x^*)d_k\|}{\|d_k\|} = 0. \tag{28}$$

Then the sequence $\{x_k\}$ converges to x^* superlinearly, that is,

$$\|x_{k+1} - x^*\| = o\bigl(\|x_k - x^*\|\bigr). \tag{29}$$

Proof From (28) and $\|d_k\| \leq \Delta_k$, we obtain

$$\lim_{k\to\infty} \frac{\|(\nabla^2 f(x^*) - B_k)d_k\|}{\|d_k\|} = \lim_{k\to\infty} \frac{\|g_k + \nabla^2 f(x^*)d_k\|}{\|d_k\|}. \tag{30}$$

Using Taylor expansion, there exists $t_k \in (0,1)$ such that

$$\begin{aligned} g_{k+1} &= g_k + \nabla^2 f(x_k + t_k d_k)d_k \\ &= g_k + \nabla^2 f\bigl(x^*\bigr)d_k + \bigl(\nabla^2 f(x_k + t_k d_k) - \nabla^2 f\bigl(x^*\bigr)d_k\bigr). \end{aligned}$$

Thus, we can obtain that

$$\frac{\|g_{k+1}\|}{\|d_k\|} \leq \frac{\|g_k + \nabla^2 f(x^*)d_k\|}{\|d_k\|} + \bigl\|\nabla^2 f(x_k + t_k d_k) - \nabla^2 f\bigl(x^*\bigr)\bigr\|.$$

From (28) and $\nabla^2 f(x^*)$ is Lipschitz continuous in a neighborhood of x^*, we get

$$\lim_{k\to\infty} \frac{\|g_{k+1}\|}{\|d_k\|} = 0. \tag{31}$$

Note that by Theorem 3.1, it is implied that

$$g_k \to 0 \quad \text{as } k \to \infty,$$

and thus, we have $d_k \to 0$. We can obtain

$$\lim_{k\to\infty} \|d_k\| = 0, \tag{32}$$

then,

$$g\bigl(x^*\bigr) = \lim_{k\to\infty} g_k = 0. \tag{33}$$

Combine $\nabla^2 f(x^*)$ is a positive definite matrix and (33). Then, there exists a constant $\varsigma > 0$, and $k_0 \geq 0$ such that

$$\|g_{k+1}\| \geq \varsigma \bigl\|x_{k+1} - x^*\bigr\|, \quad \forall k \geq k_0.$$

Thus, we obtain

$$\frac{\|g_{k+1}\|}{\|d_k\|} \geq \varsigma \frac{\|x_{k+1}-x^*\|}{\|d_k\|} \geq \varsigma \frac{\|x_{k+1}-x^*\|}{\|x_{k+1}-x^*\| + \|x_k - x^*\|} \geq \varsigma \frac{1}{1 + \frac{\|x_k - x^*\|}{\|x_{k+1}-x^*\|}}.$$

Combine above inequality with (31), we get $\lim_{k\to\infty} \frac{\|x_k-x^*\|}{\|x_{k+1}-x^*\|} = 0$. So the proof is completed. □

4 Preliminary numerical experiments

In this section, we perform numerical experiments on Algorithm 2.1. A set of unconstrained test problems are selected from [16]. The simulation experiment uses MATLAB 9.4, the processor uses Intel (R) Core (TM), 2.00 GHz, 6 GB RAM. Take exactly the same value for the public parameters of these algorithms: $\mu_1 = 0.25$,

$\mu_2 = 0.75$, $\beta_1 = 0.25$, $\beta_2 = 1.5$, $c_0 = 1$, $N = 5$. The matrix B_k is updated by (9). The stopping criterions are $\|g_k\| \leq 10^{-6}$ and the number of iterations exceeds 5000. We denote the number of gradient evaluations by "n_i", the number of function evaluations by "n_f".

For convenience, we use the following notations to represent the algorithms:

SNTR: Standard nonmonotone trust region method [17].

ATRG: Nonmonotone Shi's adaptive trust region method with $q_k = -g_k$ [18].

ATRN: Nonmonotone Shi's adaptive trust region method with $q_k = -B_k^{-1}g_k$ [18].

NLS: New nonmonotone adaptive trust region line search method.

For standard nonmonotone trust region method, we update Δ_k by the following formula

$$\Delta_{k+1} = \begin{cases} 0.75\Delta_k, & \text{if } \widehat{\rho}_k < \mu_1, \\ \Delta_k, & \text{if } \mu_1 \leq \widehat{\rho}_k < \mu_2, \\ 1.5\Delta_k, & \text{if } \widehat{\rho}_k \geq \mu_2. \end{cases}$$

Table 1 shows that the experiments were conducted to compare NLS and the standard trust region method with a different initial radius. One knows that an initial radius has a significant influence on the numerical results in the standard trust region methods. Moreover, the total number of iterations and function evaluations of the new algorithm are partly less than the standard nonmonotone trust region method. We also know that NLS outperforms with ATRG, ATRN respect to the total number of function evaluations and the total number of gradient evaluations. The performance profiles given by Dolan and More [19] are used to compare the efficiency of the three algorithms. Figures 1–2 give the performance profiles of the three algorithms for the number of function evaluations, and the number of gradient evaluations, respectively. As the figures show that Algorithm 2.1 grows up faster than the other algorithms. Therefore, we can deduce that the new algorithm is more efficient and robust than the other considered trust region algorithms for solving unconstrained optimization.

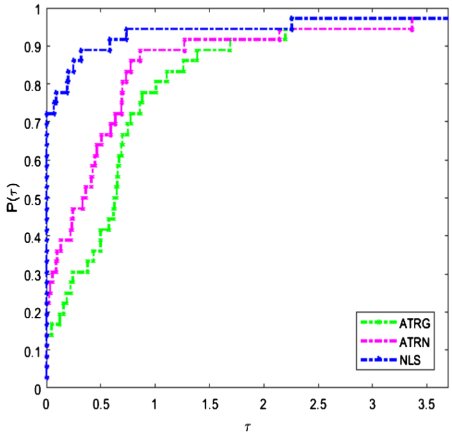
5 Conclusions

In this paper, a new nonmonotone adaptive trust region line search method is presented for unconstrained optimization problems. A new nonmonotone trust region ratio is introduced to enhance the effective of the algorithm. A new trust region radius is proposed, which relaxes the condition of accepting a trial step for the trust region methods. Theorem 3.1 and Theorem 3.2 have been shown that the proposed algorithm can preserve

Table 1 Comparison between adaptive trust region methods and a new method.

Problem	n	n_f/n_i					
		SNTR $\Delta_0 = 0.1$	SNTR $\Delta_0 = 10$	SNTR $\Delta_0 = 100$	ATRG	ATRN	NLS
Ext. Rose	4	690/353	475/243	364/185	168/88	94/65	70/57
Ext. Beale	4	504/254	27/14	29/15	41/21	27/14	19/18
Penalty i	2	129/67	38/21	34/19	33/18	47/26	29/19
Pert. Quad	6	151/80	33/17	31/16	41/21	31/16	29/1
Raydan 1	10	1445/762	40/21	40/21	40/21	40/21	22/15
Raydan 2	4	251/128	14/8	14/8	13/8	13/8	11/6
Diagonal 1	4	112/58	21/11	21/11	21/11	21/11	13/12
Diagonal 2	2	289/147	18/10	18/10	16/9	16/9	19/10
Diagonal 3	10	128/66	33/17	33/17	43/22	33/17	28/15
Hager	10	134/69	29/15	29/15	27/14	27/14	31/16
Gen. Trid 1	20	1234/618	49/25	45/23	50/26	51/26	37/23
Ext. Trid 1	20	65/35	16/9	33/17	16/9	16/9	24/16
Ext. TET	50	195/103	17/9	17/9	16/9	17/9	17/9
Diadonal 4	50	798/429	29/15	19/10	7/4	7/4	6/5
Ext. Him	50	134/69	20/11	25/13	58/44	28/15	18/10
Gen. White	50	666/363	271/151	719/384	382/200	312/186	243/142
Ext. Powell	4	892/455	654/329	–	264/133	237/125	157/98
Full. He. FH3	10	106/55	13/7	11/6	13/7	13/7	8/7
Ext. BD1	100	278/143	33/19	31/17	41/23	268/151	26/19
Pert. Quad	200	290/165	141/72	104/53	188/96	99/51	41/38
Ext. Hie	16	1821/1000	–	–	240/143	198/137	119/60
Quad. QF1	4	683/377	17/9	17/9	17/9	15/8	11/10
FLET34	50	24/13	167/101	225/127	183/110	168/98	210/108
ARWHEAD	200	172/91	44/25	56/30	55/42	21/14	24/13
NONDIA	200	75/38	102/52	71/36	116/61	63/32	63/32
DQDRTIC	200	363/191	50/27	54/29	43/23	36/25	27/23
EG2	10	458/234	25/14	25/14	24/14	24/14	14/14
Bro. Trid.	200	2797/1504	1609/827	356/187	404/216	268/176	258/133
A. Per. Quad	16	253/133	43/22	45/23	61/34	41/31	43/32
Pert. Trid	20	252/135	59/30	57/29	79/47	57/42	56/43
LIARWHD	50	114/60	–	295,148	257/176	255/167	249/132
Ext. DENSCH	100	71/37	34/18	51/26	239/154	50/26	239/154
HIMMELH	4	58/31	91/61	90/56	19/18	75/41	17/16
ENGVAL1	10	165/90	71/38	71/37	65/35	63/34	50/36
EDENSCH	100	1265/633	27/15	3073/1583	21/12	23/13	35/18

Figure 1 Performance profile for the number of function evaluations (n_f)



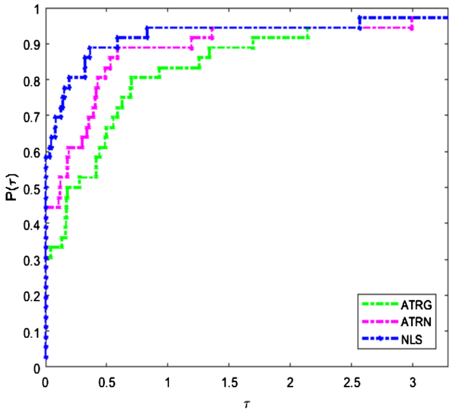


Figure 2 Performance profile for the number of function evaluations (n_i)

global convergence and superlinear convergence, respectively. Numerical experiments have been done on a set of unconstrained optimization test problems of [16]. They showed practical efficiency of the proposed algorithm.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

The main idea of this paper was proposed by WXY and DXF. QQ prepared the manuscript initially and performed all the steps of the proofs in this research. All authors read and approved the final manuscript.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 7 November 2019 Accepted: 27 March 2020 Published online: 10 April 2020

References

1. Grippo L, Lamparillo F, Lucidi S. A nonmonotone line search technique for Newton's method. SIAM J Numer Anal. 1986;23:707–16.
2. Nocedal J, Yuan Y. Combining trust region and line search techniques. In: Yuan Y, editor. Advances in nonlinear programming. Dordrecht: Kluwer Academic; 1996. p. 153–75.
3. Ahookhoosh M, Amini K, Peyghami M. A nonmonotone trust region line search method for large scale unconstrained optimization. Appl Math Model. 2012;36(1):478–87.
4. Sartenaer A. Automatic determination of an initial trust region in nonlinear programming. SIAM J Sci Comput. 1997;18(6):1788–803.
5. Zhang XS, Zhang JL, Liao LZ. An adaptive trust region method and its convergence. Sci China Ser A, Math. 2002;45(1):620–31.
6. Li D. A trust region method with automatic determination of the trust region radius. Chin J Eng Math. 2006;23(5):843–8.
7. Shi ZJ, Wang HQ. A new selfadaptive trust region method for unconstrained optimization. Technical report. College of Operations Research and Management, Qufu Normal University; 2004.
8. Shi ZJ, Guo JH. A new trust region method for unconstrained optimization. J Comput Appl Math. 2008;213(1):509–20.
9. Kimiaei M. A new class of nonmonotone adaptive trust-region methods for nonlinear equations with box constraints. Calcolo. 2017;3:769–812.
10. Amini K, Shiker MAK, Kimiaei M. A line search trust-region algorithm with nonmonotone adaptive radius for a system of nonlinear equations. 4OR. 2016;4(2):132–52.
11. Shan-min P, Lan-ping C. A new family of nonmonotone trust region algorithm. Math Pract Theory. 2011;2011(10):211–8.
12. Reza Peyghami M, Ataee Tarzanagh D. A relaxed nonmonotone adaptive trust region method for solving unconstrained optimization problems. Comput Optim Appl. 2015;61:321–41.
13. Zhou Q, Hang D. Nonmonotone adaptive trust region method with line search based on new diagonal updating. Appl Numer Math. 2015;91:75–88.
14. Wang XY, Ding XF, Qu Q. A new filter nonmonotone adaptive trust region method for unconstrained optimization. Symmetry. 2020;12(2):208.
15. Sang Z, Sun Q. A self-adaptive trust region method with line search based on a simple subproblem model. J Comput Appl Math. 2009;232(2):514–22.
16. Andrei N. An unconstrained optimization test functions collection. Environ Sci Technol. 2008;10:6552–8.

17. Gu NZ, Mo JT. Incorporating nonmonotone strategies into the trust region for unconstrained optimization. Comput Math Appl. 2008;55:2158–72.
18. Ahookhosh M, Amini K. A nonmonotone trust region method with adaptive radius for unconstrained optimization. Comput Math Appl. 2010;60:411–22.
19. Dolan ED, More JJ. Benchmarking optimization software with performance profiles. Math Program. 2002;91:201–13.