

Research Article

A Nonmonotone Adaptive Trust Region Method Based on Conic Model for Unconstrained Optimization

Zhaocheng Cui

Department of Mathematics and Physics, Shandong Jiaotong University, Ji'nan, Shandong Province 250023, China

Correspondence should be addressed to Zhaocheng Cui; cui Zhaocheng@sina.com

Received 20 August 2013; Accepted 29 November 2013; Published 27 January 2014

Academic Editor: Adil M. Bagirov

Copyright © 2014 Zhaocheng Cui. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We propose a nonmonotone adaptive trust region method for unconstrained optimization problems which combines a conic model and a new update rule for adjusting the trust region radius. Unlike the traditional adaptive trust region methods, the subproblem of the new method is the conic minimization subproblem. Moreover, at each iteration, we use the last and the current iterative information to define a suitable initial trust region radius. The global and superlinear convergence properties of the proposed method are established under reasonable conditions. Numerical results show that the new method is efficient and attractive for unconstrained optimization problems.

1. Introduction

In this paper, we consider the following unconstrained optimization problem:

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

where $f: R^n \rightarrow R$ is a continuously differentiable function.

Trust region method is effective for solving (1). In 1970s, Powell [1] established the convergence result of trust region method. Yuan [2], Nocedal, and Yuan [3] proposed various trust region methods for optimization problems. It calculates a trial step by solving the subproblem

$$\min \phi_k(d) = g_k^T d + \frac{1}{2} d^T B_k d, \quad \text{s.t. } \|d\| \leq \Delta_k, \quad (2)$$

where $g_k = \nabla_x f(x_k)$ is the gradient of the objective function at x_k , $B_k \in R^{n \times n}$ is a symmetric matrix which is either the Hessian matrix of f at x_k or an approximation to it, $\|\cdot\|$ refers to the Euclidean norm, and $\Delta_k > 0$ is the radius of an area, called the trust region, where the model is trusted. Let d_k be the solution of (2). The actual reduction of the objective function is defined by

$$\text{Ared}_k = f_k - f(x_k + d_k), \quad (3)$$

and the predicted reduction of $f(x)$ is defined by

$$\text{Pred}_k = \phi_k(0) - \phi_k(d_k). \quad (4)$$

The ratio between these two reductions is defined by

$$r_k = \frac{\text{Ared}_k}{\text{Pred}_k} = \frac{f_k - f(x_k + d_k)}{\phi_k(0) - \phi_k(d_k)}, \quad (5)$$

and it is normally used to test whether the trial step d_k is accepted or the trust region radius needs to be adjusted. The next iterate x_{k+1} is chosen by the following formula:

$$x_{k+1} = \begin{cases} x_k + d_k & \text{if } r_k \geq \eta_0, \\ x_k & \text{otherwise,} \end{cases} \quad (6)$$

where $\eta_0 \in [0, 1)$ is a constant. The next trust region radius is chosen as

$$\Delta_{k+1} = \begin{cases} c_0 \Delta_k & \text{if } r_k < \eta_1, \\ c_1 \Delta_k & \text{if } r_k > \eta_2, \\ \Delta_k & \text{otherwise,} \end{cases} \quad (7)$$

where $0 \leq \eta_1 < \eta_2 < 1$, $0 < c_0 < 1 < c_1$ are constants. In comparison with quasi-Newton methods, trust region methods converge to a point which not only is a stationary

point but also satisfies second-order necessary conditions. Trust region method has strong convergence and robustness, so many authors have studied it; see [4–7]. Conn et al. [8], Nocedal and Wright [9], and Yuan and Sun [10] presented the complete introduction.

Recently, Grippo et al. [11] presented the nonmonotone line search method for unconstrained optimization problems. Due to its high efficiency of nonmonotone method, this technique has been combined with trust region methods; we can see [12–16].

Sartenaer [17] presented a strategy for determining automatically an initial trust region radius. Fan and Yuan [18] proposed a trust region method with the trust region radius converging to zero. Recently, Zhang et al. [19] gave an adaptive trust region method, and they solve the subproblem (2) with $\Delta_k = c^p \|g_k\| \|\hat{B}_k^{-1}\|$, $0 < c < 1$, and p is a nonnegative integer. \hat{B}_k is a safely positive definite matrix based on Schnabel and Eskow modified Cholesky factorization. Therefore, instead of adjusting Δ_k , one adjusts p at each iteration. Based on [19] and a simple subproblem model, Sang and Sun [20] proposed another adaptive trust region method with line search. The numerical results show that the adaptive trust region algorithms [21–24] are more effective than traditional trust region methods for unconstrained optimization problems.

The adaptive trust region methods listed above are based on quadratic model; however, when the objective function has strong nonquadratic behavior, the quadratic model methods often produce a poor prediction of the minimizer of the function. In 1980, Davidon [25] proposed the conic model methods for unconstrained optimization problems. A typical conic trust region subproblem is as follows:

$$\begin{aligned} \min \varphi_k(s) &= \frac{g_k^T s}{1 - \alpha_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1 - \alpha_k^T s)^2}, \\ \text{s.t. } 1 - \alpha_k^T s &> 0, \\ \|s\| &\leq \Delta_k, \end{aligned} \quad (8)$$

where $\varphi_k(s)$ is called conic model which is an approximation to $f(x_k + s_k) - f(x_k)$. The vector α_k is the associated vector for the collinear scaling in the k th iteration, and it is normally called the horizontal vector. If $\alpha_k = 0$, the conic model reduces to a quadratic model. Also, one can see that $\varphi_k(s)$ is quadratic along any direction $s \in R^n$ satisfying $\alpha_k^T s = 0$. Recently, trust region methods based on conic model have been presented and studied; see [26–32]. Numerical results show the efficiency of the conic trust region methods.

In the traditional trust region methods, the next trust region radius is updated by the current ratio r_k , which is reasonable if the matrix B_k is exactly as the Hessian H_k and if the trust region subproblem (2) is solved exactly. However, in practical computations, the matrix B_k is often obtained approximately and the subproblem is solved approximately. In such a case, it may be more reasonable to adjust the next trust region Δ_{k+1} according to not only r_k but also the previous ratios $\{r_{k-q}, \dots, r_k\}$, where q is some nonnegative integer.

In order to measure the agreement between the model function and the objective function, we define the following ratio:

$$\hat{r}_k = \sum_{i=0}^{\min\{k,q\}} \omega_{ki} r_{k-i}, \quad (9)$$

where $\omega_{ki} \in [0, 1]$ is the weight of r_{k-i} , such that

$$\sum_{i=0}^{\min\{k,q\}} \omega_{ki} = 1. \quad (10)$$

In this paper, we propose a new adaptive trust region method which combines the conic model and nonmonotone technique which was given in [33] and a new update rule for adjusting the trust region radius. Our method is different from other adaptive trust region methods in three points. First, the subproblem is conic model which can produce a better prediction of the minimizer of the function than quadratic model methods when the objective function has strong nonquadratic behavior. Second, at each iteration, our method generates a suitable trust region radius automatically based on the current and last iterative information. Last, we use a new update rule (8) to adjust the next trust region radius. The new trust region model is more consistent with the objective function at the current iterative point. According to the nonmonotone technique in [33], our method constructs an iterative series $\{x_k\}$ such that the sequence $\{f(x_k)\}$ is not monotonic, which can accelerate the convergence rate of the minimization process, especially in the narrow curved valley. The global and superlinear convergence properties of the proposed method are proved under reasonable conditions. The numerical results show that the new method is more effective than other trust region methods.

The rest of this paper is organized as follows. In next section, we describe a new nonmonotone adaptive trust region method based on conic model for unconstrained optimization problems. In Sections 3 and 4, we prove the global and superlinear convergence properties of the proposed method under reasonable conditions, respectively. Some numerical results are given in Section 5. Conclusions are summarized in Section 6.

2. Algorithm Description

In this section, we describe a new nonmonotone adaptive trust region method based on conic model for unconstrained optimization problems.

We obtain the trial step s_k by solving the following subproblem:

$$\begin{aligned} \min \varphi_k(s) &= \frac{g_k^T s}{1 - \alpha_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1 - \alpha_k^T s)^2}, \\ \text{s.t. } 1 - \alpha_k^T s &> 0, \\ \|s\| &\leq \Delta_k, \end{aligned} \quad (11)$$

where $\Delta_k = \mu_k (\|s_{k-1}\|^2 / s_{k-1}^T \hat{B}_k s_{k-1}) \|g_k\|$, $\Delta_0 = \mu_0 \|g_0\|$, $\mu_0 > 0$. μ_k is updated according to (8), s_{k-1} is the solution of the last

subproblem. \hat{B}_k is generated by the procedure $s_{k-1}^T \hat{B}_k s_{k-1} = s_{k-1}^T B_k s_{k-1} + i \|s_{k-1}\|^2$ and i is the smallest nonnegative integer such that

$$s_{k-1}^T \hat{B}_k s_{k-1} = s_{k-1}^T B_k s_{k-1} + i \|s_{k-1}\|^2 > 0. \quad (12)$$

The new reduction of $f(x)$ is defined by

$$\text{Ared}_k = M_k - f(x_k + s_k), \quad (13)$$

where

$$M_k = \max \left\{ f(x_k), \sum_{r=0}^{m(k)-1} \lambda_{kr} f(x_{k-r}) \right\}, \quad (14)$$

$\lambda \in (0, 1]$, $m \geq 1$ is a positive integer. We define $m(k) = \min\{k+1, m\}$ and choose

$$\lambda_{kr} \geq \lambda, \quad r = 0, 1, \dots, m(k)-1, \quad \sum_{r=0}^{m(k)-1} \lambda_{kr} = 1. \quad (15)$$

The predicted reduction of $f(x)$ is

$$\text{Pred}_k = -\frac{g_k^T s_k}{1 - \alpha_k^T s_k} - \frac{1}{2} \frac{s_k^T B_k s_k}{(1 - \alpha_k^T s_k)^2}. \quad (16)$$

The new ratio r_k is defined by

$$r_k = \frac{\text{Ared}_k}{\text{Pred}_k}. \quad (17)$$

In what follows, we describe our new nonmonotone adaptive trust region method based on conic model.

Algorithm 1 (nonmonotone adaptive trust region method based on conic model). Consider the following.

Step 0. Given $x_0 \in R^n$, $B_0 \in R^{n \times n}$ is a symmetric positive definite matrix, $\varepsilon > 0$, $\mu_0 > 0$, $\Delta_0 = \mu_0 \|g_0\|$, $0 < \lambda \leq 1$, $0 < \eta_0 < \eta_1 < 1$, $0 < c_0 < 1 < c_1$, and an integer constant $m \geq 1$; set $m(0) = 0$, $k = 0$.

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

Step 2. Solve the subproblem (11) and let s_k be an approximate solution of the subproblem (11).

Step 3. Compute Ared_k , Pred_k , and r_k .

Step 4. If $r_k < \eta_0$, set $\mu_k := c_0 \mu_k$ and go to Step 2. Otherwise, go to Step 5.

Step 5. Choose $\omega_{ki} \in [0, 1]$ that satisfying (9). Compute \hat{r}_k and, by (8), set $x_{k+1} = x_k + s_k$ and

$$\mu_{k+1} = \begin{cases} c_1 \mu_k & \text{if } \eta_1 \leq \hat{r}_k, \\ \mu_k & \text{if } \eta_0 \leq \hat{r}_k < \eta_1. \end{cases} \quad (18)$$

Step 6. Update α_{k+1} , $m(k+1)$, and the symmetric matrix B_{k+1} . Let $\Delta_{k+1} = \mu_{k+1} (\|d_k\|^2 / d_k^T \hat{B}_{k+1} d_k) \|g_{k+1}\|$ and set $k = k+1$ and go to Step 1.

Remark 2. In Algorithm 1, we define a new reduction of $f(x)$ which is different from the actual reduction $f_k - f(x_k + s_k)$ in other methods. Instead of requiring f_{k+1} to be smaller than f_k , it is only required that f_{k+1} is either less than f_k or less than the weighted mean of the function values at the last $m(k)$ iterates.

Remark 3. How to choose the horizontal vector α_k and the approximate Hessian matrix B_k is one of the crucial issues of a conic model method. More details for generating α_k and B_k can be found in [25]. If $\alpha_k = 0$, Algorithm 1 reduces to the quadratic model trust region method. If we choose $m = 1$, we obtain a monotonic adaptive trust region method. Hence, Algorithm 1 is a generalization and development of quadratic model trust region method and monotonic trust region method.

Remark 4. “Step 2-Step 3-Step 4” are called the internal circulation.

3. Convergence Analysis

In this section we discuss the global convergence of Algorithm 1. Before we address some theoretical issues, we would like to give the following assumptions.

Assumption 5. (i) The level set $L(x_0) = \{x \in R^n \mid f(x) \leq f(x_0)\}$ is bounded for any given $x_0 \in R^n$ and $f(x)$ is twice continuously differentiable in $L(x_0)$. (ii) The sequences $\{B_k\}$ and $\{\alpha_k\}$ are uniformly bounded; that is, there exist two positive scalars M_1, M_2 such that $\|B_k\| \leq M_1$ and $\|\alpha_k\| \leq M_2$ hold for all k . (iii) There exists a positive constant $\sigma \in (0, 1)$, such that $\|\alpha_k\| \Delta_k \leq \sigma$.

It is similar to Theorem 3.1 in [28], and we can prove the following lemma.

Lemma 6. Suppose that Assumption 5 holds. Then, there exists a scalar $\beta \in (0, 1)$ such that

$$\begin{aligned} \text{Pred}_k &\geq \beta \|g_k\| \min \left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\} \\ &\geq \beta \|g_k\| \min \left\{ \|s_k\|, \frac{\|g_k\|}{\|B_k\|} \right\} \end{aligned} \quad (19)$$

holds for all k , where s_k is an inexact solution of the subproblem (11).

Lemma 7. Suppose that Assumption 5 holds. Then,

$$|f(x_k) - f(x_k + s_k) - \text{Pred}_k| \leq O(\|s_k\|^2), \quad (20)$$

where s_k is the solution of (11) and is sufficiently close to zero.

Proof. Since $\{\|\alpha_k\|\}$ is bounded, we can have $1/(1 - \alpha_k^T s_k) = 1 + O(\|s_k\|)$. Then, from the boundedness of $\{g_k\}$ and $\{B_k\}$, we obtain

$$\begin{aligned} \frac{g_k^T s_k}{1 - \alpha_k^T s_k} &= g_k^T s_k + O(\|s_k\|^2), \\ \frac{s_k^T B_k s_k}{(1 - \alpha_k^T s_k)^2} &= s_k^T B_k s_k + O(\|s_k\|^2). \end{aligned} \quad (21)$$

By $f(x)$ which is twice continuously differentiable in $L(x_0)$, there exists $M_3 > 0$ such that $\|H(x)\| \leq M_3$ holds for all Hessian $H(x)$ and all $x \in L(x_0)$.

Therefore, we have that

$$\begin{aligned} &|f(x_k) - f(x_k + s_k) - \text{Pred}_k| \\ &= \left| f(x_k) - f(x_k + s_k) + \frac{g_k^T s_k}{1 - \alpha_k^T s_k} + \frac{1}{2} \frac{s_k^T B_k s_k}{(1 - \alpha_k^T s_k)^2} \right| \\ &= \left| -g_k^T s_k - \frac{1}{2} s_k^T H(x_k + \theta_k s_k) s_k \right. \\ &\quad \left. + g_k^T s_k + \frac{1}{2} s_k^T B_k s_k + O(\|s_k\|^2) \right| \\ &\leq \frac{1}{2} (M_1 + M_3) \|s_k\|^2 + O(\|s_k\|^2) = O(\|s_k\|^2). \end{aligned} \quad (22)$$

This completes the proof. \square

Lemma 8. Suppose that Assumption 5 holds. Then, Algorithm 1 is well defined; that is, Algorithm 1 can not cycle infinitely in the internal circulation.

Proof. Suppose that Algorithm 1 cycles infinitely many times between Steps 2 and 4 at the current point x_k . We denote the solution of the conic model (11) by $s_{k(i)}$ and the corresponding predicted reduction by $\text{Pred}_{k(i)}$ with the cycling index i . Then, we have

$$r_{k(i)} < \eta_0, \quad i = 0, 1, 2, \dots \quad (23)$$

$$\Delta_{k(i)} = c_0^i \Delta_k \longrightarrow 0, \quad i \longrightarrow \infty. \quad (24)$$

From Lemmas 6 and 7, we can have

$$\begin{aligned} &\left| \frac{f_k - f(x_k + s_{k(i)})}{\text{Pred}_{k(i)}} - 1 \right| \\ &= \left| \frac{f_k - f(x_k + s_{k(i)}) - \text{Pred}_{k(i)}}{\text{Pred}_{k(i)}} \right| \\ &\leq \frac{O(\|s_{k(i)}\|^2)}{\text{Pred}_{k(i)}} \\ &\leq \frac{O(\Delta_{k(i)}^2)}{\beta \|g_k\| \min\{\Delta_{k(i)}, \|g_k\| / \|B_k\|\}} \longrightarrow 0, \\ &\quad \text{as } i \longrightarrow \infty. \end{aligned} \quad (25)$$

Hence, the following

$$\frac{f_k - f(x_k + s_{k(i)})}{\text{Pred}_{k(i)}} \geq \eta_0 \quad (26)$$

holds for i sufficiently large. From $M_k \geq f_k$, we obtain that

$$r_{k(i)} = \frac{M_k - f(x_k + s_{k(i)})}{\text{Pred}_{k(i)}} \geq \eta_0, \quad (27)$$

holds for i sufficiently large, which contradicts with (23). The proof is completed. \square

In the next lemma, an important decrease property of the function value $f(x_k)$ is established.

Lemma 9. Suppose that Assumption 5 holds and $\{x_k\}$ is generated by Algorithm 1. Then, there exists a constant $\tau > 0$ such that the following

$$\begin{aligned} f(x_k) &\leq f(x_0) - \tau \lambda \sum_{r=0}^{k-2} \|g_r\| \min \left\{ \Delta_r, \frac{\|g_r\|}{\|B_r\|} \right\} \\ &\quad - \tau \|g_{k-1}\| \min \left\{ \Delta_{k-1}, \frac{\|g_{k-1}\|}{\|B_{k-1}\|} \right\} \end{aligned} \quad (28)$$

holds for $k \geq 1$.

Proof. We prove it by induction. For simplicity, we denote

$$\beta_k = \|g_k\| \min \left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\}. \quad (29)$$

(1) If $k = 1$, by $r_0 \geq \eta_0$, $M_0 = f_0$, and Lemma 6, we have

$$\begin{aligned} f(x_1) &\leq f(x_0) - \eta_0 (\varphi_0(0) - \varphi_0(s_0)) \\ &\leq f(x_0) - \eta_0 \beta \|g_0\| \min \left\{ \Delta_0, \frac{\|g_0\|}{\|B_0\|} \right\} \\ &= f(x_0) - \tau \|g_0\| \min \left\{ \Delta_0, \frac{\|g_0\|}{\|B_0\|} \right\}, \end{aligned} \quad (30)$$

where $\tau = \eta_0 \beta$.

(2) Now, we assume that it holds for $1, 2, \dots, k$, and we consider two cases. \square

Case 1. If $M_k = f(x_k)$, by $r_{k+1} \geq \eta_0$, $0 < \lambda \leq 1$, and Lemma 6, then we have

$$\begin{aligned} f(x_{k+1}) &= f(x_k + s_k) \leq f(x_k) - \eta_0 (\varphi_k(0) - \varphi_k(s_k)) \\ &\leq f(x_0) - \tau \lambda \sum_{r=0}^{k-2} \beta_r - \tau \beta_{k-1} - \eta_0 \beta \beta_k \\ &\leq f(x_0) - \tau \lambda \sum_{r=0}^{k-1} \beta_r - \tau \beta_k. \end{aligned} \quad (31)$$

Case 2. $M_k = \sum_{r=0}^{m(k)-1} \lambda_{kr} f(x_{k-r})$. Let $q = \min[k, m-1]$, by $r_{k+1} \geq \eta_0$, $0 < \lambda \leq 1$, and Lemma 6. Then, we have

$$\begin{aligned} f(x_{k+1}) &= f(x_k + s_k) \\ &\leq \sum_{p=0}^q \lambda_{kp} f(x_{k-p}) - \eta_0 (\varphi_k(0) - \varphi_k(s_k)) \\ &\leq \sum_{p=0}^q \lambda_{kp} \left(f(x_0) - \tau \lambda \sum_{r=0}^{k-p-2} \beta_r - \tau \beta_{k-p-1} \right) \\ &\quad - \eta_0 (\varphi_k(0) - \varphi_k(s_k)). \end{aligned} \quad (32)$$

Using $(0, 1, 2, \dots, q) \times (0, 1, 2, \dots, k-p-2) \subset \{(p, r); 0 \leq p \leq q, 0 \leq r \leq k-p-2\}$, $\lambda_{kp} \geq \lambda$, $\sum_{p=0}^q \lambda_{kp} = 1$, we have

$$\begin{aligned} f(x_{k+1}) &\leq f(x_0) - \tau \lambda \sum_{r=0}^{k-q-2} \left(\sum_{p=0}^q \lambda_{kp} \right) \beta_r - \tau \sum_{p=0}^q \lambda_{kp} \beta_{k-p-1} - \tau \beta_k \\ &\leq f(x_0) - \tau \lambda \sum_{r=0}^{k-q-2} \beta_r - \tau \lambda \sum_{r=k-q-1}^{k-1} \beta_r - \tau \beta_k \\ &\leq f(x_0) - \tau \lambda \sum_{r=0}^{k-1} \beta_r - \tau \beta_k. \end{aligned} \quad (33)$$

Lemma 10. Suppose that the conditions in Lemma 9 hold. Then, $\{x_k\} \subset L(x_0)$.

Proof. By Lemma 9, $f(x_k) \leq f(x_0) - \tau \lambda \sum_{r=0}^{k-1} \|g_r\| \min\{\Delta_r, \|g_r\|/\|B_r\|\} \leq f(x_0)$, the conclusion holds obviously.

Now, we prove the global convergence of Algorithm 1. \square

Theorem 11. Suppose that Assumption 5 holds. If $\varepsilon = 0$, Algorithm 1 either terminates in finite iterations or generates an infinite sequence $\{x_k\}$ which satisfies

$$\liminf_{k \rightarrow \infty} \|g_k\| = 0. \quad (34)$$

Proof. If Algorithm 1 terminates in finite iterations, the theorem is obviously true. Assume that Algorithm 1 generates an infinite sequence $\{x_k\}$ in the following proof.

If (34) is not true, then there exists a positive constant ε_0 , such that

$$\|g_k\| \geq \varepsilon_0, \quad \forall k. \quad (35)$$

From Lemma 9, $0 < \lambda \leq 1$, and $\|g_k\| \geq \varepsilon_0$, we can have

$$f(x_k) \leq f(x_0) - \tau \lambda \varepsilon_0 \sum_{r=0}^{k-1} \min \left\{ \Delta_r, \frac{\varepsilon_0}{\|B_r\|} \right\}. \quad (36)$$

By Assumption 5(ii), we can obtain

$$\tau \lambda \varepsilon_0 \sum_{r=0}^{k-1} \min \left\{ \Delta_r, \frac{\varepsilon_0}{M_1} \right\} \leq f(x_0) - f(x_k). \quad (37)$$

Therefore, by Assumption 5(i), we have

$$\sum_{k=0}^{\infty} \Delta_k < \infty. \quad (38)$$

Thus, we can obtain

$$\lim_{k \rightarrow \infty} \Delta_k = 0. \quad (39)$$

By Assumption 5(ii), then $\|\widehat{B}_k\| \leq 2M_1 + 1$. From Algorithm 1, we know that

$$\begin{aligned} \Delta_k &= \mu_k \frac{\|s_{k-1}\|^2}{s_{k-1}^T \widehat{B}_k s_{k-1}} \|g_k\| \\ &\geq \mu_k \frac{\|s_{k-1}\|^2}{(2M_1 + 1) \|s_{k-1}\|^2} \varepsilon_0 = \frac{\mu_k \varepsilon_0}{2M_1 + 1}. \end{aligned} \quad (40)$$

By (39) and (40), we obtain

$$\lim_{k \rightarrow \infty} \mu_k = 0. \quad (41)$$

On the other hand, we have from Lemma 6 that

$$\begin{aligned} &\left| \frac{f_k - f(x_k + s_k)}{\text{Pred}_k} - 1 \right| \\ &= \frac{O(\|s_k\|) + O(\|s_k\|^2 \|B_k\|)}{\text{Pred}_k} \\ &\leq \frac{O(\|s_k\|) + O(\|s_k\|^2 \|B_k\|)}{\beta \|g_k\| \min\{\Delta_k, \|g_k\|/\|B_k\|\}} \\ &\leq \frac{O(\|s_k\|)}{\Delta_k} \rightarrow 0. \end{aligned} \quad (42)$$

The above inequality implies that

$$r_k = \frac{M_k - f(x_k + s_k)}{\text{Pred}_k} \geq \frac{f_k - f(x_k + s_k)}{\text{Pred}_k} \geq \eta_0 \quad (43)$$

holds for k sufficiently large. Hence, there exists a positive constant μ^* such that

$$\mu_k \geq \mu^*. \quad (44)$$

holds for all sufficiently large k , which contradicts with (41). The proof is completed. \square

4. Superlinear Convergence

In general, the superlinear convergence property of Algorithm 1 requires further assumptions, we give the assumptions and the superlinear convergence property in the following theorem.

Theorem 12. Suppose that Assumption 5 holds and Algorithm 1 generates an infinite sequence $\{x_k\}$ which converges to

x^* , where $H(x^*)$ is positive definite and $H(x)$ is Lipschitz continuous in a neighborhood of x^* . If the following condition holds

$$\lim_{k \rightarrow \infty} \frac{\|g_k + H(x^*) s_k\|}{\|s_k\|} = 0, \quad (45)$$

then the rate of convergence is superlinear; that is, $\|x_{k+1} - x^*\| = o(\|x_k - x^*\|)$.

Proof. From $x_k \rightarrow x^*$, we can have $\lim_{k \rightarrow \infty} \|s_k\| = 0$. By the Taylor expansion, we can obtain

$$\begin{aligned} g_{k+1} &= g_k + \nabla^2 f(x_k + \theta_k s_k) s_k \\ &= g_k + H(x^*) s_k + (\nabla^2 f(x_k + \theta_k s_k) - H(x^*)) s_k, \end{aligned} \quad (46)$$

where $\theta_k \in (0, 1)$.

So,

$$\begin{aligned} \|g_{k+1}\| &\leq \|g_k + H(x^*) s_k\| + \|\nabla^2 f(x_k + \theta_k s_k) - H(x^*)\| \cdot \|s_k\|. \end{aligned} \quad (47)$$

Dividing both sides by $\|s_k\|$, we get

$$\frac{\|g_{k+1}\|}{\|s_k\|} \leq \frac{\|g_k + H(x^*) s_k\|}{\|s_k\|} + \|\nabla^2 f(x_k + \theta_k s_k) - H(x^*)\|. \quad (48)$$

By (45), $\|s_k\| \rightarrow 0$, and Lipschitz continuous property of $H(x)$, we can have

$$\lim_{k \rightarrow \infty} \frac{\|g_{k+1}\|}{\|s_k\|} = 0. \quad (49)$$

By $H(x^*)$ which is positive definite, and f which is a twice continuously differentiable function, there exists $\beta > 0$, such that

$$\beta \|x_{k+1} - x^*\| \leq \|g_{k+1}\| \quad (50)$$

for sufficiently large k .

By (49) and (50), it follows that

$$\begin{aligned} \frac{\|g_{k+1}\|}{\|s_k\|} &\geq \frac{\beta \|x_{k+1} - x^*\|}{\|s_k\|} \geq \frac{\beta \|x_{k+1} - x^*\|}{\|x_{k+1} - x_k\|} \\ &\geq \frac{\beta \|x_{k+1} - x^*\|}{\|x_{k+1} - x^*\| + \|x_k - x^*\|} \\ &= \beta \frac{\|x_{k+1} - x^*\| / \|x_k - x^*\|}{1 + (\|x_{k+1} - x^*\| / \|x_k - x^*\|)}, \end{aligned} \quad (51)$$

and thus

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0, \quad (52)$$

which implies that the sequence $\{x_k\}$ converges to x^* super-linear. \square

5. Numerical Results

In this section, we compare the performance of Algorithm 1, denoted by NACTR, with NAQTR if $\alpha_k = 0$ for all k in Algorithm 1, MACTR if $m = 1$ in Algorithm 1, CTR method which is a conic trust region algorithm without adaptive technique, and QTR method which is the traditional trust region method without adaptive technique. All programs are written in MATLAB with double precision.

For these methods, the trial step s_k is computed approximately by the algorithm proposed by Lu and Ni in [34] for solving the subproblem (11). B_0 is chosen as the identity matrix. The parameters in our algorithm are chosen as $\eta_0 = 0.25$, $\eta_1 = 0.7$, $c_0 = 0.25$, $c_1 = 1.5$, $\lambda_{kr} = 1/m(k)$, ($r = 0, 1, \dots, m(k) - 1$), $m = 4$, and $\alpha_0 = 0$. In our implementation, we prefer the following choice

$$\hat{r}_k = 0.85r_k + 0.15\hat{r}_{k-1}, \quad (53)$$

where $\hat{r}_0 = r_0$.

The stopping condition is

$$\|g_k\| \leq 10^{-4}. \quad (54)$$

The iteration is also terminated if the number of evaluations exceeds 300.

We provide the results of our tests in Table 1. The numerical results are given in the form of N_i, N_f , where N_i, N_f denote the numbers of iterations and function evaluations, respectively. We chose the test problems from [35]. From Table 1, we can see that, in most of the problems considered, the numbers N_i and N_f for the conic model methods are considerably smaller than those required for the quadratic model methods, especially for those functions with strong nonquadratic behavior. This means that the conic model methods are very effective to many unconstrained optimization. Compared with the methods without adaptive technique, the adaptive trust region methods are more attractive. Hence, our new nonmonotone adaptive conic trust region method is an improvement of the existing trust region methods.

6. Conclusions

In this paper, we propose a new nonmonotone adaptive conic model trust region method for unconstrained optimization. The global and superlinear convergence properties of the new method are proved under reasonable conditions. Our method is attractive in the following aspects. First, the subproblem is conic model which can produce a better prediction of the minimizer of the function than quadratic model methods

TABLE I: Numerical comparisons.

Problems	QTR N_i/N_f	CTR N_i/N_f	NAQTR N_i/N_f	MACTR N_i/N_f	NACTR N_i/N_f
Helical Valley	62/74	43/53	56/61	32/45	29/42
Gaussian	7/12	3/8	3/5	3/5	3/5
Box 3-D	41/45	33/37	36/41	30/35	30/33
Var. Dimen.	17/26	11/11	18/27	12/18	13/21
Waston	72/78	66/73	62/81	54/98	48/86
Penalty I	18/21	16/17	18/24	15/20	15/18
Penalty II	21/26	21/24	23/31	17/26	16/25
Brown-Dennis	62/66	78/84	62/75	55/67	53/58
Gulf R.D.	88/83	96/98	66/46	38/42	46/47
Trigonometric	46/52	38/39	45/48	32/39	34/36
Ex. Ros.	91/124	Failed	48/62	33/46	35/48

when the objective function has strong nonquadratic behavior. Second, at each iteration, our method generates a suitable trust region radius automatically based on the current and last iterative information. Last, we use a new update rule (8) to adjust the next trust region radius. According to the nonmonotone technique in [33], our method constructs an iterative series $\{x_k\}$ such that the sequence $\{f(x_k)\}$ is not monotonic, which can accelerate the convergence rate of the minimization process, especially in the narrow curved valley. Numerical results show that the new method is efficient for unconstrained optimization problems.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author is thankful to the editor and the reviewers for their constructive comments, which greatly improved this paper. This work was supported by the National Science Foundation of China (Grant no. 11226229) and Foundation of Shandong Jiaotong University.

References

- [1] M. J. D. Powell, "Convergence properties of a class of minimization algorithms," in *Nonlinear Programming*, J. B. Rosen, O. L. Mangasarian, and K. Ritter, Eds., pp. 1–27, Academic press, New York, NY, USA, 1975.
- [2] Y. Yuan, "On the convergence of trust region algorithms," *Mathematica Numerica Sinica*, vol. 16, pp. 333–346, 1996.
- [3] J. Nocedal and Y. Yuan, "Combining trust region and line search techniques," *Advances in Nonlinear Programming*, Y. Yuan, Ed., pp. 153–175, Kluwer Academic, Dordrecht, The Netherlands, 1996.
- [4] M. J. D. Powell, "A new algorithm for unconstrained optimization," in *Nonlinear Programming*, J. B. Rosen, O. L. Mangasarian, and K. Ritter, Eds., pp. 31–65, Academic press, New York, NY, USA, 1970.
- [5] M. J. D. Powell, "On the global convergence of trust region algorithms for unconstrained optimization," *Mathematical Programming*, vol. 29, no. 3, pp. 297–303, 1984.
- [6] T. Steihaug, "The conjugate gradient method and trust region in large scale optimization," *SIAM Journal on Numerical Analysis*, vol. 20, pp. 626–637, 1983.
- [7] E. M. Gertz, "A quasi-Newton trust-region method," *Mathematical Programming A*, pp. 1–24, 1963.
- [8] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust-Region Methods*, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2000.
- [9] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, NY, USA, 1999.
- [10] Y. X. Yuan and W. Y. Sun, *Optimization Theory and Methods*, Science Press, Beijing, China, 1997.
- [11] L. Grippo, F. Lampariello, and S. Lucidi, "A nonmonotone line search technique for Newton's method," *SIAM Journal on Numerical Analysis*, vol. 23, no. 4, pp. 707–716, 1986.
- [12] N. Y. Deng, Y. Xiao, and F. J. Zhou, "Nonmonotonic trust region algorithm," *Journal of Optimization Theory and Applications*, vol. 76, no. 2, pp. 259–285, 1993.
- [13] W. Y. Sun, "Nonmonotone trust region method for solving optimization problems," *Applied Mathematics and Computation*, vol. 156, no. 1, pp. 159–174, 2004.
- [14] P. L. Toint, "Non-monotone trust-region algorithms for nonlinear optimization subject to convex constraints," *Mathematical Programming B*, vol. 77, no. 1, pp. 69–94, 1997.
- [15] J. T. Mo, C. Y. Liu, and S. C. Yan, "A nonmonotone trust region method based on nonincreasing technique of weighted average of the successive function values," *Journal of Computational and Applied Mathematics*, vol. 209, no. 1, pp. 97–108, 2007.
- [16] X. Ke and J. Han, "A class of nonmonotone trust region algorithms for unconstrained optimization problems," *Science in China A*, vol. 41, no. 9, pp. 927–932, 1998.
- [17] A. Sartenaer, "Automatic determination of an initial trust region in nonlinear programming," *SIAM Journal on Scientific Computing*, vol. 18, no. 6, pp. 1788–1803, 1997.
- [18] J. Y. Fan and Y. X. Yuan, "A new trust region algorithm with trust region radius converging to zero," in *Proceedings of the 5th International Conference on Optimization: Techniques and Applications*, Hong Kong, China, December 2001.

- [19] X. S. Zhang, J. L. Zhang, and L. Z. Liao, "An adaptive trust region method and its convergence," *Science in China A*, vol. 45, no. 5, pp. 620–631, 2002.
- [20] Z. Y. Sang and Q. Y. Sun, "A self-adaptive trust region method with line search based on a simple subproblem model," *Journal of Computational and Applied Mathematics*, vol. 232, no. 2, pp. 514–522, 2009.
- [21] L. Hei, "A self-adaptive trust region algorithm," *Journal of Computational Mathematics*, vol. 21, pp. 229–236, 2003.
- [22] G. D. Li, "A trust region method with automatic determination of the trust region radius," *Chinese Journal of Engineering Mathematicss*, vol. 23, pp. 843–848, 2006.
- [23] Z. J. Shi and J. H. Guo, "A new trust region method for unconstrained optimization," *Applied Mathematics and Computation*, vol. 213, pp. 509–520, 2008.
- [24] J. L. Zhang and X. S. Zhang, "A nonmonotone adapyive trust region method and its convergence," *Computers and Mathematics with Application*, vol. 45, pp. 1469–1477, 2003.
- [25] W. C. Davidson, "Conic approximation and collinear scaling for optimizers," *SIAM Journal on Numerical Analysis*, vol. 17, pp. 268–281, 1980.
- [26] S. Di and W. Y. Sun, "A trust region method for conic model to solve unconstrained optimization," *Optimization Methods and Software*, vol. 6, no. 4, pp. 237–263, 1996.
- [27] Q. Ni, "Optimality conditions for trust-region subproblems involving a conic model," *SIAM Journal on Optimization*, vol. 15, no. 3, pp. 826–837, 2005.
- [28] S.-J. Qu, K.-C. Zhang, and J. Zhang, "A nonmonotone trust-region method of conic model for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 220, no. 1-2, pp. 119–128, 2008.
- [29] W. Sun, "On nonquadratic model optimization methods," *Asia and Pacific Journal of Operational Research*, vol. 13, pp. 43–63, 1996.
- [30] H. Gourgeon and J. Nocedal, "A Conic algorithm for optimization," *SIAM Journal of Scientific Stochastic Computation*, vol. 6, pp. 253–267, 1985.
- [31] W. Y. Sun and Y. X. Yuan, "A conic trust-region method for nonlinearly constrained optimization," *Annal of Operational Research*, vol. 103, pp. 175–191, 2001.
- [32] S. Sheng, "Interpolation by conic model for unconstrained optimization," *Computing*, vol. 54, pp. 83–98, 1995.
- [33] Z. S. Yu and D. G. Pu, "A new nonmonotone line search technique for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 219, no. 1, pp. 134–144, 2008.
- [34] X. P. Lu and Q. Ni, "A quasi-Newton trust region method with a new conic model for the unconstrained optimization," *Applied Mathematics and Computation*, vol. 204, no. 1, pp. 373–384, 2008.
- [35] J. J. Moré, B. S. Grabow, and K. E. Hillstrom, "Testing unconstrained optimization software," *ACM Transactions on Mathematical Software*, vol. 7, pp. 17–41, 1981.

