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

Jian Zhang,† Kecun Zhang, Shaojian Qu
Faculty of Science, Xi’an Jiaotong University, P.O.Box 2325
Xi’an 710049, P.R. China

Abstract

In this paper, we present a nonmonotone adaptive trust region method for unconstrained optimization based on conic model. The new method combines nonmonotone technique and a new way to determine trust region radius at each iteration. The local and global convergence properties are proved under reasonable assumptions. Numerical experiments show that our algorithm is effective.

Key words: unconstrained optimization trust region method conic model nonmonotone technique adaptive.

1 Introduction

In this paper, the following unconstrained optimization is considered

$$\min_{x \in \mathbb{R}^n} f(x) \tag{1.1}$$

where $f(x)$ is a twice continuously differentiable function.

Trust region methods of quadratic model for unconstrained optimization have been studied by many researchers [1,2,3,4,5]. They are robust, can be applied to ill-conditioned problems and have strong global convergence properties. Another advantage of trust region methods is that there is no need to require the approximate Hessian of the trust region subproblem to be positive definite. For problem (1.1), Nocedal and Yuan [6] showed

---

*This work is supported by the National Natural Science Foundation of China Grant (Grant Number: 10671057).
†Email address: jianzhang1985@gmail.com
that a trust region trial step is always a descent direction for any approximate Hessian. It is well known that for line search methods one generally has to assume the approximate Hessian to be positive definite in order to ensure that the search direction is a descent direction.

In stead of quadratic approximation to objective function, David [7] proposed conic model to approximate the objective function. Di and Sun [8] first presented a trust region method based on conic model for unconstrained optimization where the trust region subproblem has the form:

\[
\begin{align*}
\min \varphi_k(s) &= \frac{g_k^T s}{1-\alpha_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1-\alpha_k^T s)^2}, \\
\text{s.t.} \quad \|D_k s\| &\leq \Delta_k,
\end{align*}
\]

Sun et al. [9,10] generalized the trust region method of conic model for unconstrained optimization to solve linearly constrained optimization and nonlinear equality constrained optimization problem. They also established the global convergence of these methods. In this paper, we choose \(D_k = I\), thus the conic trust region subproblem can be reduced to:

\[
\begin{align*}
\min \varphi_k(s) &= \frac{g_k^T s}{1-\alpha_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1-\alpha_k^T s)^2}, \\
\text{s.t.} \quad \|s\| &\leq \Delta_k,
\end{align*}
\] (1.2)

where \(\varphi_k(s)\) is called conic model which is an approximation to \(f(x_k + s) - f(x_k)\), \(g_k\) is the gradient of \(f(x)\) at \(x_k\). \(B_k\) is an approximate Hessian of \(f(x)\) at \(x_k\) and \(\Delta_k\) is the trust region radius. The vector \(\alpha_k\) is the associated vector for the colinear 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. Therefore the conic model methods are the generalization of the quadratic model methods. They have several advantages. First, if the objective function has strong non-quadratic behavior or its curvature changes severely, the quadratic model methods often produce a poor prediction of the minimizer of the function. In this case, conic model approximates the objective function better than a quadratic approximation, because it has more freedom in the model. Second, the quadratic model does not take into account the information concerning the function value in the previous iteration which is useful for algorithms. However, the conic model possesses richer interpolation information and satisfies four interpolation conditions of the function values and the gradient values at the current and the previous points. Using these rich interpolation information may improve the performance of the algorithms. Third, the initial and limited numerical results provided in [8,11] etc. show that the conic model methods give improvement over the quadratic model ones. Finally, the conic model methods have the similar global and local convergence properties as the quadratic model ones.
Furthermore it is known that the objective function sequences generated by these algorithms are monotonically decreasing: i.e., \( f(x_k) \geq f(x_{k+1}) \), \( k = 0, 1, \cdots \).

Recently, nonmonotone line search techniques have been studied by many authors since Grippo et al.\[12\]. Many authors generalized the nonmonotone technique to trust region methods and proposed nonmonotone trust region methods \[13,14,15,16\]. Theorectic analysis and numerical results show that the algorithms with nonmonotone properties are more efficient than usual monotone algorithms. To our knowledge, the nonmontone trust region methods listed above are mostly based on quadratic model. Recently, Qu et al. \[17\] proposed nonmonotone trust region methods based on conic model.

In trust region methods, how to choose a suitable trust region radius is very crucial to the effect of the algorithm. In fact, if the trust region radius is too large, the approximate minimizer of the model is likely to be a poor indicator of an improved iterate for the objective function. On the other hand, too small a trust region may lead to slow improvement in the estimate of the solution. In practice, people make their own choice of trust region radius based on their knowledge to given problems. Recently, Sartenaer \[18\] presented a strategy for automatically determining an initial trust region radius. The basic idea is to determine an initial radius through many repeated trials in the direction \(-g_0\) in order to guarantee a sufficient agreement between the model and the objective function. Here \(-g_0\) denotes the minus gradient direction at initial point \(x_0\). Zhang et al. \[19,20\] proposed some other strategies for determining the trust region radius. Fu and Sun \[21\] presented nonmonotone adaptive trust region for unconstrained optimization problems, but all the methods above are based on quadratic model. Recently, Fu et al. \[22\] presented an adaptive trust region algorithm based on conic model. Han et al. \[23\] constructed a variant conic model by choosing an adaptive horizontal vector. They also reported numerical experiments of their adaptive conic trust region algorithm.

In this paper, we combine the subproblem (1.2) with nonmonotone technique to propose a nonmonotone adaptive trust region method based on conic model. We give a simple way to determine the trust region radius at each iterate point. The local and global convergence properties of algorithm are proved under some reasonable assumptions. Finally, we conduct numerical experiments. The results show the efficiency of the new algorithm. We also compare our algorithm with \[17\]. Numerical results show that our algorithm is promising.

The rest of this paper is organized as follows. In Section 2, we present the nonmonotone adaptive trust region method based on conic model. In Section 3, the global and local convergence properties are studied. Numerical results in Section 4 indicate that the algorithm is efficient and promising.
2 The algorithm

In this section, we give a nonmonotone trust region algorithm based on conic model. Before giving the algorithm, the following definitions are needed:

\[
l(k) = \arg \max_{0 \leq j \leq m(k)} \{ f_k - j \}, \quad k = 0, 1, 2, \ldots \quad (2.1)
\]

\[
f_{l(k)} = \max_{0 \leq j \leq m(k)} \{ f_k - j \}, \quad k = 0, 1, 2, \ldots \quad (2.2)
\]

where \( m(k) = \min \{ m(k-1) + 1, M \} \), \( m(0) := 0 \), \( M \geq 0 \) is an integer constant. Let \( s_k \) be the solution of the subproblem (1.2). Then either \( x_k + s_k \) is accepted as a new iteration point or the trust region radius is reduced according to a comparison between the actual reduction of the objective function

\[
ared_k(s_k) = f_{l(k)} - f(x_k + s_k) \quad (2.3)
\]

and the reduction predicted by the conic model

\[
pred_k(s_k) = -\frac{g_k^T s_k}{1 - \alpha_k s_k} - \frac{1}{2} \frac{s_k^T B_k s_k}{(1 - \alpha_k s_k)^2}. \quad (2.4)
\]

That is, if the reduction in the objective function is satisfactory, then we finish the current iteration by taking

\[
x_{k+1} = x_k + s_k \quad (2.5)
\]

and adjusting the trust region radius; otherwise the iteration is repeated at point \( x_k \) with a reduced trust region radius. Now we are ready to state our algorithm.

**Algorithm NACTR** (The nonmonotone adaptive conic trust region algorithm for unconstrained optimization)

Step 0: Choose parameters \( 0 < c_0, c < 1 \), \( \gamma > 0 \) and \( \varepsilon \geq 0 \). Give an arbitrary initial point \( x_0 \in \mathbb{R}^n \), \( B_0 \in \mathbb{R}^{n \times n} \), \( a_0 \in \mathbb{R}^n \) and an integer constant \( M \geq 0 \). Set \( k := 0 \), \( m(0) := 0 \), \( p(0) := 0 \).

Step 1: If \( \| g_k \| < \varepsilon \), then stop with \( x_k \) as the approximate optimal solution; otherwise go to Step 2.

Step 2: Set

\[
\Delta_k := c^{p(k)} \| g_k \|^{\gamma} M_k. \quad (2.6)
\]

Solve the conic minimization subproblem (1.2) and let \( s_{p(k)} \) be one solution of the subproblem (1.2).

Step 3: If \( k \geq 1 \), set \( m(k) := \min \{ m(k-1) + 1, M \} \). Compute \( ared_k(s_{p(k)}) \), \( pred_k(s_{p(k)}) \) and

\[
r_k = \frac{ared_k(s_{p(k)})}{pred_k(s_{p(k)})}. \]
If \( r_k \leq c_0 \), then set \( p(k) := p(k) + 1 \) and go to Step 2; otherwise let

\[
\begin{align*}
    s_k &:= s_{p(k)} \\
    x_{k+1} &= x_k + s_k \\
    p(k) &:= 0
\end{align*}
\]

and go to Step 4.

**Step 4:** Generate \( \alpha_{k+1} \) and \( B_{k+1} \). Set \( k := k + 1 \) and go to Step 1.

**Remarks:**

(i) For the trust region based methods, the main computation is spent to solve the trust region subproblem. It is well known that solving the subproblem exactly is expensive. Hence developing approximate methods for the trust region subproblem has been a popular research topic since 1980’s and numerous algorithms have been proposed. The methods for generating \( \alpha_{k+1} \) and \( B_{k+1} \) can be seen, for example, in \([8,24,25]\). The conditions that we assume for proving global convergence are that the matrices \( B_k \) are uniformly bounded and

\[
\forall k, \exists \sigma \in (0, 1) : \| \alpha_k \| \triangle_k \leq \sigma, \quad (2.7)
\]

which ensures that the conic model function \( \varphi_k(s) \) is bounded over the trust region \( \{s \| s \| \leq \triangle_k \} \). We would like to reiterate the fact that our algorithm reduces to a quadratic model based algorithm if \( \alpha_k = 0 \) for all \( k \). Note that, under the smoothness assumptions taken in this paper, the objective function is locally convex quadratic around a local minimizer. It means that choosing \( \alpha_k \approx 0 \) asymptotically is suitable when \( x_k \) is near the minimizer.

(ii) If \( M = 0 \), this algorithm reduces to monotone one.

(iii) If we set \( M_k = 1, \alpha_k = 0 \) for all \( k \), and choose \( M = 0 \), this model reduces to \([26]\). We can also get models in \([19,21]\) if we choose \( M = 0, \gamma = 1 \) and \( M_k = \| B_k^{-1} \| \) when \( B_k \) is positive definite.

(iv) In this algorithm, the procedure of “Step 2-Step 3-Step 2” is named as inner cycle.

### 3 Convergence analysis

In this section, we establish the convergence results of our algorithm given in the previous section. Before we address some theoretical issues, we would like to make the following assumption.

**Assumption 3.1:**
(i) The level set $L(x) = \{ x | f(x) \leq f(x_0) \}$ generated by Algorithm NACTR is contained in a bounded compact set $\Omega$ and $f(x)$ is twice continuously differentiable in $\Omega$ for any given $x_0 \in \mathbb{R}^n$.

(ii) The sequences $\{B_k^{-1}\}$, $\{B_k\}$ and $\{\alpha_k\}$ are all uniformly bounded.

(iii) There exists two positive constant $a$ and $b$ such that $a \leq M_k \leq b$.

Assumption 3.1(ii) implies that there exists a constant $\Lambda > 0$ such that

$$\| B_k \| \leq \Lambda, \quad \| B_k^{-1} \| \leq \Lambda, \quad \| \alpha_k \| \leq \Lambda, \quad \forall k. \quad (3.1)$$

The method for generating $B_k$ guarantees matrices $\{B_k\}$ are positive definite. So they are invertible. From $\| B_k \| \| B_k^{-1} \| \geq 1$, we have that there exists a positive number $\bar{\Lambda}$ such that

$$\| B_k^{-1} \| \geq \bar{\Lambda}, \quad \forall k. \quad (3.2)$$

**Theorem 3.1** Suppose that (2.7) and Assumption 3.1 hold. Then there exists a positive constant $\delta_1$ such that

$$\text{pred}_k(s_k) \geq \delta_1 \| g_k \| \min \{ \Delta_k, \| B_k \| \} \quad (3.3)$$

for all $k$, where $s_k$ is the solution to (1.2).

**Proof.** Firstly, we let

$$s_k(t) = -tg_k,$$

where $t \in [0, \frac{\Delta_k}{\| g_k \|}]$ such that $s_k(t)$ is feasible to (1.2). So, according to the definitions of $s_k$ and $s_k(t)$, we have

$$\varphi_k(0) - \varphi_k(s_k) \geq \max_{t \in [0, \frac{\Delta_k}{\| g_k \|}]} \varphi_k(0) - \varphi_k(s_k(t)). \quad (3.4)$$

From (2.7), it is easy to see that $\alpha_k^T s_k(t) \leq \| \alpha_k \| \Delta_k \leq \sigma$. By using Cauchy-Schwartz inequality, we obtain

$$\varphi_k(0) - \varphi_k(s_k(t)) = t \frac{\| g_k \|^2}{1 - \alpha_k^T s_k(t)} - \frac{t^2 g_k^T B_k g_k}{2 (1 - \alpha_k^T s_k(t))^2} \geq t \frac{\| g_k \|^2}{1 + \sigma} - \frac{t^2 g_k^T B_k g_k}{2 (1 - \sigma)^2} \geq \frac{\| g_k \|^2}{2(1 + \sigma)} (2t - t^2 \frac{1 + \sigma}{(1 - \sigma)^2} \| B_k \|) \quad (3.5)$$

for all $t \in [0, \frac{\Delta_k}{\| g_k \|}]$. We need to estimate the right side of (3.5). For simplicity, we denote $A = \frac{1 + \sigma}{(1 - \sigma)^2} \| B_k \|$ and $h(t) = 2t - t^2 \frac{1 + \sigma}{(1 - \sigma)^2} \| B_k \| = 2t - At^2$. If $\frac{\Delta_k}{\| g_k \|} \geq \frac{1}{A}$, we have

$$\max_{t \in [0, \frac{\Delta_k}{\| g_k \|}]} h(t) = \frac{1}{A} = \min \{ \| g_k \|, \frac{1}{A} \}.$$
If \( \frac{\Delta_k}{\| g_k \|} \leq \frac{1}{A} \), we can obtain
\[
\max_{t \in \left[0, \frac{\Delta_k}{\| g_k \|}\right]} h(t) = 2 \frac{\Delta_k}{\| g_k \|} - \left( \frac{\Delta_k}{\| g_k \|} \right)^2 A \geq \frac{\Delta_k}{\| g_k \|} = \min \left\{ \frac{\Delta_k}{\| g_k \|}, \frac{1}{A} \right\}.
\]

In either case we can obtain
\[
\max_{t \in \left[0, \frac{\Delta_k}{\| g_k \|}\right]} (2t - t^2) \frac{1 + \sigma}{(1 - \sigma)^2} \| B_k \| \geq \min \left\{ \frac{\Delta_k}{\| g_k \|}, \frac{(1 - \sigma)^2}{2(1 + \sigma)^2} \| B_k \| \right\}. \tag{3.6}
\]

From (3.4), (3.5) and (3.6) we get
\[
\text{pred}_k s_k = \varphi_k(0) - \varphi_k(s_k)
\geq \max_{t \in \left[0, \frac{\Delta_k}{\| g_k \|}\right]} \varphi_k(0) - \varphi_k(s_k(t))
\geq \frac{\| g_k \|^2}{2(1 + \sigma)} \max_{t \in \left[0, \frac{\Delta_k}{\| g_k \|}\right]} (2t - t^2) \frac{1 + \sigma}{(1 - \sigma)^2} \| B_k \|
\geq \| g_k \| \min \left\{ \frac{\Delta_k}{2(1 + \sigma)}, \frac{(1 - \sigma)^2}{2(1 + \sigma)^2} \| B_k \| \right\}. \tag{3.7}
\]

Therefore the theorem follows from (3.7) with \( \delta_1 = \min \left\{ \frac{1}{2(1 + \sigma)} : \frac{(1 - \sigma)^2}{2(1 + \sigma)^2} \right\} = \frac{(1 - \sigma)^2}{2(1 + \sigma)^2} \).

**Lemma 3.2** Suppose that (2.7) and Assumption 3.1 hold, then there exists one positive constant \( \delta_2 \) such that
\[
|f_k - f(x_k + s_k) - \text{pred}_k(s_k)| \leq \delta_2 \| s_k \|^2, \quad \forall k. \tag{3.8}
\]

**Proof.** From the definition of \( \text{pred}_k(s_k) \), we have that
\[
|f_k - f(x_k + s_k) - \text{pred}_k(s_k)|
= \left| - g_k^T s_k - \frac{1}{2} s_k^T \nabla^2 f(x_k + \theta_k s_k) s_k + \frac{g_k^T s_k}{1 - \alpha_k^T s_k} + \frac{1}{2} \frac{s_k^T B_k s_k}{1 - \alpha_k^T s_k} \right|
= \left| \frac{(g_k^T s_k)(g_k^T s_k)}{1 - \alpha_k^T s_k} - \frac{1}{2} s_k^T (\nabla^2 f(x_k + \theta_k s_k) - B_k) s_k - \frac{1}{2} s_k^T B_k s_k (1 - \frac{1}{(1 - \alpha_k^T s_k)^2}) \right|
\leq \left\| \frac{\alpha_k}{1 - \sigma} \right\| \left\| g_k \right\| + \frac{1}{2} \left\| \nabla^2 f(x_k + \theta_k s_k) - B_k \right\| + \frac{1}{2} (1 + \frac{1}{(1 - \sigma)^2}) \left\| B_k \right\| \left\| s_k \right\|^2 \tag{3.9}
\]

where \( \theta_k \in [0, 1] \). It follows from (3.9) and Assumption 3.1 that the lemma is true with
\[
\delta_2 \geq \frac{\| g_k \|}{1 - \sigma} + \frac{1}{2} \left\| \nabla^2 f(x_k + \theta_k s_k) - B_k \right\| \left( 1 + \frac{1}{(1 - \sigma)^2} \right) \left\| B_k \right\|. \]

The following theorem guarantees that the NACTR algorithm does not cycle infinitely in the inner cycle.
Theorem 3.3 Suppose that (2.7) and Assumption 3.1 holds. If the process does not terminate at \( x_k \), then we must have \( r_k > c_0 \) after a finite number of inner iterations at most.

Proof. We assume that the algorithm does not terminate at \( x_k \), that is, \( \| g_k \| \neq 0 \). From Theorem 3.1 and Lemma 3.2, we can get

\[
\frac{f(x_k) - f(x_k + s_{p(k)}) - \text{pred}_k(s_{p(k)})}{\text{pred}_k(s_{p(k)})} \leq \frac{\delta_2 \| s_{p(k)} \|^2}{\delta_1 \| g_k \| \min \{ \Delta_{p(k)}, \frac{\| g_k \|}{\| B_k \|} \}} \leq \frac{\delta_2 \| \Delta_{p(k)} \|^2}{\delta_1 \| g_k \| \min \{ \Delta_{p(k)}, \frac{\| g_k \|}{\| B_k \|} \}}. \tag{3.10}
\]

From the definition of \( \Delta_k \), we know that \( \Delta_k \to 0 \) as \( p(k) \to \infty \), so (3.10) implies that

\[
\frac{f(x_k) - f(x_k + s_{p(k)})}{\text{pred}_k(s_{p(k)})} \to 1 \quad \text{as} \quad p(k) \to \infty. \tag{3.11}
\]

From the definition of \( r_k \) we can get

\[
r_k \geq \frac{f(x_k) - f(x_k + s_{p(k)})}{\text{pred}_k(s_{p(k)})} \to 1 > c_0, \tag{3.12}
\]
which means after a finite number of inner iterations at most we must have \( r_k > c_0 \).

In order to prove global convergence of our algorithm, we first give two lemmas below.

Lemma 3.4 Suppose that Assumption 3.1 holds and \( \{ x_k \} \) is generated by the algorithm. Then \( \{ x_k \} \subseteq L(x) \).

Proof. We proof it by induction. If \( k = 0 \), it is evident that \( x_0 \in L(x) \). Assume that \( x_k \in L(x) \) holds. That is to say, \( f(x_j) \leq f(x_0) \) holds for all \( j \leq k \). Since \( f(x_{k+1}) \leq f(l_k) \), \( l(k) \leq k \), and \( f(l_k) \leq f(x_0) \), we can get \( f(x_{k+1}) \leq f(x_0) \). Thus we complete the proof.

Lemma 3.5 Suppose that Assumption 3.1 holds. Then the sequence \( \{ f(l_k) \} \) is not increasing monotonically and convergent.

Proof. From the definition of our algorithm, we have that \( f(x_{k+1}) \leq f(l_k) \). If \( k < M \), by induction, we can prove that \( f(l_k) \leq f(x_0) \); otherwise

\[
f(l_{k+1}) = \max_{0 \leq j \leq m(k+1)} \{ f_{k+1-j} \} \leq \max_{0 \leq j \leq m(k)} \{ f_{k-j} \}, f_{k+1} = f(l_k).
\]

So the sequence \( \{ f(l_k) \} \) is not increasing monotonically. From Assumption 3.1(i) and Lemma 3.4, we know that \( \{ f_k \} \) is bounded. Hence, \( \{ f(l_k) \} \) is convergent.

Now we prove the global convergence of Algorithm NACTR.
**Theorem 3.6** Under the same conditions as Theorem 3.3, assume that \( \{x_k\} \) is an infinite sequence generated by Algorithm NACTR. Then \( \lim_{k \to \infty} \|g_k\| = 0 \).

**Proof.** Suppose that the conclusion is not true, then there must exists a positive constant \( \epsilon_0 \) such that \( \|g_k\| \geq \epsilon_0 \) for all \( k \). On one hand, we obtain that

\[
 f_{l(k)} - f(x_{k+1}) \geq c_0 \text{pred}_k(s_k). \tag{3.13}
\]

Then for all \( k > M \), we can write

\[
 f_{l(l(k)-1)} - f(x_{l(k)}) \geq c_0 \text{pred}_{l(k)-1}(s_{l(k)-1}). \tag{3.14}
\]

By Lemma 3.5, \( \{f_{l(k)}\} \) is not increasing monotonically and convergent. We take limit on both sides of (3.14) and get

\[
 \lim_{k \to \infty} \text{pred}_{l(k)-1}(s_{l(k)-1}) = 0. \tag{3.15}
\]

From Theorem 3.1, (3.1) and Assumption 3.1 (iii) we have

\[
 \text{pred}_{l(k)-1}(s_{l(k)-1}) \geq \delta_1 \|g_{l(k)-1}\| \min \{ \Delta_{l(k)-1}, \|g_{l(k)-1}\|, \|B_{l(k)-1}\| \} \geq \delta_1 \epsilon_0 \min \{ c^{p(l(k)-1)} \epsilon_0^{\gamma} a, \epsilon_0 \}. \tag{3.16}
\]

Combining (3.15) and (3.16) we conclude \( p(l(k) - 1) \to \infty \) as \( k \to \infty \).

On the other hand, from the algorithm we know \( \overline{s}_{l(k)-1} \) corresponding to the following subproblem is unacceptable:

\[
 \begin{cases}
 \min \ \varphi_{l(k)-1}(s) = \frac{g_{l(k)-1}^T s}{1 - \alpha_{l(k)-1}} + \frac{1}{2} s^T B_{l(k)-1} s \\
 \text{s.t.} \quad \| s \| \leq \Delta_{l(k)-1}/c
\end{cases} \tag{3.17}
\]

i.e.,

\[
 \text{ared}_{l(k)-1}(\overline{s}_{l(k)-1}) \leq c_0 \text{pred}_{l(k)-1}(\overline{s}_{l(k)-1}). \tag{3.18}
\]

From Theorem 3.1 and Assumption 3.1 (iii), we obtain

\[
 \text{pred}_{l(k)-1}(\overline{s}_{l(k)-1}) \geq \delta_1 \|g_{l(k)-1}\| \min \{ \Delta_{l(k)-1}/c, \|g_{l(k)-1}\|, \|B_{l(k)-1}\| \} \geq \delta_1 \epsilon_0 \min \{ c^{p(l(k)-1)} \epsilon_0^{\gamma} a/c, \epsilon_0 \}. \tag{3.19}
\]

Assumption 3.1 (i) and Lemma 3.4 imply that there exists a positive number \( c_1 \) such that \( \| g_k \| \leq c_1 \). Thus from Lemma 3.2 and Assumption 3.1(iii) we have

\[
 |f_{l(k)-1} - f(x_{l(k)-1} + \overline{s}_{l(k)-1}) - \text{pred}_{l(k)-1}(\overline{s}_{l(k)-1})| \leq \delta_2 \| \overline{s}_{l(k)-1} \|^2 \leq \delta_2 (\Delta_{l(k)-1}/c)^2 \leq \delta_2 (c^{p(l(k)-1)} c_1^2 b)^2. \tag{3.20}
\]
Since we have $p(l(k) - 1) \to \infty$ as $k \to \infty$, from (3.19) and (3.20) we obtain
\[
|\frac{f_{l(k)-1} - f(x_{l(k)-1} + s_{l(k)-1})}{pred_{l(k)-1}(s_{l(k)-1})} - 1| \leq \frac{\delta_0^2 (\epsilon_0 b)^2}{\delta_1 \epsilon_0 \min\{\epsilon_0 a/c, \epsilon_0 \Lambda\}} \to 0,
\] (3.21)
which means
\[
\frac{f_{l(k)-1} - f(x_{l(k)-1} + s_{l(k)-1})}{pred_{l(k)-1}(s_{l(k)-1})} \to 1.
\] (3.22)

It follows that
\[
\frac{are_{l(k)-1}(s_{l(k)-1})}{pred_{l(k)-1}(s_{l(k)-1})} \geq \frac{f_{l(k)-1} - f(x_{l(k)-1} + s_{l(k)-1})}{pred_{l(k)-1}(s_{l(k)-1})} > c_0
\] (3.23)
for $k$ sufficiently large, which contradicts with (3.18). The proof is completed. ■

By Theorem 3.3, Algorithm NACTR stops in a finite number of iterations under Assumption 3.1 and (2.7). In order to explore the superlinear convergence we give the following assumption.

Assumption 3.2:
(i) The sequence \(\{x_k\}\) generated by Algorithm NACTR converges to a stationary point \(x^*\), i.e.,
\[
\lim_{k \to \infty} x_k = x^* \text{ and } \lim_{k \to \infty} \|g_k\| = \|g^*\| = 0.
\]
(ii) If
\[
\frac{\|B_k^{-1}g_k\|}{1 - g_k^T B_k^{-1} \alpha_k} \leq \Delta_k,
\]
then
\[
s_k = \frac{B_k^{-1}g_k}{1 - g_k^T B_k^{-1} \alpha_k}.
\]

Theorem 3.7 Suppose that Assumption 3.1 and Assumption 3.2 hold. If \(\nabla^2 f(x^*)\) is positive definite and
\[
\lim_{k \to \infty} \frac{\|s_k\|}{\|s_k\|} = 0,
\]
then the sequence \(\{x_k\}\) converges to \(x^*\) superlinearly.

Proof. see [17]. ■

4 Numerical experiments

In this part, we will carry numerical experiments to test our algorithm. All programs are written in Matlab code. Numerical test in PC, CPU Main Frequency 3.08G EMS, 512M run circumstance matlab 6.5. We choose the parameters in algorithm as follows: \(c_0 = 0.1\), \(c_1 = 0.1\), and \(\delta_0 = 0.1\).
The convergence criterion: $\|g_k\| \leq 10^{-9}$ is used for the termination test. We use algorithm 4.1 in [27] to solve the conic subproblem (1.2). Theorem 4.2 in [27] shows that (3.2) holds. The method for generating $B_{k+1}$ and $\alpha_{k+1}$ can be seen, for example in [8,24,25]. We update $B_{k+1}$ in the following way:

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

where

$$y_k = \begin{cases} \frac{g(x_k + s_k)}{g(x_k)} & \text{if } (\frac{g(x_k)}{g(x_k)})^T s_k > 0.2 s_k^T B_k s_k \\ \theta g(x_k + s_k) & \text{otherwise} \end{cases}$$

$$\theta = 0.8 s_k^T B_k s_k / (s_k^T B_k s_k - \frac{g(x_k)}{g(x_k)} s_k).$$

The following four tested problems from [28] are presented:

1. Rosenbrock function
   $$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

2. Brown badly scaled function
   $$f(x) = (x_1 - 10^6)^2 + (x_2 - 2 \cdot 10^6)^2 + (x_1 x_2 - 2)^2.$$

3. Cube function
   $$f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2.$$

4. Freudenstein and Roth function
   $$f(x) = (-13 + x_1 + ((5 - x_2)x_2 - 2)x_2)^2 + (-29 + x_1 + ((x_2 + 1)x_2 - 14)x_2)^2.$$

Algorithm NACTR is used to solve the unconstrained optimization problems (1.1) with objective functions defined as above respectively. We list the results in Table 1. In Table 1, $x_0$, Iter and $x_*$ stand for initial point, iterations and optimal solution of problem (1.1) respectively. Pro represents the problem number. As we can see that these problems are actually the nonlinear least squares problems. Note that, in general, these problems are not easy to be solved by general minimization algorithms since they tend to ignore the structure in these problems. For each problem, the code runs from $M = 0$ to 4, where $M = 0$ means the monotone trust region method of conic model. Therefore, we have actually computed 20 problems and for each problem we have five cases. Analyzing the numerical results, we have the following conclusion: for the four tested problems, our nonmonotone adaptive method is effective.

We also compare our algorithm with NCTR algorithm in [17]. The results can be seen in Table 2. Here we use the same initial point, same algorithm to solve conic subproblem.
and also the way we update $B_{k+1}$. For each algorithm, we carry experiments in three cases. As pointed in Section 1, we use a different way to update the trust region radius. Three adaptive methods are provided in Table 2. Column 2-5 report numerical results experimented by algorithm NACTR for three different rules to determine trust region radius respectively, where $\Delta_{k1} = \epsilon^{p(k)}\|g_k\|$, $\Delta_{k2} = \epsilon^{p(k)}\|g_k\|^2\|B_k\|$ and $\Delta_{k3} = \epsilon^{p(k)}\|B_k\|$. Column 6-8 report the numerical results of various initial trust region radius in algorithm NCTR. All numbers in column 3-8 denote the iteration we take in computation. From Table 2, we can see that our algorithm is efficient, especially for Problem 1, while most are failed when we use NCTR. We can also observe that the results improve slightly for various initial trust region radius. To our algorithm, the results may be improved severely if we choose a suitable way to update the trust region radius. For most cases, our algorithm works better than NCTR.

Acknowledgments

The authors wish to express their sincere thanks to the editor and referee for their constructive comments.

References


