A Simple Alternating Direction Method for the Conic Trust Region Subproblem

Honglan Zhu and Qin Ni

Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
Business School, Huaiyin Institute of Technology, Huaian 223003, China

Correspondence should be addressed to Qin Ni; niqfs@nuaa.edu.cn

Received 23 September 2018; Accepted 27 November 2018; Published 18 December 2018

Academic Editor: Fazal M. Mahomed

A simple alternating direction method is used to solve the conic trust region subproblem of unconstrained optimization. By use of the new method, the subproblem is solved by two steps in a descent direction and its orthogonal direction, the original conic trust domain subproblem into a one-dimensional subproblem and a low-dimensional quadratic model subproblem, both of which are very easy to solve. Then the global convergence of the method under some reasonable conditions is established. Numerical experiments show that the new method seems simple and effective.

1. Introduction

In this paper, we consider the unconstrained optimization problem

\[ \min_{x \in \mathbb{R}^n} f(x), \]  

where \( f(x) \) is continuously differentiable. The trust region method is a very effective method for the unconstrained optimization problem (1) (see [1–6]). Traditional trust region methods are based on a quadratic model and the corresponding quadratic program subproblem is, at the \( k \)th iteration,

\[ \min_{s \in \mathbb{R}^n} q_k(s) = g_k^T s + \frac{1}{2} s^T B_k s, \]  

s.t. \( \|s\| \leq \Delta_k, \)  

where \( x_k \) is the current iterate point, \( g_k = \nabla f(x_k), B_k \) is symmetric and an approximation to the Hessian of \( f(x), \) \( \|\| \) refers to the Euclidean norm, and \( \Delta_k \) is the trust region radius at the \( k \)th iteration. There are many methods that can be used to solve the subproblems (2)-(3). The simple, low cost, and effective methods are dogleg methods (see [7–11]). Now, we recall the simple dogleg algorithm for solving trust region subproblem with the quadratic model as in [12].

Algorithm 1.

Step 0. Input the data of the \( k \)th iteration i.e., \( g_k, B_k \) and \( \Delta_k. \)

Step 1. Compute \( s_k^N = -B_k^{-1} g_k. \) If \( \|s_k^N\| \leq \Delta_k, \) then \( s_* = s_k^N, \) and stop.

Step 2. Compute \( s_k^c = -(g_k^T g_k / g_k^T B_k g_k) g_k. \) If \( \|s_k^c\| \geq \Delta_k, \) then \( s_* = -\Delta_k g_k / \|g_k\|, \) and stop. Otherwise, go to Step 3.

Step 3. Compute

\[ d = \|s_k^N - s_k^c\|^2, \]
\[ e = (s_k^N - s_k^c)^T s_k^c, \]
\[ f = \|s_k^c\|^2 - \Delta_k^2, \]

then \( s_* = s_k^c + \lambda (s_k^N - s_k^c), \) where \( \lambda = (-e + \sqrt{e^2 - df})/d. \)

In 1980, Davidon first proposed the conic model (see [13]). It is an alternative model to substitute the quadratic model. For optimization problems; if the objective function has a strong nonquadratic or its curvature changes severely,
then the conic model is better than the quadratic model both the effect of data fitting and the result of numerical calculations. In addition, the conic model can supply enough freedom to make best use of both information of gradients and function values in iterate points. In view of these good properties of conic model, it has attracted wide attention of many scholars [14–28]. In [28], Ni proposed a new trust region subproblem and gave the optimality conditions for the trust region subproblems of a conic model. That is, at the kth iteration, the trial step $s_k$ is computed by solving the following conic model trust region subproblem

$$\min_{s \in S} \phi_k (s) = \frac{g_k^T s}{1 - a_k^T s} + \frac{s^T B_k s}{2(1 - a_k^T s)^2},$$  \hspace{1cm} (5)$$

$$\text{s.t. } s \in S, \hspace{1cm} (6)$$

where

$$S = \{ s \mid \|s\| \leq \Delta_k, \quad 1 - a_k^T s \geq \varepsilon_0 \}, \hspace{1cm} (7)$$

horizon vector $a_k \in \mathbb{R}^n$, $B_k$ is symmetric and positive semidefinite, and $\varepsilon_0$ ($0 < \varepsilon_0 < 1$) is a sufficiently small positive number. We note that the conic model $\phi_k (s)$ has a denominator and the shape of the trust region $S$ is irregular; therefore the conic trust region subproblems (5)-(7) are not easy to search for the descent point and difficult to solve. The trust region method often does not require the exact solution of trust region subproblem but only requires an approximate solution. The dogleg method for solving the trust region subproblem based on conic model is an approximate solution method; however its calculation is relatively complicated.

In this paper, we continue to study the subproblems (5)-(7). In order to find an easy way to solve, inspired by the alternating direction method of multipliers (ADMM), we consider obtaining the approximate solution by the two-step search in two orthogonal directions in the trust region $S$. ADMM is an algorithm that solves convex optimization problems by breaking them into smaller pieces, each of which are then easier to handle. Because of its significant efficiency and easy implementation, it has recently found wide application in a number of areas (see [29–45]).

In the following, we use the alternating orthogonal direction search method to find the approximate solution of the subproblems (5)-(7). The rest of this paper is organized as follows. In the next section, the motivation and description of the simple alternating direction search method are presented. In Section 3, we give the quasi-Newton method based on the conic model for solving unconstrained optimization problems and prove its global convergence properties. The numerical results are provided in Section 4.

2. Range of $\varepsilon_0$ and the Approximate Solution of the Subproblem

In this section, we will modify the range of $\varepsilon_0$ and give the motivation and description of the algorithm. We note that the conic model $\phi_k (s)$ has one more parameter $a_k$ than $\phi_k (s)$. Therefore, $\phi_k (s)$ can make full use of the existing function information to satisfy more interpolation conditions by using of the function values and the gradient values. All of these will improve the effectiveness of the algorithm. In general, $a_k$ chooses a descent direction, such as $g_{k-1}$, $g_k$, or $s_{k-1}$ (see [13–17]). For convenience, we omit the index $k$ of $a_k$, $g_k$, and $B_k$ in this section. Therefore, in this paper we assume that

$$a^T g < 0$$

and $B$ is positive (abbreviated as $B > 0$).

Let

$$B_{ag} = a^T B a - a^T a a^T g,$$

$$\varepsilon_0 = \min \left\{ 1, \frac{a^T B a}{B_{ag}} \right\}. \hspace{1cm} (10)$$

From (8), we have $a \neq 0$ and

$$B_{ag} > 0. \hspace{1cm} (11)$$

Although in principle we are seeking the optimal solution of the subproblems (5)-(7), it is enough to find an approximate solution $s_k$ in the feasible region and guarantee a sufficient reduction in the model and the global convergence. Therefore, in order to simplify this algorithm we choose $\varepsilon_0$ in (7) such that it satisfies

$$0 < \varepsilon_0 < \varepsilon_0. \hspace{1cm} (12)$$

In the following, we consider the alternating direction search method to solve the subproblems (5)-(7) by making full use of the parameters $a$. The new method is divided into two steps. First, we search in the direction of $a$ and then search in the direction $y$ which is perpendicular to $a$.

Let

$$s = \tau a + y, \hspace{1cm} (13)$$

where $\tau \geq 0, a \neq 0, y \in \mathbb{R}^n$, and $a^T y = 0$. Then, the solving process of subproblems (5)-(7) is divided into the following two stages.

In the first stage, set $y = 0$ and then $s = \tau a$. Substituting it into (5)-(7), we have

$$\min_{\tau \in \mathbb{R}} \rho (\tau) = \frac{\tau a^T g}{1 - \tau a^T a} + \frac{\tau^2 a^T B a}{2 (1 - \tau a^T a)^2}, \hspace{1cm} (14)$$

$$\text{s.t. } \tau \in \Omega, \hspace{1cm} (15)$$

where

$$\Omega = \{ \tau \mid 0 \leq \tau \leq \tau_3 \} \cap \{ \tau \mid \tau \leq \tau_d \text{ or } \tau \geq \tau_u \}, \hspace{1cm} (16)$$

$$\tau_3 = \frac{\Delta}{\|a\|}, \hspace{1cm} (17)$$

$$\tau_d = \frac{1 - \varepsilon_0}{\|a\|^2},$$

$$\tau_u = \frac{1 + \varepsilon_0}{\|a\|^2}. \hspace{1cm} (17)$$
By the direct computation, we have that the derivative of \( \rho(\tau) \) is

\[
\rho'(\tau) = \frac{B_{ag} \tau + a^T g}{\|a\|^2 (\tau - \tau_m)^3},
\]

where

\[
\tau_m = \frac{1}{\|a\|^2}.
\]

From (17), we know that \( 0 < \tau_d < \tau_m < \tau_u \) and then \( \tau_m \notin \Omega \). Since \( B_{ag} > 0 \), then \( \rho(\tau) \) has only one stationary point

\[
\tau_{cp} = \frac{-a^T g}{B_{ag}}.
\]

By simple calculation, the following lemmas can be easily obtained.

**Lemma 2.** Suppose \( a^T g < 0, B > 0 \) and \( 0 < \epsilon_0 < \tau_0 \). Then \( \tau_{cp} < \tau_d \).

**Proof.** Combining with the definition of \( \epsilon_0 \) as in (10), we easily know that

\[
0 < \epsilon_0 < \frac{a^T Ba}{B_{ag}}.
\]

From (17) and (20), we have

\[
\tau_{cp} - \tau_d = \frac{\epsilon_0 - \frac{a^T Ba}{B_{ag}}}{\|a\|^2} < 0.
\]

**Lemma 3.** Under the same conditions as Lemma 2, then \( \rho(\tau) \) is monotonically increasing in the trust region \( (\tau_{cp}, \tau_m) \); \( \rho(\tau) \) is monotonically decreasing for \( 0 < \tau < \tau_{cp} \) and \( \tau > \tau_m \).

**Proof.** From (19) and (20), we know that

\[
\tau_m - \tau_{cp} = \frac{a^T Ba}{B_{ag} \|a\|^2} > 0.
\]

Then combining with (18) we can obtain that the lemma obviously holds.

Let \( \tau_* \) be the solution of the subproblems (14)-(15). Then, we can get the following theorems by analysis.

**Theorem 4.** Under the same conditions as Lemma 2, then the solution of the subproblems (14)-(15) is

\[
\tau_* = \begin{cases} 
\min \{\tau_{cp}, \tau_d\}, & \text{if } \tau_d \leq \tau_u, \\
\tau_{cp}, & \text{if } \tau_d > \tau_u.
\end{cases}
\]

**Proof.** (1) If \( \tau_d \leq \tau_u \) then from (16) we know that \( \Omega = [0, \tau_u] \). From Lemmas 2 and 3, we can obtain that if \( \tau_{cp} \leq \tau_u \) then \( \tau_* = \tau_{cp} \) and if \( \tau_u < \tau_{cp} \) then \( \tau_* = \tau_d \). Therefore, \( \tau_* = \min\{\tau_{cp}, \tau_d\} \).

(2) If \( \tau_d < \tau_u < \tau_m \) then \( \Omega = [0, \tau_d] \). From Lemmas 2 and 3, we can similarly get \( \tau_* = \tau_{cp} \).

(3) If \( \tau_u \geq \tau_m \), then \( \Omega = [0, \tau_u] \cup [\tau_u, \tau_m] \). From Lemmas 2 and 3, we know that

\[
\tau_* = \arg \min \{\rho(\tau_{cp}), \rho(\tau_u)\}.
\]

Because of \( \tau_u \geq \tau_m \), then from (17) we have \( 1 - \Delta \|a\| \leq -\epsilon_0 < 0 \). Since \( a^T g < 0 \), then

\[
\rho(\tau_u) = \frac{\Delta a^T g}{\|a\| (1 - \Delta \|a\|)} - \frac{\Delta^2 a^T B a}{2 \|a\|^2 (1 - \Delta \|a\|)^2} > 0.
\]

Then \( \tau_* = \tau_{cp} \). The theorem is proved.

It is worth noting that if \( \tau_* = \tau_d \) then from (17) we have \( \|a\| \|a\| = \Delta \). Therefore, for this case we set \( \tau_* = \tau_d \) and exit the calculation of subproblem. Otherwise, we know that \( \tau_* a \) is inside the trust region \( \Omega \). Then, we should carry out the calculation of the second stage below.

We set \( s = \tau_* a + y \) and substitute it into \( f_\theta(s) \). Then the subproblems (5)-(7) become

\[
\min_{y \in \mathbb{R}^n} \psi(y)
\]

\[
= g^T (\tau_* a + y) + (\tau_* a + y)^T B (\tau_* a + y),
\]

\[
\text{s.t. } \|y\| \leq \tilde{\Delta},
\]

\[
a^T y = 0,
\]

where

\[
\tilde{\Delta} = \sqrt{\Delta^2 - (\tau_*)^2 \|a\|^2}.
\]

In order to remove the equality constraint in (28), we use the null space technique. That is, for \( a \neq 0 \) then there exist \( n - 1 \) mutually orthogonal unit vectors \( q_1, q_2, \ldots, q_{n-1} \) orthogonal to the parameter vector \( a \). Set \( Q = [q_1, q_2, \ldots, q_{n-1}] \) and \( y = Qu \), where \( u \in \mathbb{R}^{n-1} \). Then (27)-(28) can be simplified as the following subproblem:

\[
\min_{u \in \mathbb{R}^{n-1}} \tilde{\psi}(u) = \tilde{g}^T u + \frac{1}{2} u^T \tilde{B} u,
\]

\[
\text{s.t. } \|u\| \leq \tilde{\Delta},
\]

where

\[
\tilde{g} = \frac{Q^T g}{1 - \tau_* a^T a} + \frac{\tau_* Q^T Ba}{(1 - \tau_* a^T a)^2},
\]

\[
\tilde{B} = \frac{Q^T B Q}{(1 - \tau_* a^T a)^2}.
\]
Set $g_k = \bar{g}$, $B_k = \bar{B}$, and $\Delta_k = \Delta$. By Algorithm 1, we can obtain the solution $u_*$ of the subproblems (30)-(31). Then $y_* = Qu_*$ and $s_* = \tau_* a + y_*$. Thus, the subproblems (5)-(7) are solved approximately. Now we could give the alternating direction search method for solving the conic trust region subproblems (5)-(7) as follows.

**Algorithm 5.** Given $a$, $g$, $B$, and $\Delta$,

1. If $a^T g \geq 0$, then set $s = 0$. Then solve the subproblems (5)-(7) by Algorithm 1 to get $s_k$, and stop.
2. Compute $B_{ag}$ and $\bar{E}$ by (9) and (10). let $\varepsilon_0 = 0.9\varepsilon_0$.
3. Compute $\tau_{\Delta}$, $\tau_d$ and $\tau_c$ by (17) and (20).
4. Compute $\tau$, by (24).
5. If $\tau = \tau_{\Delta}$, then $s_k = \tau a$, and stop; otherwise, compute $Q, \bar{\Delta}, \bar{g}$, and $\bar{B}$ by (29) and (32).
6. Set $g_k = \bar{g}$, $B_k = \bar{B}$ and $\Delta_k = \bar{\Delta}$. Then solve the subproblems (30)-(31) by Algorithm 1 to get $u_*$.
7. Set $y_* = Qu_*$ and $s_* = \tau_* a + y_*$, and stop.

In order to discuss the lower bound of predicted reduction in each iteration, we define the following predicted reduction:

$$
pred(s) = \phi(0) - \phi(s),$$

$$
pred_1(\tau) = \rho(0) - \rho(\tau),$$

$$
pred_2(y) = \psi(0) - \psi(y),$$

$$
pred_3(u) = \bar{\psi}(0) - \bar{\psi}(u)$$

(33) If $s_k$ is obtained by Step 7 of Algorithm 5, then $s_k = \tau a$, where $\tau$ is defined in (24). By computation, we have

$$
pred(s_k) = \rho(\tau)$$

$$
= -\Delta \left[ \frac{a^T (\bar{B} g) + 2 \|a\|^2 \bar{g} a^T g - \Delta \|a\|^2 a^T g}{2 \|a\|^2 (1 - \Delta \|a\|)^2} \right].$$

(37)

From (24), we know that if $\tau = \tau_{\Delta}$, then $\tau \leq \tau_c$ and $\tau \leq \tau_d$. And then from (17) and (20), we can obtain

$$
\Delta \geq 1 - \|a\| \geq \varepsilon_0 > 0.
$$

(39)

Combining with (37)-(39), we know that

$$
pred(s_k) \geq \frac{-\Delta a^T g}{2 \|a\|^2 (1 - \Delta \|a\|)} \geq \frac{\varepsilon \Delta \|g\|}{2},$$

(40)

where

$$
\varepsilon = \frac{-a^T g}{\|a\|^2}.
$$

(41)

For $a^T g < 0$, then $0 < \varepsilon \leq 1$ holds obviously.

(3) If $s_k$ is obtained by Step 7 of Algorithm 5, then $s_k = \tau a + Qu_*$, where $\tau \neq \tau_{\Delta}$. From (24), we know that $\tau_* = \tau_c$. Combining with (33) and (34), we have

$$
pred(s_k) = \rho(\tau_c) + \rho_3(\tau_{\Delta}).$$

(42)

Because $u_*$ is obtained by Algorithm 1, then from [46] we have

$$
pred_3(\tau_{\Delta}) \geq \frac{1}{2} c_2 \|g\| \min \left( \bar{\Delta}, \frac{\|g\|}{\|B\|} \right).$$

(43)

where $c_2 \in (0, 1]$ and $\bar{\Delta}, \bar{g}$, and $\bar{B}$ are defined by (29) and (32). Thus,

$$
pred(s_k) \geq \frac{1}{2} c_2 \|g\| \min \left( \bar{\Delta}, \frac{\|g\|}{\|B\|} \right).$$

(35)

Proof. (1) If $s_k$ is obtained by Step 1 of Algorithm 5, then from Nocedal and Wright [46] we have

$$
pred(s_k) \geq \frac{1}{2} c_1 \|g\| \min \left( \bar{\Delta}, \frac{\|g\|}{\|B\|} \right),$$

(36)

where $c_1 \in (0, 1]$.
3. The Algorithm and Its Convergence

In this section, we propose a quasi-Newton method with a conic model for unconstrained minimization and prove its convergence under some reasonable conditions. In order to solve the problem (1), we approximate \( f(x) \) with a conic model of the form

\[
m_k(s) = f_k + s^T g_k + \frac{1}{2} s^T B_k s,
\]

where \( f_k = f(x_k), g_k = \nabla f(x_k), B_k \in \mathbb{R}^{n \times n}, \) and \( a_k \in \mathbb{R}^n \) are parameter vectors.

Now we give the simple alternating direction trust region algorithm based on conic model (46).

Algorithm 7.

Step 0. Choose parameters \( \epsilon, \epsilon, \in (0, 1), 0 < \eta_1 < \eta_2 < 1, 0 < \delta_1 < 1 < \delta_2, \) and \( \Delta > 0; \) give a starting point \( x_0 \in \mathbb{R}^n, B_0 \in \mathbb{R}^{n \times n}, a_0 \in \mathbb{R}^n, \) and an initial trust region radius \( \Delta_0 \in (0, \Delta); \) set \( k = 0. \)

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

Step 2. Set \( a = a_k, g = g_k, B = B_k, \) and \( \Delta = \Delta_k. \) Then solve the subproblem (5)-(7) by Algorithm 5 to get one of the approximate solution \( s_k. \)

Step 3. Compute the ratio of predicted reduction and the actual reduction

\[
r_k = \frac{\text{ared}(s_k)}{\text{pred}(s_k)},
\]

where

\[
\text{ared}(s_k) = f(x_k) - f(x_k + s_k),
\]

\[
\text{pred}(s_k) = \frac{g_k^T s_k}{1 - a_k^T s_k} - \frac{1}{2} s_k^T B_k s_k.
\]

Step 4. If \( r_k < \eta_1, \) then set \( \Delta_k = \delta_1 \Delta_k \) and go to Step 2. If \( r_k > \eta_1, \) then set \( x_{k+1} = x_k + s_k \) and choose the new trust region bound satisfying

\[
\Delta_k = \begin{cases}
\text{min} \{ \delta_2 \Delta_k, \Delta \}, & \text{if } r_k \geq \eta_2, \|s_k\| = \Delta_k, \\
\Delta_k, & \text{otherwise},
\end{cases}
\]

Step 5. Generate \( a_{k+1} \) and \( B_{k+1}; \) set \( k = k + 1, \) and go to Step 1.

The choice of parameter \( a_{k+1} \) and \( B_{k+1} \) in the cone model method is crucial. In general, \( a_{k+1} \) and \( B_{k+1} \) are chosen to satisfy certain interpolation conditions, which means that the conic model function interpolates both the function values and the gradient values of the objective function at \( x_k \) and \( x_{k+1}. \) The choice of the parameters \( a_{k+1} \) and \( B_{k+1} \) can refer to [13–17] and [47–49], respectively. In this paper, we are not prepared to study the specific iterative formulas of \( a_{k+1} \) and \( B_{k+1} \) in depth and directly adopt the choice of \( a_{k+1} \) in [16] and the choice of \( B_{k+1} \) in [49].

From (49) and Theorem 6, we have

\[
\text{pred}(s_k) \geq \frac{1}{2} c_3 \|g_k\| \min \left\{ \Delta_k, \frac{\|g_k\|}{\|B_k\|} \right\},
\]

where \( c_3 \) as defined by (45). The following theorem guarantees that the Algorithm 7 is globally convergent.

Theorem 8. Under the same conditions as Lemma 2, suppose that the level set

\[
L(x_0) = \{ x | f(x) \leq f(x_0) \}
\]

and the sequence \( \{a_k\}, \{g_k\}, \) and \( \|B_k\| \) is all uniformly bounded, \( B_k \) is symmetric and positive definite, and \( f \) is twice continuously differentiable in \( L(x_0). \) Then for any \( \epsilon > 0, \) Algorithm 7 terminates in finite number of iterations, that is,

\[
\lim_{k \to \infty} \|g_k\| = 0.
\]

Proof. We give the proof by contradiction. Suppose that there is \( \epsilon_1 > 0 \) such that

\[
\|g_k\| \geq \epsilon_1, \quad \forall k.
\]

From the hypothesis, we have

\[
\|a_k\| \leq \bar{a},
\]

\[
\|g_k\| \leq \bar{g},
\]

\[
0 < \|B_k\| \leq \bar{B}.
\]

Combining with (51)-(55), we have

\[
\text{pred}(s_k) \geq \frac{1}{2} c_3 \|g_k\| \min \left\{ \Delta_k, \frac{\epsilon_1}{\bar{B}} \right\} \geq \frac{1}{2} \zeta \Delta_k,
\]

where the first inequality follows from

\[
\min \{p, q\} \geq \frac{pq}{p + q}, \quad \forall p, q > 0,
\]

and the second inequality is from \( \Delta_k \leq \bar{\Delta} \) and

\[
\zeta = \frac{c_3 \epsilon_1}{\epsilon_1 + \bar{B}}.
\]

From Step 4 of Algorithm 7 and (56), we obtain that for all \( k \)

\[
f_k - f_{k+1} \geq \eta_1 \text{pred}(s_k) \geq \frac{1}{2} \eta_1 \zeta \Delta_k.
\]

Since \( f(x) \) is bounded from below and \( f_{k+1} < f_k, \) then we have

\[
\sum_{k} (f_k - f_{k+1}) \geq \sum_{k} \left( \frac{1}{2} \eta_1 \zeta \Delta_k \right),
\]
which implies that
\[ \sum_{k=1}^{\infty} \Delta_k < \infty \]  
(61)

And then
\[ \lim_{k \to \infty} \Delta_k = 0, \]
\[ \lim_{k \to \infty} \|s_k\| = 0. \]  
(62)

On the other hand, from (55) and (62) we can get
\[ \frac{1}{1 - a_k^T s_k} = 1 + a_k^T s_k + o \left( \|s_k\| \right), \]  
(63)

\[ \frac{s_k^T B_k s_k}{2 (1 - a_k^T s_k)_+} = \frac{1}{2} (s_k)^T B_k s_k + o \left( \|s_k\|^2 \right). \]  
(64)

Then from (55) to (63), we have
\[ |f_k - f(x_k + s_k) - \text{pred}(s_k)| = |f_k - f(x_k + s_k) + (1 + a_k^T s_k) g_k s_k + o \left( \|s_k\|^2 \right)| \]
\[ = \frac{1}{2} \eta_k \Delta_k + o \left( \|s_k\|^2 \right) \leq \frac{1}{2} (M_1 + 2a \bar{g} + \bar{B} + O(1)) \|s_k\|^2 \]
\[ \leq \frac{1}{2} (Q + O(1)) \Delta_k^2, \]
where \( \eta_k \in (0, 1) \) and \( Q = M_1 + 2a \bar{g} + \bar{B} \). Combining with (56) and (65), we can get that
\[ |r_k - 1| = \left| \frac{f_k - f(x_k + s_k)}{\text{pred}(s_k)} - 1 \right| \leq \frac{Q + O(1)}{\zeta} \Delta_k. \]  
(66)

From (62) we have \( r_k \to 1 \). Hence, there is a sufficiently large positive number \( K \) such that \( \forall k \geq K \) and
\[ r_k > \eta_1 \]  
(67)

holds. From Step 4 of Algorithm 7, it follows that
\[ \Delta_{k+1} \geq \Delta_k, \quad \forall k \geq K, \]  
(68)

which is a contradiction to (62). The theorem is proved. \( \square \)

4. Numerical Tests

In this section, Algorithm 7 is tested with some standard test problems from [16, 50]. The names of the 16 test problems are listed in Table 1. All the computations are carried out in Matlab R2016b on a microcomputer in double precision arithmetic. These tests use the same stopping criterion \( \|g_k\| \leq 10^{-5} \). The columns in the Tables have the following meanings: No. denotes the numbers of the test problems; \( n \) is the dimension of the test problems; \( \text{Iter} \) is the number of iterations; \( nf \) is the number of function evaluations performed; \( ng \) is the number of gradient evaluations; \( f_k \) is the final objective function value; \( \|g\| \) is the Euclidean norm of the final gradient; CPU(s) denotes the total iteration time of the algorithm in seconds.

The parameters in these algorithms are
\[ \alpha_0 = 0, \]
\[ B_0 = 1, \]
\[ \epsilon_0 = \epsilon = 10^{-5}, \]
\[ \Delta_0 = 1, \]
\[ \lambda = 10, \]
\[ \eta_1 = 0.01, \]
\[ \eta_2 = 0.75, \]
\[ \delta_1 = 0.5, \]
\[ \delta_2 = 2. \]  
(69)

In order to analyze the effectiveness of our new algorithm, we compare Algorithm 7 with the alternating direction trust region method based on conic model (abbreviated as ADCTR) in [12]. The numerical results of ADCTR and Algorithm 7 are listed in Table 2. We note that the optimal value of these test problems is \( f_* = 0 \). From Table 2, we can see that the performance of Algorithm 7 is feasible and effective. For the above 16 problems, Algorithm 7 is better than the ADCTR for 13 tests and is somewhat bad for 4 tests, and the two algorithms are same in efficiency for the other 1 tests. Therefore, it seems that Algorithm 7 is better than algorithm ADCTR in [12].

5. Conclusions

The algorithm ADCTR and Algorithm 7 are similar; that is, the idea of alternating direction method is used to solve the conic trust region subproblem. However, Algorithm 7 in this paper takes into account the special property that the parameter vector \( a \) is generally taken as the descending direction. Thus, under the assumption of \( a^T g < 0 \), the calculation of Algorithm 7 is simpler to calculate and has
Table 2: Numerical results.

<table>
<thead>
<tr>
<th>No.</th>
<th>( n )</th>
<th>Starting point</th>
<th>Algorithm</th>
<th>Iter</th>
<th>( nf/ng )</th>
<th>( f_k )</th>
<th>( | g | )</th>
<th>CPU (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>(2,1.5)</td>
<td>ADCTR</td>
<td>52</td>
<td>53/43</td>
<td>1.0377e-15</td>
<td>1.9477e-06</td>
<td>0.064681</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>56</td>
<td>57/37</td>
<td>7.7243e-13</td>
<td>5.0126e-06</td>
<td>0.049091</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>(1,2)</td>
<td>ADCTR</td>
<td>10</td>
<td>11/11</td>
<td>9.0831e-06</td>
<td>8.9419e-06</td>
<td>0.048493</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>8</td>
<td>9/9</td>
<td>9.0831e-06</td>
<td>8.9416e-06</td>
<td>0.033654</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>(2,-2)</td>
<td>ADCTR</td>
<td>18</td>
<td>19/18</td>
<td>9.0379e-15</td>
<td>9.3925e-07</td>
<td>0.053525</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>16</td>
<td>17/17</td>
<td>7.3180e-16</td>
<td>4.3875e-08</td>
<td>0.039899</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>(-1,1)</td>
<td>ADCTR</td>
<td>16</td>
<td>17/13</td>
<td>1.1407e-12</td>
<td>2.1360e-06</td>
<td>0.050445</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>9</td>
<td>10/10</td>
<td>9.8953e-13</td>
<td>9.8985e-06</td>
<td>0.038502</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>(3,-1,0,1)</td>
<td>ADCTR</td>
<td>41</td>
<td>42/34</td>
<td>4.8648e-09</td>
<td>4.5887e-06</td>
<td>0.062011</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>107</td>
<td>108/83</td>
<td>9.8794e-12</td>
<td>3.9079e-06</td>
<td>0.054692</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>0.25\ast(3,2,1,0)</td>
<td>ADCTR</td>
<td>32</td>
<td>33/29</td>
<td>2.3856e-14</td>
<td>3.0965e-06</td>
<td>0.066287</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>27</td>
<td>28/27</td>
<td>3.6994e-13</td>
<td>5.7059e-06</td>
<td>0.059686</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>(-1,2,1)</td>
<td>ADCTR</td>
<td>50</td>
<td>51/49</td>
<td>1.5486e-14</td>
<td>5.4101e-06</td>
<td>0.064227</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>47</td>
<td>48/40</td>
<td>2.4987e-19</td>
<td>2.1924e-08</td>
<td>0.048847</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>0.25\ast(1,1,1,1)</td>
<td>ADCTR</td>
<td>47</td>
<td>48/34</td>
<td>7.9158e-15</td>
<td>4.1153e-07</td>
<td>0.076865</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>28</td>
<td>29/22</td>
<td>3.0282e-04</td>
<td>8.2932e-06</td>
<td>0.085045</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>1.5\ast(1,1,1,1)</td>
<td>ADCTR</td>
<td>7</td>
<td>8/8</td>
<td>8.1577e-12</td>
<td>4.5905e-06</td>
<td>0.058505</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>6</td>
<td>7/7</td>
<td>5.2383e-13</td>
<td>2.5151e-06</td>
<td>0.054092</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>10\ast(1,1,1,1)</td>
<td>ADCTR</td>
<td>81</td>
<td>82/58</td>
<td>5.8024e-18</td>
<td>4.6604e-07</td>
<td>0.089702</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>56</td>
<td>57/40</td>
<td>7.1131e-18</td>
<td>2.9344e-07</td>
<td>0.077486</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>(-1,-1,-1,-1)</td>
<td>ADCTR</td>
<td>64</td>
<td>65/52</td>
<td>9.9974e-14</td>
<td>4.4986e-06</td>
<td>0.082793</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>51</td>
<td>52/41</td>
<td>2.0655e-13</td>
<td>5.6223e-06</td>
<td>0.060827</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>(-1,-1,-1,-1)</td>
<td>ADCTR</td>
<td>48</td>
<td>49/43</td>
<td>1.1247e-08</td>
<td>5.2578e-06</td>
<td>0.068215</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>128</td>
<td>129/125</td>
<td>4.3892e-08</td>
<td>6.6956e-06</td>
<td>0.054621</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>(-1,-1,-1,-1)</td>
<td>ADCTR</td>
<td>35</td>
<td>36/19</td>
<td>1.4498e-11</td>
<td>5.0442e-06</td>
<td>0.063276</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>38</td>
<td>39/19</td>
<td>1.4483e-11</td>
<td>6.4239e-06</td>
<td>0.047787</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>(0,1,0,1)</td>
<td>ADCTR</td>
<td>91</td>
<td>92/52</td>
<td>0.1998e-06</td>
<td>2.5916e-07</td>
<td>0.089294</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>63</td>
<td>64/41</td>
<td>1.3996e-05</td>
<td>5.7261e-07</td>
<td>0.052852</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>-0.08\ast(2,3,3,2)</td>
<td>ADCTR</td>
<td>23</td>
<td>24/15</td>
<td>2.0042e-12</td>
<td>8.2898e-06</td>
<td>0.061544</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>19</td>
<td>20/12</td>
<td>2.3574e-12</td>
<td>6.3180e-06</td>
<td>0.048328</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>0.25\ast(1,1,1,1)</td>
<td>ADCTR</td>
<td>14</td>
<td>15/15</td>
<td>3.0282e-04</td>
<td>4.9068e-06</td>
<td>0.048488</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Algorithm 7</td>
<td>14</td>
<td>15/11</td>
<td>3.6700e-04</td>
<td>8.1833e-06</td>
<td>0.049998</td>
</tr>
</tbody>
</table>

shorter CPU time, better calculation effect, and also global convergence.

However, there are still many aspects worthy of further study, for example, weakening the positive definite condition of \( B_k \), using algorithms to solve large-scale problems, calculation of convergence rate, and so on.

Data Availability

All data generated or analysed during this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (11071117, 11771210) and the Natural Science Foundation of Jiangsu Province (BK20141409, BK20150420).

References


