A new alternating direction trust region method based on conic model for solving unconstrained optimization

Honglan Zhu[a,b,c], Qin Ni[a,*], Chuangyin Dang[c]

[a]*Department of Mathematics, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, People's Republic of China.*
[b]*Business School, Huaiyin Institute of Technology, Huaian 223003, People's Republic of China.*
[c]*Department of Systems Engineering & Engineering Management, City University of Hong Kong, Kowloon, Hong Kong SAR.*

Abstract

In this paper, a new alternating direction trust region method based on conic model is used to solve unconstrained optimization problems. By use of the alternating direction method, the new conic model trust region subproblem is solved by two steps in two orthogonal directions. This new idea overcomes the shortcomings of conic model subproblem which is difficult to solve. Then the global convergence of the method under some reasonable conditions is established. Numerical experiment shows that this method may be better than the dogleg method to solve the subproblem, especially for large-scale problems.

Keywords: Unconstrained optimization, conic model, trust region method, alternating direction method, global convergence

1. Introduction

In this paper, we consider the unconstrained optimization problem

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

where $f(x)$ is continuously differentiable. The problem (1.1) have been studied by many researchers, including Han [1], Powell [2], Yuan and Sun [3], Powell and Yuan [4], etc. There are many methods to solve problem (1.1), and trust region method is a very effective method (see [4–9]). In addition, the book of Conn, Gould and Toint [10] is an excellent and comprehensive one on trust region methods. Most optimization theory is based on the quadratic model and uses the quadratic model to approximate $f(x)$. That is, at the kth iteration, the following subproblem:

$$\min_{s \in R^n} \varrho_k(s) = g_k^T s + \frac{1}{2} s^T B_k s, \tag{1.2}$$

$$\text{s.t.} \quad \|s\| \le \Delta_k, \tag{1.3}$$

is solved to obtain a search direction s_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 $f(x)$, $\|\cdot\|$ refers to the Euclidean norm, Δ_k is the trust region radius at the kth iteration.

There are many methods can be used to solve the subproblem (1.2)-(1.3). The simple, low cost and effective methods are dogleg methods, such as Powell's single dogleg method [11] and Dennis and Mei's double dogleg method [12]. Then there are other scholars have studied the dogleg method [13–15]. Now, we recall the simple dogleg algorithm for solving trust region subproblem with the quadratic model as following algorithm.

Algorithm 1.1

Step 0. Input the data of the kth iteration i.e., g_k, B_k and Δ_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 $s_k^c = -\frac{g_k^T g_k}{g_k^T B_k g_k} g_k$. If $\|s_k^c\| \geq \Delta_k$, then $s_* = -\frac{\Delta_k g_k}{\|g_k\|}$, and stop. Otherwise, go to Step 3.

Step 3. Compute

$$d = \|s_k^N - s_k^c\|^2, \quad e = (s_k^N - s_k^c)^T s_k^c, \quad f = \|s_k^c\|^2 - \Delta_k^2, \tag{1.4}$$

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

We note that the solution of the subproblem obtained by dogleg methods is only an approximate solution of (1.2)-(1.3). Moreover, practice experience shows that the quadratic model is not always effective. If the objective function possesses high non-linear property and the iterative point is far away from the minimum, the quadratic model could not approximate the original problem very well, which may lead to iteration proceed slowly.

In 1980, Davidon [16] proposed the conic model for solving unconstrained optimization. It is an alternative model to substitute the quadratic model. And it has attracted wide attention of many authors in various areas [17–23]. A typical trust-region subproblem with conic model was first proposed by Di and Sun in [24] as following.

$$\min_{s \in R^n} \phi_k(s) = \frac{g_k^T s}{1 - a_k^T s} + \frac{s^T B_k s}{2(1 - a_k^T s)^2}, \tag{1.5}$$

$$\text{s.t.} \quad \|s\| \leq \Delta_k, 1 - a_k^T s > 0, \tag{1.6}$$

where horizon vector $a_k \in R^n$, and B_k is symmetric and positive semidefinite. In [25], 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

$$\min_{s \in R^n} \phi_k(s) = \frac{g_k^T s}{1 - a_k^T s} + \frac{s^T B_k s}{2(1 - a_k^T s)^2}, \tag{1.7}$$

$$\text{s.t.} \quad \|s\| \leq \Delta_k, \ |1 - a_k^T s| \geq \varepsilon_0, \tag{1.8}$$

where ε_0 ($0 < \varepsilon_0 < 1$) is a sufficiently small positive number. The subproblem (1.7)-(1.8) considered more comprehensive than (1.5)-(1.6), and will not miss the solution of the original problem (1.1).

The research demonstrated that the conic model is superior to quadratic model to some extent, in particular, for those class of objective functions with highly vibrating; 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 this good properties of conic model, we will continue to study it.

It is noteworthy that the simple dogleg algorithm for solving trust region subproblem based on the conic model (DCTR) is similar to the above Algorithm 1.1, where

$$s_k^N = \frac{-B_k^{-1} g_k}{1 - a_k^T B_k^{-1} g_k},$$

$$s_k^c = \frac{-g_k^T g_k}{g_k^T B_k g_k - a_k^T g_k g_k^T g_k} g.$$

However, the calculation of DCTR is much more complicated (see [26–28])

In order to find a simpler method and which is more suitable for the unique structure of the conic model, we considered to using the alternating directions method for solving the conic model subproblem. Alternating directions method (ADM) could date back to [29]. It has been well studied in the linearly constrained convex programming problems. Because of its significant efficiency and easy implementation, ADM has attracted wide attention of many authors in various areas, see [30–35].

In this paper, we combine the subproblem (1.7)-(1.8) with alternating direction search method to propose a new method for solving the conic trust region subproblem. The rest of this paper is organized as follows. In the next section, the motivation and description of the simple alternating direction search algorithm 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 in Section 4 indicate that the algorithm is efficient and robust.

2. A simple alternating direction search method

The conic model $\phi_k(s)$ in the subproblem (1.7)-(1.8) has one more parameter a_k than $\varrho_k(s)$, so $\phi_k(s)$ has more freedom which can take into account the information concerning the function value in the previous iteration which is useful for algorithms. Furthermore, the conic model possesses richer interpolation information and can satisfy 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. Generally, the choice of the parameters a_k is a descent direction, such as $g(x_{k-1})$, $g(x_k)$ or s_{k-1} (see [16–18, 26, 27]).

In view of the unique importance of the parameters a_k, we consider the following alternating direction search method to solve the subproblem (1.7)-(1.8). The new method is divided into two steps. First, we search along the direction parallel to a_k. And then search along the direction y_k which is perpendicular to a_k. For convenience, we omit the index k of a_k, g_k and B_k in this section.

In this paper, we assume that $a \neq 0$ and B is positive (abbreviated as $B > 0$).

Let

$$s = \tau a + y, \tag{2.1}$$

where $\tau \in R, y \in R^n$ and $a^T y = 0$. Then, the solving process of subproblem (1.7)-(1.8) is divided into the following two stages.

In the first stage, we set $y = 0$ and then $s = \tau a$. Substituting it into (1.7)-(1.8), we have

$$\min \quad \rho(\tau) = \frac{\tau a^T g}{1 - \tau a^T a} + \frac{\tau^2 a^T B a}{2(1 - \tau a^T a)^2}, \tag{2.2}$$

$$\text{s.t.} \quad \tau \in \Omega, \tag{2.3}$$

where $\Omega = \{\tau \mid |\tau| \|a\| \leq \Delta, |1 - \tau \|a\|^2|^2 \geq \varepsilon_0\}$.

For the purpose of clarity, we denote

$$\tau_\Delta = \frac{\Delta}{\|a\|}, \quad \tau_d = \frac{1 - \varepsilon_0}{\|a\|^2}, \quad \tau_m = \frac{1}{\|a\|^2}, \quad \tau_u = \frac{1 + \varepsilon_0}{\|a\|^2}. \tag{2.4}$$

Then,

$$\Omega = \{\tau \mid |\tau| \leq \tau_\Delta\} \cap \{\tau \mid \tau \leq \tau_d \text{ or } \tau \geq \tau_u\}. \tag{2.5}$$

In the following, we consider three different cases of (2.2)-(2.3):

(1) If $\Delta \|a\| \leq 1 - \varepsilon_0$, then $\tau_\Delta \leq \tau_d$ and (2.2)-(2.3) becomes

$$(P1) \begin{cases} \min \quad \rho(\tau), \\ \text{s.t.} \quad \tau \in [-\tau_\Delta, \tau_\Delta]. \end{cases} \tag{2.6}$$

(2) If $|1 - \Delta \|a\|| < \varepsilon_0$, then $\tau_d < \tau_\Delta < \tau_u$ and (2.2)-(2.3) becomes

$$(P2) \begin{cases} \min \quad \rho(\tau), \\ \text{s.t.} \quad \tau \in [-\tau_\Delta, \tau_d]. \end{cases} \tag{2.7}$$

(3) If $\Delta \|a\| \geq 1 + \varepsilon_0$, then $\tau_u \leq \tau_\Delta$ and (2.2)-(2.3) becomes

$$(P3) \begin{cases} \min \quad \rho(\tau), \\ \text{s.t.} \quad \tau \in [-\tau_\Delta, \tau_d] \cup [\tau_u, \tau_\Delta]. \end{cases} \tag{2.8}$$

3

Now, we discuss the stationary points of $\rho(\tau)$. By the direct computation, we have that the derivative of $\rho(\tau)$ is

$$\rho'(\tau) = \frac{a_\tau \tau + a^T g}{-\|a\|^6 (\tau - \tau_m)^3}, \tag{2.9}$$

where

$$a_\tau = a^T B a - a^T a a^T g. \tag{2.10}$$

From (2.4), we know that $0 < \tau_d < \tau_m < \tau_u$ and then from (2.5) $\tau_m \notin \Omega$. Therefore, if $a_\tau \neq 0$ then $\rho(\tau)$ has only one stationary point

$$\tau_{cp} = \frac{-a^T g}{a_\tau}. \tag{2.11}$$

Lemma 2.1. *(1) If $a_\tau < 0$ then $\tau_m < \tau_{cp}$ and $\rho(\tau)$ is monotonically decreasing in the in the trust region (τ_m, τ_{cp}); $\rho(\tau)$ is monotonically increasing for $\tau < \tau_m$ and $\tau > \tau_{cp}$.*
(2) If $a_\tau = 0$, then $a^T g > 0$ and $\rho(\tau)$ is monotonically increasing for $\tau < \tau_m$; $\rho(\tau)$ is monotonically decreasing for $\tau > \tau_m$.
(3) If $a_\tau > 0$, then $\tau_{cp} < \tau_m$ and $\rho(\tau)$ is monotonically increasing in the trust region (τ_{cp}, τ_m); $\rho(\tau)$ is monotonically decreasing for $\tau < \tau_{cp}$ and $\tau > \tau_m$.

Proof. From (2.4) and (2.11), we know that if $a_\tau \neq 0$ then

$$\tau_{cp} - \tau_m = \frac{a^T B a}{-a_\tau \|a\|^2}. \tag{2.12}$$

Then, since $B > 0$, combining with (2.9) we can obtain that the lemma obviously holds. □

Theorem 2.1. *If $a^T g = 0$ then the optimal solution of the subproblem (P1), (P2) and (P3) is*

$$\tau_* = 0. \tag{2.13}$$

Proof. If $a^T g = 0$ then from (2.2) we have

$$\rho(\tau) = \frac{\tau^2 a^T B a}{2(1 - \tau a^T a)^2} \geq 0. \tag{2.14}$$

Hence, the theorem holds. □

Theorem 2.2. *If $a^T g \neq 0$, then the optimal solution of the subproblem (P1) is*

$$\tau_* = \begin{cases} -\tau_\Delta, & \text{if } a_\tau \leq 0, \\ \max\{-\tau_\Delta, \tau_{cp}\}, & \text{if } a_\tau > 0, a^T g > 0, \\ \min\{\tau_{cp}, \tau_\Delta\}, & \text{if } a_\tau > 0, a^T g < 0. \end{cases} \tag{2.15}$$

Proof. For the subproblem (P1), we know that $\Omega = [-\tau_\Delta, \tau_\Delta]$ where $\tau_\Delta \leq \tau_d < \tau_m$.
(1) If $a_\tau \leq 0$, then from Lemma 2.1 (1)(2) we can easily obtain $\tau_* = -\tau_\Delta$.
(2) If $a_\tau > 0, a^T g > 0$, then $\tau_{cp} < 0$. From Lemma 2.1 (3), we can obtain that if $\tau_{cp} \leq -\tau_\Delta$ then $\tau_* = -\tau_\Delta$; If $-\tau_\Delta < \tau_{cp} < 0$, then $\tau_* = \tau_{cp}$. Therefore, $\tau_* = \max\{-\tau_\Delta, \tau_{cp}\}$.
(3) If $a_\tau > 0, a^T g < 0$, then $\tau_{cp} > 0$. From Lemma 2.1 (3), we can obtain that if $0 < \tau_{cp} \leq \tau_\Delta$ then $\tau_* = \tau_{cp}$; If $\tau_\Delta < \tau_{cp} < \tau_m$, then $\tau_* = \tau_\Delta$. Therefore, $\tau_* = \min\{\tau_{cp}, \tau_\Delta\}$. □

Theorem 2.3. *If $a^T g \neq 0$, then the optimal solution of the subproblem (P2) is*

$$\tau_* = \begin{cases} -\tau_\Delta, & \text{if } a_\tau \leq 0, \\ \max\{-\tau_\Delta, \tau_{cp}\}, & \text{if } a_\tau > 0, a^T g > 0, \\ \min\{\tau_{cp}, \tau_d\}, & \text{if } a_\tau > 0, a^T g < 0. \end{cases} \tag{2.16}$$

4

Proof. The proof process is similar to the above Theorem 2.2, so we omitted it. □

Theorem 2.4. *If $a_\tau < 0$, then $a^T g > 0$, $\tau_m < \tau_{cp}$ and the optimal solution of the subproblem (P3) is*

$$\tau_* = \begin{cases} \tau_u, & \text{if } \tau_m < \tau_{cp} \le \tau_u, \\ \tau_{cp}, & \text{if } \tau_u < \tau_{cp} < \tau_\Delta, \\ \tilde{\tau}_\Delta, & \text{if } \tau_{cp} \ge \tau_\Delta, \end{cases} \tag{2.17}$$

where

$$\tilde{\tau}_\Delta = \arg\min\{\rho(-\tau_\Delta), \rho(\tau_\Delta)\}. \tag{2.18}$$

Proof. For the subproblem (P3), we know that

$$\Omega = [-\tau_\Delta, \tau_d] \cup [\tau_u, \tau_\Delta], \tag{2.19}$$

where $\tau_d < \tau_m < \tau_u$. If $a_\tau < 0$, then $a^T g > 0$. And from Lemma 2.1 (1) we can easily obtain that $\tau_m < \tau_{cp}$ and

$$\tau_* = \begin{cases} \arg\min\{\rho(-\tau_\Delta), \rho(\tau_u)\}, & \text{if } \tau_m < \tau_{cp} \le \tau_u, \\ \arg\min\{\rho(-\tau_\Delta), \rho(\tau_{cp})\}, & \text{if } \tau_u < \tau_{cp} < \tau_\Delta, \\ \arg\min\{\rho(-\tau_\Delta), \rho(\tau_\Delta)\}, & \text{if } \tau_{cp} \ge \tau_\Delta. \end{cases} \tag{2.20}$$

(1) If $\tau_m < \tau_{cp} \le \tau_u$, then from (2.2) we have

$$\rho(\tau_u) - \rho(-\tau_\Delta) = \frac{\Delta^2 \|a\|^2 a_\Delta + 2\Delta \|a\| b_\Delta + c_\Delta}{2\varepsilon_0^2 \|a\|^4 (1 + \Delta \|a\|)^2}, \tag{2.21}$$

where

$$a_\Delta = (1 + 2\varepsilon_0) a^T B a - 2\varepsilon_0 \|a\|^2 a^T g, \tag{2.22}$$
$$b_\Delta = (1 + \varepsilon_0)^2 a^T B a - (2 + \varepsilon_0)\varepsilon_0 \|a\|^2 a^T g, \tag{2.23}$$
$$c_\Delta = (1 + \varepsilon_0)^2 a^T B a - 2(1 + \varepsilon_0)\varepsilon_0 \|a\|^2 a^T g. \tag{2.24}$$

Because $\tau_{cp} \le \tau_u$, then from (2.4) and (2.11) we have

$$-\varepsilon_0 \|a\|^2 a^T g \le -(1 + \varepsilon_0) a^T B a. \tag{2.25}$$

And then

$$a_\Delta \le -a^T B a < 0, \tag{2.26}$$
$$b_\Delta \le -(1 + \varepsilon_0) a^T B a < 0, \tag{2.27}$$
$$c_\Delta \le -(1 + \varepsilon_0)^2 a^T B a < 0. \tag{2.28}$$

Combining with (2.21), then

$$\rho(\tau_u) < \rho(-\tau_\Delta).$$

Hence, $\tau_* = \tau_u$.

(2) If $\tau_u < \tau_{cp} < \tau_\Delta$, then from (2.2) we have

$$\rho(\tau_{cp}) - \rho(-\tau_\Delta) = -\frac{a_\tau^2 \Delta^2 - 2a_\tau a^T g \|a\| \Delta + \|a\|^2 (a^T g)^2}{2\|a\|^2 (1 + \Delta \|a\|)^2 a^T B a}. \tag{2.29}$$

Because $a_\tau < 0, a^T g > 0$, then

$$\rho(\tau_{cp}) < \rho(-\tau_\Delta).$$

Therefore, $\tau_* = \tau_{cp}$. The theorem is proved. □

Theorem 2.5. *If $a_\tau \geq 0$ and $a^T g \neq 0$, then the optimal solution of the subproblem (P3) is*

$$\tau_* = \begin{cases} -\tau_\Delta, & \text{if } a_\tau = 0, \\ \max\{-\tau_\Delta, \tau_{cp}\}, & \text{if } a_\tau > 0, a^T g > 0, \\ \min\{\tau_{cp}, \tau_d\}, & \text{if } a_\tau > 0, a^T g < 0. \end{cases} \tag{2.30}$$

Proof. (1) If $a_\tau = 0$ then $a^T g > 0$. Combining (2.19) and Lemma 2.1 (2), we know that

$$\tau_* = \arg\min\{\rho(-\tau_\Delta), \rho(\tau_\Delta)\}. \tag{2.31}$$

However, by calculation we have

$$\rho(\tau_\Delta) - \rho(-\tau_\Delta)$$
$$= \frac{2\Delta a^T g}{\|a\|(1 - \Delta^2\|a\|^2)} + \frac{2\Delta^3 a^T Ba}{\|a\|(1 - \Delta^2\|a\|^2)^2} \tag{2.32}$$
$$= \frac{2\Delta(\Delta^2 a_\tau + a^T g)}{\|a\|(1 - \Delta^2\|a\|^2)^2}. \tag{2.33}$$

For $a_\tau = 0$ and $a^T g > 0$, then

$$\rho(\tau_\Delta) > \rho(-\tau_\Delta).$$

Hence, $\tau_* = -\tau_\Delta$ and (2.30) holds.

(2) If $a_\tau > 0, a^T g > 0$ then $\tau_{cp} < 0$. Combining (2.19) and Lemma 2.1 (3), we know that the optimal solution of the subproblem (P3) is

$$\tau_* = \begin{cases} \arg\min\{\rho(-\tau_\Delta), \rho(\tau_\Delta)\}, & \text{if } \tau_{cp} \leq -\tau_\Delta, \\ \arg\min\{\rho(\tau_{cp}), \rho(\tau_\Delta)\}, & \text{if } -\tau_\Delta < \tau_{cp} < 0. \end{cases} \tag{2.34}$$

For $a_\tau > 0, a^T g > 0$, then from (2.33) we note that

$$\rho(\tau_\Delta) > \rho(-\tau_\Delta). \tag{2.35}$$

If $-\tau_\Delta < \tau_{cp} < 0$, then from Lemma 2.1 (3) we know that

$$\rho(-\tau_\Delta) > \rho(\tau_{cp}).$$

Thus,

$$\tau_* = \begin{cases} -\tau_\Delta, & \text{if } \tau_{cp} \leq -\tau_\Delta, \\ \tau_{cp}, & \text{if } -\tau_\Delta < \tau_{cp} < 0. \end{cases} \tag{2.36}$$

Then, (2.30) holds.

(3) If $a_\tau > 0, a^T g < 0$, then from (2.11) and (2.12) we can get $0 < \tau_{cp} < \tau_m$. Combining (2.19) and Lemma 2.1 (3), we know that the optimal solution of the subproblem (P3) is

$$\tau_* = \begin{cases} \arg\min\{\rho(\tau_{cp}), \rho(\tau_\Delta)\}, & \text{if } 0 < \tau_{cp} < \tau_d, \\ \arg\min\{\rho(\tau_d), \rho(\tau_\Delta)\}, & \text{if } \tau_d \leq \tau_{cp} < \tau_m. \end{cases} \tag{2.37}$$

For the subproblem (P3), we note that $1 - \Delta\|a\| \leq -\varepsilon_0$. Because of $a^T g < 0$, then

$$\rho(\tau_\Delta) = \frac{\Delta a^T g}{\|a\|(1 - \Delta\|a\|)} + \frac{\Delta^2 a^T Ba}{2\|a\|^2(1 - \Delta\|a\|)^2} > 0. \tag{2.38}$$

However, from $\rho(0) = 0$ and Lemma 2.1 (3) we can obtain that if $0 < \tau_{cp} < \tau_d$ then $\rho(\tau_{cp}) < 0$; If $\tau_d \leq \tau_{cp} < \tau_m$ then $\rho(\tau_d) < 0$ holds too. Therefore, it follows that

$$\tau_* = \begin{cases} \tau_{cp}, & \text{if } 0 < \tau_{cp} < \tau_d, \\ \tau_d, & \text{if } \tau_d \leq \tau_{cp} < \tau_m. \end{cases} \tag{2.39}$$

Then, (2.30) holds too and the theorem is proved. $\qquad\square$

If $\tau_* = \tau_\Delta$, then from (2.4) we know that $\|\tau_* a\| = \Delta$. Therefore, for this case we set $s_* = \tau_* a$ and exit the calculation of subproblem. Otherwise, we know that $\tau_* a$ is inside the trust region. Then, we should carry out the calculation of the second stage below.

We set $s = \tau_* a + y$ and substitute it into $\phi_k(s)$. And then the subproblem (1.7)-(1.8) becomes

$$\min \quad \psi(y) = \frac{g^T(\tau_* a + y)}{1 - \tau_* a^T a} + \frac{(\tau_* a + y)^T B(\tau_* a + y)}{2(1 - \tau_* a^T a)^2}, \tag{2.40}$$

$$\text{s.t.} \quad \|y\| \leq \tilde{\Delta}, \ a^T y = 0, \tag{2.41}$$

where

$$\tilde{\Delta} = \sqrt{\Delta^2 - (\tau_*)^2 \|a\|^2}. \tag{2.42}$$

In order to remove the equality constraint in (2.41), we use the null space technique. That is, for $a \neq 0$ then there exist $n - 1$ mutually orthogonal unit vectors $q, q, \cdots, q_{n-1}$ orthogonal to the parameter vector a. Set $Q = [q, q, \cdots, q_{n-1}]$ and $y = Qu$, where $u \in R^{n-1}$. Then (2.40)-(2.41) can be simplified as following subproblem

$$\min \quad \tilde{\psi}(u) = \tilde{g}^T u + \frac{1}{2} u^T \tilde{B} u, \tag{2.43}$$

$$\text{s.t.} \quad \|u\| \leq \tilde{\Delta}, \tag{2.44}$$

where

$$\tilde{g} = \frac{Q^T g}{1 - \tau_* a^T a} + \frac{\tau_* Q^T B a}{(1 - \tau_* a^T a)^2}, \quad \tilde{B} = \frac{Q^T B Q}{(1 - \tau_* a^T a)^2} \tag{2.45}$$

Set $g_k = \tilde{g}$, $B_k = \tilde{B}$ and $\Delta_k = \tilde{\Delta}$. By Algorithm 1.1, we can obtain the solution u_* of the subproblem (2.43)-(2.44). Then $y_* = Qu_*$ and $s_* = \tau_* a + y_*$. Thus, the subproblem (1.7)-(1.8) is solved approximately.

Now we could give the alternating direction search method for solving the conic trust region subproblem (1.7)-(1.8) as following.

Algorithm 2.1
Given ε_0, a, g, B and Δ.
Step 1. If $a^T g = 0$, then $\tau_* = 0$. Set $a = 0$ and use Algorithm 1.1 to get s_k, stop.
Step 2. Compute $\tau_{cp}, \tau_d, \tau_u, \tau_\Delta$ and a_τ by (2.4), (2.10) and (2.11).
Step 3. Compute $1 - \Delta \|a\|$.
Step 4. Solve the subproblem (2.2)-(2.3).
      Step 4.1. If $1 - \Delta \|a\| \geq \varepsilon_0$, then calculate τ_* by (2.15); If $|1 - \Delta \|a\|| < \varepsilon_0$, then calculate τ_* by (2.16); Otherwise, go to step 4.2.
      Step 4.2. If $a_\tau < 0$ then calculate τ_* by (2.17); If $a_\tau \geq 0$ then calculate τ_* by (2.30);
Step 5. If $\tau_* = \pm \tau_\Delta$, then $s_k = \pm \tau_\Delta a$, and stop. Otherwise, compute $Q, \tilde{\Delta}, \tilde{g}$ and $\tilde{B}$ by (2.42) and (2.45).
Step 6. Set $g_k = \tilde{g}$, $B_k = \tilde{B}$ and $\Delta_k = \tilde{\Delta}$. Then solve the subproblem (2.43)-(2.44) by Algorithm 1.1 to get u_*.
Step 7. Set $y_* = Qu_*$ and $s_k = \tau_* a + y_*$, and stop.

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

$$\text{pred}(s) = \phi(0) - \phi(s), \quad \text{pred}_1(\tau) = \rho(0) - \rho(\tau) \tag{2.46}$$

$$\text{pred}_2(y) = \psi(0) - \psi(y), \quad \text{pred}_3(u) = \tilde{\psi}(0) - \tilde{\psi}(u) \tag{2.47}$$

Now we should prove the following theorem to guarantee the global convergence of the algorithm proposed in the next section.

Theorem 2.6. *Under the same conditions as Lemma 2.1. If $s_k = \pm\tau_\Delta a$ are obtained by Steps 5 in Algorithm 2.1, then there exists a positive constant c_1 such that*

$$pred(s_k) \geq \frac{1}{2}c_1\Delta\|g\|. \tag{2.48}$$

Proof. (1) If $s_k = \tau_\Delta a$, then we know that $\tau_* = \tau_\Delta$. By computation, we have

$$pred(s_k) = pred_1(\tau_\Delta) = -\rho(\tau_\Delta)$$
$$= \frac{-\Delta(\Delta a_\tau + 2\|a\|a^T g - \Delta\|a\|^2 a^T g)}{2\|a\|^2(1 - \Delta\|a\|)^2},$$

where τ_Δ is generated in two cases as defined in (2.15) and (2.17). In both cases, we can find $\tau_\Delta \leq \tau_{cp}$ and

$$\Delta a_\tau + \|a\|a^T g \leq 0. \tag{2.49}$$

Then

$$pred(s_k) = pred_1(\tau_\Delta) \geq \frac{-\Delta a^T g}{2\|a\|(1 - \Delta\|a\|)}. \tag{2.50}$$

(1a) For $1 - \Delta\|a\| \geq \varepsilon_0$, then from (2.15) we know that $a_\tau > 0$ and $a^T g < 0$. Combining with (2.49) and (2.50) , we have

$$pred(s_k) = pred_1(\tau_\Delta) \geq \frac{\epsilon\Delta\|g\|}{2}, \tag{2.51}$$

where

$$\epsilon = \frac{|a^T g|}{\|a\|\|g\|}. \tag{2.52}$$

(1b) For $1 - \Delta\|a\| \leq -\varepsilon_0$, then from (2.17) we know that $a_\tau < 0$ and $a^T g > 0$. Because of $1 - \Delta\|a\| < 0$ and $a^T g > 0$, then from (2.50) we also have (2.51) holds.

(2) If $s_k = -\tau_\Delta a$, then

$$pred(s_k) = pred_1(-\tau_\Delta) = -\rho(-\tau_\Delta)$$
$$= \frac{\Delta(-a_\tau\Delta + 2a^T g\|a\| + a^T g\|a\|^2\Delta)}{2\|a\|^2(1 + \Delta\|a\|)^2}. \tag{2.53}$$

where $-\tau_\Delta$ is generated in the following three cases as defined in (2.15)-(2.17) and (2.30).

(2a) For $1 - \Delta\|a\| \geq \varepsilon_0$, then $1 \leq 1 + \Delta\|a\| \leq 2 - \varepsilon_0$.

From (2.15), we know that if $a_\tau \leq 0$ then $a^T g > 0$. Thus,

$$-a_\tau\Delta + a^T g\|a\| \geq 0. \tag{2.54}$$

And then, from (2.53) we know

$$pred(s_k) = pred_1(-\tau_\Delta)$$
$$\geq \frac{\Delta a^T g}{2\|a\|(1 + \Delta\|a\|)} \geq \frac{\epsilon\Delta\|g\|}{2(2 - \varepsilon_0)}. \tag{2.55}$$

On the other hand, if $a_\tau > 0, a^T g > 0$ then $-\tau_\Delta \geq \tau_{cp}$. Then from (2.4) and (2.11) we have (2.54) holds too. It follows that (2.55) holds.

(2b) For $|1 - \Delta\|a\|| < \varepsilon_0$, then $2 - \varepsilon_0 < 1 + \Delta\|a\| < 2 + \varepsilon_0$.

Combining with (2.16), we can prove that (2.54) holds by the same way and

$$
\begin{aligned}
\operatorname{pred}(s_k) &= \operatorname{pred}_1(-\tau_\Delta) \\
&\geq \frac{\Delta a^T g}{2\|a\|(1 + \Delta\|a\|)} \geq \frac{\epsilon\Delta\|g\|}{2(2 + \varepsilon_0)}.
\end{aligned}
\tag{2.56}
$$

(2c) For $1 - \Delta\|a\| \leq -\varepsilon_0$, then $1 + \Delta\|a\| \geq 2 + \varepsilon_0$.
From (2.17), we know that if $a_\tau < 0$, then

$$
\rho(-\tau_\Delta) \leq \rho(\tau_\Delta).
$$

By the definition of $\operatorname{pred}_1(\tau)$ in the (2.46), we get

$$
\operatorname{pred}_1(-\tau_\Delta) \geq \operatorname{pred}_1(\tau_\Delta).
$$

Combining with the proof of the above case (1a) in this theorem, we have

$$
\begin{aligned}
\operatorname{pred}(s_k) &= \operatorname{pred}_1(-\tau_\Delta) \\
&\geq \operatorname{pred}_1(\tau_\Delta) \geq \frac{\epsilon\Delta\|g\|}{2}.
\end{aligned}
\tag{2.57}
$$

Therefore, the theorem follows from (2.51) and (2.55)-(2.57) with

$$
c_1 = \min\left\{\epsilon, \frac{\epsilon}{2 - \varepsilon_0}, \frac{\epsilon}{2 + \varepsilon_0}\right\} = \frac{\epsilon}{2 + \varepsilon_0}.
\tag{2.58}
$$

$\square$

Theorem 2.7. *Under the same conditions as Lemma 2.1. If s_k is obtained from the above Algorithm 2.1, then there exists a positive constant c_4 such that*

$$
pred(s_k) \geq \frac{1}{2} c_4 \|g\| \min\left\{\Delta, \frac{1}{\|a\|}, \frac{\|g\|}{\|B\|}\right\}.
\tag{2.59}
$$

Proof. (1) If s_k is obtained by Algorithm 1.1, then from Nocedal and Wright [36] we have

$$
\operatorname{pred}(s_k) \geq \frac{1}{2} c_2 \|g\| \min\left\{\Delta, \frac{\|g\|}{\|B\|}\right\},
\tag{2.60}
$$

where $c_2 \in (0, 1]$.
(2) If $s_k = \pm\tau_\Delta a$, then (2.48) holds.
(3) $s_k = \tau_* a + Q u_*$, where $\tau_* \neq \pm\tau_\Delta$. Combining with (2.46) and (2.47), we have

$$
\operatorname{pred}(s_k) = \operatorname{pred}_1(\tau_*) + \operatorname{pred}_3(u_*).
$$

Because of u_* is obtained by Algorithm 1.1, then from [36] we have

$$
\operatorname{pred}_3(u_*) \geq \frac{1}{2} c_3 \|\tilde{g}\| \min\left\{\tilde{\Delta}, \frac{\|\tilde{g}\|}{\|\tilde{B}\|}\right\},
\tag{2.61}
$$

where $c_3 \in (0, 1]$, $\tilde{\Delta}$, $\tilde{g}$ and $\tilde{B}$ as defined by (2.42) and (2.45). Thus,

$$
\operatorname{pred}(s_k) \geq \operatorname{pred}_1(\tau_*),
\tag{2.62}
$$

where τ_* can be τ_{cp}, τ_d or τ_u.

9

(3a) If $\tau_* = \tau_{cp}$, then from (2.62) we have

$$
\begin{aligned}
\text{pred}(s_k) &\geq \text{pred}_1(\tau_{cp}) \\
&= -\frac{\tau_{cp}a^T g}{1 - \tau_{cp}\|a\|^2} - \frac{\tau_{cp}^2 a^T Ba}{2(1 - \tau_{cp}\|a\|^2)^2} \\
&= \frac{(a^T g)^2}{2a^T Ba} = \frac{\epsilon^2\|g\|^2}{2\|B\|},
\end{aligned} \tag{2.63}
$$

where the second equality is from (2.11) and the last equality is from (2.52).

(3b) If $\tau_* = \tau_d$, then

$$
\begin{aligned}
\text{pred}(s_k) &\geq \text{pred}_1(\tau_d) \\
&= -\frac{\tau_d a^T g}{1 - \tau_d\|a\|^2} - \frac{\tau_d^2 a^T Ba}{2(1 - \tau_d\|a\|^2)^2} \\
&= \frac{-(1 - \varepsilon_0)[2\varepsilon_0\|a\|^2 a^T g + (1 - \varepsilon_0)a^T Ba]}{2\varepsilon_0^2\|a\|^4}.
\end{aligned}
$$

From (2.15)-(2.17) and (2.30), we know that $\tau_d \leq \tau_{cp}$ and $a^T g < 0$. For $\tau_d \leq \tau_{cp}$, then we have

$$
\varepsilon_0\|a\|^2 a^T g + (1 - \varepsilon_0)a^T Ba \leq 0
$$

and

$$
\begin{aligned}
\text{pred}(s_k) &\geq \text{pred}_1(\tau_d) \\
&\geq \frac{-(1 - \varepsilon_0)a^T g}{2\varepsilon_0\|a\|^2} = \frac{\epsilon(1 - \varepsilon_0)\|g\|}{2\varepsilon_0\|a\|},
\end{aligned} \tag{2.64}
$$

where $0 < \varepsilon_0 < 1$.

(3c) If $\tau_* = \tau_u$, then

$$
\begin{aligned}
\text{pred}(s_k) &\geq \text{pred}_1(\tau_u) \\
&= -\frac{\tau_u a^T g}{1 - \tau_u\|a\|^2} - \frac{\tau_u^2 a^T Ba}{2(1 - \tau_u\|a\|^2)^2} \\
&= \frac{(1 + \varepsilon_0)[2\varepsilon_0\|a\|^2 a^T g - (1 + \varepsilon_0)a^T Ba]}{2\varepsilon_0^2\|a\|^4}.
\end{aligned}
$$

From (2.15)-(2.17) and (2.30), we know that $\tau_{cp} \leq \tau_u$, $a_\tau < 0$ and $a^T g > 0$. For $\tau_{cp} \leq \tau_u$, then we have

$$
\varepsilon_0\|a\|^2 a^T g - (1 + \varepsilon_0)a^T Ba \geq 0
$$

and

$$
\begin{aligned}
\text{pred}(s_k) &\geq \text{pred}_1(\tau_u) \\
&\geq \frac{(1 + \varepsilon_0)a^T g}{2\varepsilon_0\|a\|^2} = \frac{\epsilon(1 + \varepsilon_0)\|g\|}{2\varepsilon_0\|a\|}.
\end{aligned} \tag{2.65}
$$

Therefore, the theorem follows from (2.48), (2.60) and (2.63)-(2.65) with

$$
c_4 = \min\left\{c_1, c_2, \epsilon^2, \frac{\epsilon(1 - \varepsilon_0)}{\varepsilon_0}\right\}. \tag{2.66}
$$

$\square$

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 (1.1), we approximate $f(x)$ with a conic model of the form

$$m_k(s) = f_k + \frac{g_k^T s}{1 - a_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1 - a_k^T s)^2}, \tag{3.1}$$

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

The choice of the parameters a_k and B_k in (3.1) can refer to [16–18, 26, 27] and [37, 38] respectively. We set

$$s_{k-1} = x_k - x_{k-1}, \tag{3.2}$$

$$\beta = (f_k - f_{k-1})^2 - (g_{k-1}^T s_{k-1})(g_k^T s_{k-1}), \tag{3.3}$$

If $\beta > 0$, then

$$\beta_k = \frac{f_{k-1} - f_k + \sqrt{\beta}}{-g_{k-1}^T s_{k-1}}; \tag{3.4}$$

otherwise, $\beta_k = 1$. In the updating process, we compute

$$a_k = \frac{1 - \beta_k}{g_{k-1}^T s_{k-1}} g_{k-1}, \tag{3.5}$$

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{z_k z_k^T}{z_k^T s_k}, \tag{3.6}$$

where

$$z_k = \theta y_k + (1 - \theta) B_k s_k, \ \theta \in [0, 1], \tag{3.7}$$

$$\theta = \begin{cases} 1, & \text{if } y_k^T s_k \geq 0.2 s_k^T B_k s_k, \\ \dfrac{0.8 s_k^T B_k s_k}{s_k^T B_k s_k - y_k^T s_k}, & \text{otherwise,} \end{cases} \tag{3.8}$$

and $y_k = g_{k+1} - g_k$.

Let s_k be the solution of the subproblem (1.7)-(1.8) by Algorithm 2.1. 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

$$\text{ared}(s_k) = f(x_k) - f(x_k + s_k) \tag{3.9}$$

and the reduction predicted by the conic model

$$\text{pred}(s_k) = -\frac{g_k^T s_k}{1 - a_k^T s_k} - \frac{1}{2} \frac{s_k^T B_k s_k}{(1 - a_k^T s_k)^2} \tag{3.10}$$

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

$$x_{k+1} = x_k + s_k \tag{3.11}$$

and adjusting the trust-region radius; otherwise the iteration is repeated at point x_k with a reduced trust-region radius.

Now we give the alternating direction trust-region algorithm based on conic model (3.1).

Algorithm 3.1 (**ADCTR**).

Step 0. Choose parameters $\epsilon, \varepsilon, \varepsilon_0 \in (0, 1)$, $0 < \eta_1 < \eta_2 < 1$, $0 < \delta_1 < 1 < \delta_2$ and $\bar{\Delta} > 0$; give a starting point $x_0 \in R^n$, $B_0 \in R^{n \times n}$, $a_0 \in R^n$ and an initial trust region radius $\Delta_0 \in (0, \bar{\Delta}]$; set $k = 0$.

Step 1. Compute f_k and g_k. If $\|g_k\| < \varepsilon$, 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 (1.7)-(1.8) by Algorithm 2.1 to get one of the approximate solution s_k.

Step 3. Compute ared(s_k), pred(s_k) and

$$r_k = \frac{\text{ared}(s_k)}{\text{pred}(s_k)}, \tag{3.12}$$

If $r_k \le \eta_1$, 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

$$\Delta_k = \begin{cases} \min\{\delta_2 \Delta_k, \bar{\Delta}\}, & \text{if } r_k \ge \eta_2, \|s_k\| = \Delta_k, \\ \Delta_k, & \text{otherwise.} \end{cases}$$

Step 4. Generate a_{k+1} and B_{k+1}; set $k = k + 1$, and go to Step 1.

In this algorithm, the procedure of "Step 2-Step 3-Step 2" is named as inner cycle. The following theorem guarantees that the ADCTR algorithm does not cycle infinitely in the inner cycle.

Assumption 3.1. The level set

$$L(x_0) = \{x | f(x) \le f(x_0)\}$$

and the sequence $\{\|a_k\|\}$, $\{\|g_k\|\}$ and $\{\|B_k\|\}$ are all uniformly bounded, B_k is symmetric and positive definite and f is twice continuously differentiable in $L(x_0)$.

From (3.10) and Theorem 2.2, we have

$$\text{pred}(s_k) \ge \frac{1}{2} c_4 \|g_k\| \min \left\{ \Delta_k, \frac{1}{\|a_k\|}, \frac{\|g_k\|}{\|B_k\|} \right\}, \tag{3.13}$$

where c_1 as defined by (2.66).

Theorem 3.1. *Suppose that Assumption 3.1 holds. s_k is the solution of conic trust-region subproblem (1.7)-(1.8). If the process does not terminate at x_k, then we must have $r_k > \eta_1$ after a finite number of inner iterations.*

Proof. We assume that the algorithm does not terminate at x_k, then there is $\varepsilon_1 > 0$ such that

$$\|g_k\| \ge \varepsilon_1. \tag{3.14}$$

From Assumption 3.1 we have

$$\|a_k\| \le \bar{a}, \quad \|g_k\| \le \bar{g}, \quad 0 < \|B_k\| \le \bar{B}. \tag{3.15}$$

For simplicity, we suppose that the superscript denotes the iterative step of inner iteration at x_k, then

$$r_k \le \eta_1, \quad \Delta_{k+1}^j = \delta_1 \Delta_k^j, \quad j = 1, 2, \cdots \tag{3.16}$$

Assume s_k^j is a solution of subproblem (1.7)-(1.8) with trust-region radius Δ_k^j, then it is easy to know that

$$\lim_{j \to \infty} \Delta_k^j = 0, \quad \lim_{j \to \infty} \|s_k^j\| = 0. \tag{3.17}$$

From (3.14), (3.15) and (3.17), we can obtain that there exist an integer j_1 and a constant $\eta_3 > 0$ such that

$$\text{pred}(s_k^j) \ge \eta_3 \Delta_k^j, \quad \forall j \ge j_1. \tag{3.18}$$

It follows from (3.16) that

$$r_k^j = \frac{f_k - f(x_k + s_k^j)}{\text{pred}(s_k^j)} \le \eta_1. \tag{3.19}$$

12

On the other hand, from (3.17) and (3.15) we can get

$$\frac{1}{1 - a_k^T s_k^j} = 1 + a_k^T s_k^j + o(\|s_k^j\|), \tag{3.20}$$

$$\frac{(s_k^j)^T B_k s_k^j}{2(1 - a_k^T s_k^j)^2} = \frac{1}{2}(s_k^j)^T B_k s_k^j + o(\|s_k^j\|^2). \tag{3.21}$$

And then, from (3.15)-(3.20) we have

$$\begin{aligned}
&\left| f_k - f(x_k + s_k^j) - \text{pred}(s_k^j) \right| \\
&= \left| f_k - f(x_k + s_k^j) + (1 + a_k^T s_k^j)g_k^T s_k^j + \frac{1}{2}(s_k^j)^T B_k s_k^j + o(\|s_k^j\|^2) \right| \\
&= \left| -\frac{1}{2}(s_k^j)^T \nabla^2 f(x_k + \vartheta_k s_k^j)s_k^j + a_k^T s_k^j g_k^T s_k^j + \frac{1}{2}(s_k^j)^T B_k s_k^j + o(\|s_k^j\|^2) \right| \\
&\leq \frac{1}{2}(M_1 + \bar{B} + 2\bar{a}\bar{g} + O(1))\|s_k^j\|^2 \\
&\leq \frac{1}{2}(Q + O(1))(\Delta_k^j)^2, \tag{3.22}
\end{aligned}$$

where $\vartheta_k \in (0, 1)$ and $Q = M_1 + \bar{B} + 2\bar{a}\bar{g}$. Combining with (3.18) and (3.22), we can get that

$$\left| \frac{f_k - f(x_k + s_k^j)}{\text{pred}(s_k^j)} - 1 \right| \leq \frac{(Q + O(1))}{2\eta_3} \Delta_k^j, \tag{3.23}$$

holds for all $j \geq j_1$. By (3.17) and (3.23),

$$\frac{f_k - f(x_k + s_k^j)}{\text{pred}(s_k^j)} > \eta_1 \tag{3.24}$$

holds for all sufficiently large j, which contradicts (3.19). This completes the proof. □

In the following we give the global convergence property of Algorithm 3.1.

Theorem 3.2. *Suppose that Assumption 3.1 holds. Then for any $\varepsilon > 0$, the Algorithm 3.1 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 $\varepsilon_2 > 0$ such that

$$\|g_k\| \geq \varepsilon_2, \ \forall k. \tag{3.25}$$

Combining with (3.13), (3.15) and (3.25), we have

$$\text{pred}(s_k) \geq \frac{1}{2}c_4\varepsilon_2 \min\left\{ \Delta_k, \frac{1}{\bar{a}}, \frac{\varepsilon_2}{\bar{B}} \right\} \geq \frac{1}{2}\zeta\Delta_k \tag{3.26}$$

where the first inequality of (3.26) follows from

$$\min\{p, q, r\} \geq \frac{pqr}{pq + qr + rp}, \ \forall p, q, r > 0,$$

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

$$\zeta = \frac{c_1\varepsilon_2^2}{\varepsilon_2 + \bar{B}\bar{\Delta} + \varepsilon_2\bar{a}\bar{\Delta}}.$$

13

From Steps 3 of Algorithm 3.1 and (3.26), we obtain that for all k

$$f_k - f_{k+1} \geq \eta_1 \text{pred}(s_k) \geq \frac{1}{2}\eta_1 \zeta \Delta_k. \tag{3.27}$$

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

$$\infty > \sum_{k \in S}(f_k - f_{k+1}) \geq \sum_{k \in S}\left(\frac{1}{2}\eta_1 \zeta \Delta_k\right). \tag{3.28}$$

Combining with Theorem 3.1, we know that

$$\sum_{k=1}^{\infty} \Delta_k < \infty, \tag{3.29}$$

which implies that

$$\lim_{k \to \infty} \Delta_k = 0, \ \lim_{k \to \infty} \|s_k\| = 0. \tag{3.30}$$

On the other hand, similar to the proof of (3.20)-(3.24) we can obtain

$$r_k = \frac{f_k - f(x_k + s_k)}{\text{pred}(s_k)} > \eta_1, \ \forall k \geq K, \tag{3.31}$$

where K is sufficiently large. From Step 3 of Algorithm 3.1, it follows that

$$\Delta_{k+1} \geq \Delta_k, \ \forall k \geq K,$$

which is a contradiction to (3.30). The theorem is proved. $\qquad\square$

4. Numerical Tests

In this section, algorithm ADCTR is tested with some standard test problems from [26, 40]. The purpose of this paper is to propose a new method to solve the conic trust region subproblem, that is alternating direction method, so we performed algorithm ADCTR on a limited number of test problems. The names of the 16 test problems are listed in Table 1.

All the computations are carried out in Matlab R2015b 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; 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 sign * means that when the number of iterations reaches 5000, the algorithm fails to stop. The parameters in these algorithms are

$$a_0 = 0, \ B_0 = I, \ \varepsilon_0 = \epsilon = 10^{-5}, \ \Delta_0 = 1, \ \bar{\Delta} = 10, \ \eta_1 = 0.01, \ \eta_2 = 0.75, \ \delta_1 = 0.5, \ \delta_2 = 2.$$

The numerical results of algorithm ADCTR for 16 unconstrained optimization problems 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 our algorithm can obtain the minimum value of the function after a finite number of iterations. And the corresponding minimum point is the stability point, which is also the optimal solution. Therefore, the performance of ADCTR is feasible and effective.

In order to analyze the effectiveness of our new algorithm, we compare ADCTR with the conic quasi-Newton trust region algorithm in which the subproblems are solved by the dogleg method (DCTR), see Zhu [26] and Lu [27]. As the dimensions of each test problem ranging from 2 to 4000, we have actually computed 48 numerical comparisons experiments and the numerical results are listed in Table 3. Analyzing the numerical results, we have the following conclusions: for the 16 problems, our algorithm ADCTR is better than the DCTR for 12 tests, is somewhat bad for 2 tests, and the two algorithms are same in efficiency for the other 2 tests; our algorithm in which the subproblems are solved by alternating direction method is competitive with algorithm DCTR in [26]. Especially for large-scale problems, our new algorithm has a strong numerical stability.

Table 1: Test functions.

No.	Problem	No.	Problem
1	Cube	2	Penalty-I
3	Beale	4	Conic
5	Extended powell	6	Variably Dimensioned
7	Rosenbrock	8	Extended Trigonometric
9	Tridiagonal Exponential	10	Brent
11	Troesch	12	Cragg and Levy
13	Broyden Tridiagonal	14	Brown
15	Discrete Boundary Value	16	Extended Trigonometric

Table 2: Results of ADCTR.

No.	n	$Iter$	nf/ng	f_k	$\|g\|$	CPU (s)
1	2	52	53/43	1.0377e-15	1.9477e-06	0.064681
2	2	10	11/11	9.0831e-06	8.9419e-06	0.048493
3	2	18	19/18	9.0379e-15	9.3925e-07	0.053525
4	2	16	17/13	1.1407e-12	2.1360e-06	0.050445
5	4	41	42/34	4.8648e-09	4.5887e-06	0.062011
6	4	32	33/29	2.3856e-14	3.0965e-07	0.066287
7	2	50	51/49	1.5486e-14	5.4101e-06	0.064227
8	4	47	48/34	7.9158e-15	4.1153e-07	0.076865
9	4	7	8/8	8.1577e-12	4.5905e-06	0.058505
10	4	81	82/58	5.8024e-18	4.6604e-07	0.089702
11	4	59	60/51	1.0955e-13	2.7230e-06	0.077290
12	4	48	49/43	1.1247e-08	5.2578e-06	0.068215
13	4	35	36/19	1.4498e-11	5.0442e-06	0.063276
14	2	91	92/52	0.1998e-06	2.5916e-07	0.089294
15	4	23	24/15	2.0042e-12	8.2898e-06	0.061544
16	4	14	15/15	3.0282e-04	4.9068e-06	0.048488

5. Conclusions

In this paper, we propose an alternating direction trust region method based on the conic model for unconstrained optimization and investigate its convergence. Conic models are more flexible to approximate objective functions and have stronger modeling property. Alternating direction method (ADM) has been well studied in the context of linearly constrained convex programming problems. It is because of the significant efficiency and easy implementation of ADM that we consider applying it to solving the trust region subproblem based on the conic model. Initial numerical results show that our new method is competitive and it is also effective and robust for large-scale problems. The numerical results and the theoretical results lead us to believe that the method is worthy of further study.

In addition, the main purpose of this paper is to explore a new method for solving the conic model subproblem. Therefore, there are many aspects worthy of further improvement and research in this paper. For example, we can consider the weak convergence assumptions that the Hessian approximations B_k is symmetric and positive semidefinite. The rate of convergence has not been studied.

Acknowledgements

References

[1] S. P. Han. A globally convergent method for nonlinear programming, Journal of Optimization Theory and Applications, 1977, 22(3):297–309.

[2] M. J. D. Powell. Variable Metric Methods for Constrained Optimization, Springer Berlin Heidelberg, 1983.

[3] Y. X. Yuan, W. Y. Sun, Conic Methods for Unconstrained Minimization and Tensor Methods for Nonlinear Equations, Science Press, Beijing, China, 1997.

[4] M. J. D. Powell, Y. X. Yuan, A trust region algorithm for equality constrained optimization, Mathematical Programming, 1990, 49(1):189–211.

[5] A. Vardi. A trust region algorithm for equality constrained minimization: Convergence properties and implementation, Siam Journal on Numerical Analysis, 1981, 22(3):575–591.

[6] P. T. Boggs, R. H. Byrd, R. B. Schnabel. A stable and efficient algorithm for nonlinear orthogonal distance regression, SIAM Journal on Scientific and Statistical Computing, 1987, 8(6):1052–1078.

[7] P. L. TOINT. Global convergence of a class of trust region methods for nonconvex minimization in hilbert space, IMA Journal of Numerical Analysis, 1988, 8(2):231–252.

[8] J. Z. Zhang, D. T. Zhu. Projected quasi-newton algorithm with trust region for constrained optimization, Journal of Optimization Theory and Applications, 1990, 67(2):369–393.

[9] M. El-Alem. A robust trust-region algorithm with a nonmonotonic penalty parameter scheme for constrained optimization, Siam Journal on Optimization, 1995, 5(2):348–378.

[10] A. R. Conn, N. I. M. Gould, P. L. Toint. Trust-region methods, Society for Industrial and Applied Mathematics, 2000.

[11] M. J. D. Powell, A hybrid method for nonlinear equations,In :Ph. D. Rabonowitz,Gordon and Breach, eds., Numerical Methods for Nonlinear Algebraic Equations, 1970, 87–114.

[12] J. E.,Dennis, H. H. W. Mei. Two new unconstrained optimization algorithms which use function and gradient values, Journal of Optimization Theory and Applications, 1979, 28(4):453–482.

[13] L. Zhang, Z. Q. Tang. The Hybrid Dogleg Method to Solve Subproblems of Trust Region. Journal of Nanjing Normal University, 2001, 24(1):28–32.

[14] J. Z. Zhang, X. J. Xu, D. T. Zhu. A nonmonotonic dogleg method for unconstrained optimization, SIAM Journal on Scientific and Statistical Computing, 1987, 8(6):1052–1078.

[15] Y. L. Zhao, C. X. X, R. B. Schnabel. A new trust region dogleg method for unconstrained optimization, Appl. Math. J. Chinese Univ. Ser. B, 2000, 15(1):83-92.

[16] W. C. Davidon, Conic approximations and collinear scalings for optimizers, Siam Journal on Numerical Analysis, 1980, 17(2):268–281.

[17] Schnabel R. Conic methods for unconstrained minimization and tensor methods for nonlinear equations. Math Prog: The State of the Art, (eds. A. Bachem, M. Grötschel and B. Korte), Heidelberg: Springer-Verlag, 1982: 417–438.

[18] Sorensen D C. Newton's method with a model trust region modification. SIAM J Numer Analy, 1982, 19(2): 409–426.

[19] Xu C X, Yang X Y. Convergence of conic quasi-Newton trust region methods for unconstrained minimization. Math Appl, 1998, 11(2): 71–76.

[20] Y X Yuan. A review of trust region algorithms for optimization. ICIAM, 2000, 99(1): 271–282.

[21] D M Gay. Computing optimal locally constrained steps. SIAM J Sci Stat Comput, 1981, 2(2): 186–197.

[22] J M Peng, Y X Yuan. Optimality conditions for the minimization of a quadratic with two quadratic constraints. SIAM J Optim, 1997, 7(3): 579–594.

[23] W. Y. Sun, Y. X. Yuan. A conic trust-region method for nonlinearly constrained optimization, Annals of Operations Research, 2001, 103(1):175–191.

[24] S. Di and W. Y. Sun, A trust region method for conic model to solve unconstraind optimizaions, *Optimization Methods and Software*, 1996, 6(4):237–263.

[25] Q. Ni, Optimality conditions for trust-region subproblems involving a conic model, *SIAM Journal on Optimization*, 2005, 15(3):826–837.

[26] M. Zhu, Y. Xue, Z. F. Sheng. A quasi-newton type trust region method based on the conic model, Numerical Mathematics A Journal of Chinese Universities, 1995, 17(1):36–47.

[27] Lu X P, Ni Q. A quasi-newton trust region method with a new conic model for the unconstrained optimization. Appl Math Comput, 2008, 204(1): 373–384.

[28] L. J. Zhao, W. Y. Sun. A conic affine scaling method for nonlinear optimization with bound constraints, 2013, 30(3):1-30.

[29] D. Gabay, B. Mercier. A dual algorithm for the solution of nonlinear variational problems via flnite-element approximations, Computer and Mathematics with Applications,1976, 2(1):17–40.

[30] G. Chen, M. Teboulle. A proximal-based decomposition method for convex minimization problems, Mathematical Programming, 1994, 64(1-3):81–101.

[31] J. Eckstein, M. Fukushima. Some reformulation and applications of the alternating direction method of multipliers, Large Scale Optimization: State of the Art, W. W. Hager etal eds., Kluwer Academic Publishers, 1994, 115–134.

[32] B. S. He, L. Z. Liao, D. Han, H. Yang. A new inexact alternating directions method for monontone variational inequalities, Mathematical Programming, 2002, 92(1):103–118.

[33] S. Kontogiorgis, R. R. Meyer. A variable-penalty alternating directions method for convex optimization, Mathematical Programming, 1998, 83(1):29–53.

[34] K. Zhang, J. S. Li, Y. C. Song, X. S. Wang. An alternating direction method of multipliers for elliptic equation constrained optimization problem, SCIENCE CHINA Mathematics, 2017, 60(2):361–378.

[35] M. H. Xu. Proximal Alternating Directions Method for Structured Variational Inequalities, Journal of Optimization Theory and Applications, 2007, 134(1):107–117.

[36] Jorge Nocedal, Stephen J. Wright. Numerical optimization. Science Press,Beijing, China, 2006.

[37] Powell M J D. Algorithms for nonlinear constraints that use Lagrange functions. Math Prog, 1978, 14(1): 224–248

[38] M. Al-Baali. Damped techniques for enforcing convergence of quasi-Newton methods. Optim Meth Softw, 2014, 29(5): 919–936

[39] H. Zhu, Q. Ni, M. L. Zeng. A quasi-newton trust region method based on a new fractional model, Numerical Algebra, Control and Optimization, 2015, 5(3):237–249.

[40] More J J, Garbow B S, Hillstrom K E. Testing unconstrained optimization software. ACM Trans. Math. Software, 1981, 7(1): 17–41.

Table 3: Numerical results of DCTR and ADCTR

Solver		DCTR				ADCTR			
No.	n	*Iter*	nf/ng	$\|g\|$	CPU (s)	*Iter*	nf/ng	$\|g\|$	CPU (s)
	20	746	747/517	9.2593e-07	0.116584	100	101/92	2.1475e-06	0.079679
1	200	2387	2388/2023	2.6984e-08	2.377265	82	83/59	4.6090e-06	0.308822
	1000	*	*/*	*	*	74	75/56	1.4596e-06	19.36706
	200	76	77/53	3.1715e-06	0.138719	79	80/53	4.1422e-06	0.137110
2	500	96	97/62	6.5292e-06	1.118043	78	79/54	4.6576e-06	1.270450
	1000	82	83/57	5.5181e-06	6.454441	86	87/57	8.8824e-06	8.026574
	2	20	21/19	4.3711e-07	0.042443	18	19/18	9.3925e-07	0.053525
3	20	24	25/19	2.4476e-07	0.042636	24	25/25	4.9198e-06	0.056501
	200	27	28/24	3.6411e-08	0.077384	26	27/27	2.0468e-07	0.145411
	2000	29	30/25	6.1805e-06	15.70090	35	36/28	8.8657e-06	54.47877
	20	15	16/14	5.7802e-07	0.039310	16	17/13	4.8378e-09	0.055520
4	200	16	17/12	3.7354e-06	0.051864	19	20/18	3.4444e-07	0.094543
	2000	18	19/19	3.9029e-07	13.50780	19	20/17	2.5377e-06	17.65469
	40	121	122/104	8.7449e-06	0.064345	48	49/43	6.0181e-06	0.076145
5	1000	121	122/116	2.3550e-06	14.01567	92	93/77	8.1189e-06	18.098231
	2000	121	122/116	2.6445e-06	106.0106	69	70/60	6.2005e-06	97.24934
6	40	120	121/75	6.0183e-06	0.125199	145	146/116	7.2779e-06	0.115281
	400	*	*/*	*	*	1124	1125/774	3.2877e-06	11.33673
	20	90	91/69	1.3138e-06	0.106181	83	84/54	1.0093e-06	0.082832
7	200	517	518/392	4.5464e-06	0.926231	61	62/52	2.0797e-06	0.242663
	2000	326	327/294	2.2237e-06	218.2702	71	72/54	7.8519e-07	112.6556
8	4	46	47/38	9.2635e-06	0.054172	47	48/34	4.1153e-07	0.076865
	40	*	*/*	*	*	354	355/265	9.1963e-06	0.147167
	40	6	7/7	1.1958e-06	0.054117	6	7/7	1.8900e-06	0.060258
9	400	6	7/7	1.6354e-07	0.111561	6	7/7	2.3374e-07	0.133484
	4000	11	12/12	8.4467e-07	40.15594	11	12/12	8.9545e-07	47.93941
10	4	377	378/298	8.2689e-06	0.175454	81	82/58	4.6604e-07	0.089702
	40	*	*/*	*	*	1260	1261/910	5.7484e-06	0.391677
	4	70	71/37	2.9831e-06	0.073789	59	60/51	2.7230e-06	0.077290
11	40	192	193/133	4.2981e-06	0.108448	132	133/122	3.1390e-06	0.116485
	500	*	*/*	*	*	1119	1120/1023	9.3082e-06	21.02191
	4	43	44/41	4.4263e-06	0.062761	48	49/43	5.2578e-06	0.068215
12	40	1977	1978/1315	8.1245e-06	0.369235	190	191/146	9.5513e-06	0.129097
	400	*	*/*	*	*	351	352/252	8.4470e-06	4.848008
	4	35	36/16	8.9785e-06	0.053999	35	36/19	5.0442e-06	0.063276
13	40	359	360/263	9.3216e-06	0.140429	47	48/29	7.5584e-06	0.084719
	400	1996	1997/1400	9.7095e-06	14.38582	55	56/34	9.2260e-06	0.746511
	1000	*	*/*	*	*	52	53/36	9.5547e-06	10.46032
	2	98	99/59	5.6830e-06	0.058219	91	92/52	2.5916e-07	0.089294
14	20	164	165/87	6.2362e-06	0.076377	125	126/98	9.9306e-06	0.094535
	200	*	*/*	*	*	209	210/161	9.3905e-06	0.656179
	4	27	28/16	4.2390e-07	0.063120	23	24/15	8.2898e-06	0.061544
15	400	33	34/11	8.4956e-06	0.162408	35	36/15	7.7218e-06	0.202917
	1000	21	22/2	9.0840e-06	0.101637	21	22/2	9.0840e-06	0.160027
	4000	25	26/2	5.6751e-07	0.970886	25	26/2	5.6751e-07	2.331821
	4	19	20/16	1.4241e-06	0.051494	14	15/15	4.9068e-06	0.048488
16	40	518	519/329	7.5993e-06	0.117250	63	64/42	6.2298e-06	0.057708
	400	*	*/*	*	*	60	61/48	4.6296e-06	0.571713

18