

Available online at www.sciencedirect.com

Journal of Computational and Applied Mathematics 220 (2008) 119–128

JOURNAL OF
COMPUTATIONAL AND
APPLIED MATHEMATICSwww.elsevier.com/locate/cam

A nonmonotone trust-region method of conic model for unconstrained optimization[☆]

Shao-Jian Qu*, Ke-Cun Zhang, Jian Zhang

Faculty of Science, Xi'an Jiaotong University, Xi'an 710049, PR China

Received 12 May 2007; received in revised form 26 July 2007

Abstract

In this paper, we present a nonmonotone trust-region method of conic model for unconstrained optimization. The new method combines a new trust-region subproblem of conic model proposed in [Y. Ji, S.J. Qu, Y.J. Wang, H.M. Li, A conic trust-region method for optimization with nonlinear equality and inequality 4 constrains via active-set strategy, Appl. Math. Comput. 183 (2006) 217–231] with a nonmonotone technique for solving unconstrained optimization. The local and global convergence properties are proved under reasonable assumptions. Numerical experiments are conducted to compare this method with the method of [Y. Ji, S.J. Qu, Y.J. Wang, H.M. Li, A conic trust-region method for optimization with nonlinear equality and inequality 4 constrains via active-set strategy, Appl. Math. Comput. 183 (2006) 217–231].

© 2007 Elsevier B.V. All rights reserved.

Keywords: Unconstrained optimization; Trust-region method; Conic model; Nonmonotone technique

1. Introduction

In this paper, the following unconstrained optimization is considered:

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

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

Trust-region methods of quadratic model for unconstrained optimization have been studied by many researchers [5,12,13,15,16]. Trust-region methods 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 [11] show 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.

[☆] This work is supported by the National Natural Science Foundation of China Grant (Grant Number: 10671057).

* Corresponding author.

E-mail address: qushaojian@163.com (S.-J. Qu).

In [7], we proposed a new trust-region subproblem based on conic model for general constraints optimization. For unconstrained optimization this subproblem can be reduced to

$$\begin{cases} \min & \varphi_k(s) = \frac{g_k^T s}{1 - \alpha_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1 - \alpha_k^T s)^2} \\ \text{s.t.} & 1 - \alpha_k^T s > 0, \\ & \|s\| \leq \Delta_k, \end{cases} \quad (1.2)$$

where $\varphi_k(s)$ is called conic model which is an approximation to $f(x_k + s) - f(x_k)$, B_k is an approximate Hessian of $f(x)$ at x_k and Δ_k is the trust-region radius. The vector α_k is the associated vector for the collinear scaling in the k th iteration, and it is normally called the horizontal vector. If $\alpha_k = 0$, the conic model reduces to a quadratic model. Therefore the conic model methods are the generalization of the quadratic model methods. They have several advantages. First, if the objective function has strong nonquadratic 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, 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 [4,10], etc. show that the conic model method gives improvement over the quadratic model method. Finally, the conic model method has the similar global and local convergence properties as the quadratic model method.

Furthermore it is known that the objective function sequences generated by these algorithms are monotonically decreasing: i.e., $f(x_k) \geq f(x_{k+1})$, $k = 0, 1, \dots$

Recently, nonmonotone line search techniques have been studied by many authors since Grippo et al. [6]. Many authors generalized the nonmonotone technique to trust-region methods and proposed nonmonotone trust-region methods [17,8,2]. Theoretic analysis and numerical results show that the algorithms with nonmonotone properties are more efficient than the algorithms with monotone properties. To our knowledge, the nonmonotone trust region methods listed above are all based on quadratic model, but we have not seen any nonmonotone trust-region methods based on conic model.

In our paper, we combine the subproblem (1.2) with nonmonotone technique to propose a nonmonotone trust-region method based on conic model. The local and global convergence properties of the nonmonotone trust-region method based on conic model are proved under some reasonable assumptions. Finally, the numerical results show the efficiency of the new algorithm.

The rest of the paper is organized as follows. In Section 2, we present the nonmonotone 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.

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:

$$f_{l(k)} = \max_{0 \leq j \leq m(k)} \{f_{k-j}\}, \quad k = 0, 1, 2, \dots, \quad (2.1)$$

where $m(k) = \min\{m(k-1) + 1, 2M, M_k\}$, $m(0) := 0$, $M \geq 0$ is an integer constant and $M_k \geq 0$ is an integer variable. 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

$$\text{ared}_k(s_k) = f_{l(k)} - f(x_k + s_k) \quad (2.2)$$

and the reduction predicted by the conic model

$$\text{pred}_k(s_k) = -\frac{g_k^T s_k}{1 - \alpha_k^T s_k} - \frac{1}{2} \frac{s_k^T B_k s_k}{(1 - \alpha_k^T s_k)^2}. \quad (2.3)$$

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

$$x_{k+1} = x_k + s_k \quad (2.4)$$

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 the algorithm.

Algorithm NCTR (The nonmonotone conic trust-region algorithm for unconstrained optimization).

Step 0: Choose parameters $0 < c_3 < 1 < c_1$, $0 < c_0 \leq c_2 < 1$, $\Delta_{\max} > \Delta_{\min} > 0$ and $\varepsilon \geq 0$; give a starting point $x_0 \in \mathbb{R}^n$, $B_0 \in \mathbb{R}^{n \times n}$, $\alpha_0 \in \mathbb{R}^n$, an integer constant $M \geq 0$ and an initial trust-region radius $\Delta_{\min} \leq \Delta_0 < \Delta_{\max}$; set $k := 0$, $m(0) := 0$, $M_1 := M$.

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

Step 2: Solve the conic minimization subproblem (1.2) and let s_k be one approximate solution of the subproblem (1.2).

Step 3: If $k \geq 1$, set $m(k) := \min\{m(k-1) + 1, 2M, M_k\}$. Compute $\text{ared}_k(s_k)$, $\text{pred}_k(s_k)$ and

$$r_k = \frac{\text{ared}_k(s_k)}{\text{pred}_k(s_k)}.$$

If $r_k \leq c_0$, then set

$$\Delta_k := c_3 \|s_k\|, \quad M_k := M_k + 1, \quad (2.5)$$

and go to Step 2. If $r_k > c_0$, then

$$x_{k+1} := x_k + s_k, \quad M_{k+1} := M_k, \quad (2.6)$$

$$\Delta_{k+1} = \begin{cases} \max[c_1 \Delta_k, \Delta_{\min}] & \text{if } r_k \geq c_2, \\ \Delta_k & \text{otherwise.} \end{cases} \quad (2.7)$$

Step 4: Generate α_{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 trust-region subproblem exactly is expensive. Hence developing approximate methods for the trust-region subproblem has been a popular research topic since 1980s and numerous algorithms have been proposed. Recently, for solving the subproblem (1.2) an efficient approximate Algorithm 4.1 of [7] has been proposed. In this paper, we will use this algorithm to solve the conic trust-region subproblem (1.2).

(ii) The method for generating α_{k+1} and B_{k+1} can be seen, for example, in [3,14,1]. The conditions that we assume for proving global convergence are that the matrices B_k are uniformly bounded and

$$\forall k, \quad \exists \sigma \in (0, 1) : \|\alpha_k\| \Delta_k \leq \sigma \quad (2.8)$$

which ensures that the conic model function $\varphi_k(s)$ is bounded over the trust-region $\{s \mid \|s\| \leq \Delta_k\}$. 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 $\alpha_k \simeq 0$ asymptotically is suitable when x_k is near the minimizer.

(iii) If $M = 0$, this algorithm reduces to monotone one.

(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 assumptions.

Assumption 3.1. (i) The sequence $\{x_k\}$ generated by Algorithm NCTR is contained in a bounded set Ω and $f(x)$ is twice continuously differentiable in Ω for any given $x_0 \in \mathbb{R}^n$.

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

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

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

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{A} such that

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

Theorem 3.1. Suppose that (2.8) and Assumption 3.1 hold. Then there exists a positive constant δ_1 such that

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

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

Proof. Firstly, we let

$$s_k(t) = -tg_k, \quad (3.4)$$

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

$$\varphi_k(0) - \varphi_k(s_k) \geq \varphi_k(0) - \varphi_k(s_k(t)) \quad (3.5)$$

for all $t \in [0, \Delta_k/\|g_k\|]$. By using Cauchy–Schwartz inequality, we obtain

$$\begin{aligned} \varphi_k(0) - \varphi_k(s_k(t)) &\geq t \frac{\|g_k\|^2}{1+\sigma} - \frac{t^2}{2} \frac{g_k^T B_k g_k}{(1-\sigma)^2} \\ &\geq \frac{\|g_k\|^2}{2(1+\sigma)} \left(2t - t^2 \frac{1+\sigma}{(1-\sigma)^2} \|B_k\| \right) \end{aligned} \quad (3.6)$$

for all $t \in [0, \Delta_k/\|g_k\|]$. By computation, we have that

$$\max_{t \in [0, \Delta_k/\|g_k\|]} \left(2t - t^2 \frac{1+\sigma}{(1-\sigma)^2} \|B_k\| \right) \geq \min \left[\frac{\Delta_k}{\|g_k\|}, \frac{(1-\sigma)^2}{1+\sigma} \frac{1}{\|B_k\|} \right]. \quad (3.7)$$

Therefore the theorem follows from (3.6) and (3.7) with

$$\delta_1 = \frac{(1-\sigma)^2}{2(1+\sigma)^2}. \quad \square \quad (3.8)$$

Lemma 3.2. Suppose that (2.8) and Assumption 3.1 hold, then there exists one positive constant δ_2 such that

$$|f_k - f(x_k + s_k) - \text{pred}_k(s_k)| \leq \delta_2 \|s_k\|^2, \quad \forall k. \quad (3.9)$$

Proof. From the definition of $\text{pred}_k(s_k)$, we have that

$$\begin{aligned} &|f_k - f(x_k + s_k) - \text{pred}_k(s_k)| \\ &= \left| -g_k^T s_k - \frac{1}{2} s_k^T \nabla^2 f(x_k + \theta_k s_k) s_k + \frac{g_k^T s_k}{1 - \alpha_k^T s_k} + \frac{1}{2} \frac{s_k^T B_k s_k}{(1 - \alpha_k^T s_k)^2} \right| \\ &= \left| \frac{(\alpha_k^T s_k)(g_k^T s_k)}{1 - \alpha_k^T s_k} - \frac{1}{2} s_k^T (\nabla^2 f(x_k + \theta_k s_k) - B_k) s_k - \frac{1}{2} s_k^T B_k s_k \left(1 - \frac{1}{(1 - \alpha_k^T s_k)^2} \right) \right| \\ &\leq \left[\frac{\|\alpha_k\| \|g_k\|}{1 - \sigma} + \frac{1}{2} \|\nabla^2 f(x_k + \theta_k s_k) - B_k\| + \frac{1}{2} \left(1 + \frac{1}{(1 - \sigma)^2} \right) \right] \|s_k\|^2, \end{aligned} \quad (3.10)$$

where $\theta_k \in [0, 1]$. It follows from (3.10) and Assumption 3.1 that the lemma is true with

$$\delta_2 \geq \frac{\|\alpha_k\| \|g_k\|}{1 - \sigma} + \frac{1}{2} \|\nabla^2 f(x_k + \theta_k s_k) - B_k\| + \frac{1}{2} \left(1 + \frac{1}{(1 - \sigma)^2} \right). \quad \square$$

The following theorem guarantees that the NCTR algorithm does not cycle infinitely in the inner cycle.

Theorem 3.3. Suppose that (2.8) and Assumption 3.1 hold and that s_k is the solution of conic trust-region subproblem (1.2). That is, 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, $\|g_k\| \neq 0$. For simplicity, we suppose that the superscript denotes the iterative step of inner iteration at x_k , then

$$r_k^j \leq c_0, \quad \Delta_k^{j+1} = c_3 \|s_k\|, \quad j = 1, 2, \dots \quad (3.11)$$

and s_k^j is a solution of subproblem (1.2) with trust-region radius Δ_k^j . The above relations imply

$$\lim_{j \rightarrow \infty} \Delta_k^j = 0, \quad \lim_{j \rightarrow \infty} \|s_k^j\| = 0. \quad (3.12)$$

The above relation and Theorem 3.1 imply that there exist an integer j_1 and a constant $\delta_3 > 0$ such that

$$\text{pred}_k(s_k^j) \geq \delta_3 \Delta_k^j, \quad \forall j \geq j_1. \quad (3.13)$$

From the definition of $f_{l(k)}$ we have $f_{l(k)} \geq f_k$. It follows from (3.11) that

$$c_0 \geq r_k^j = \frac{f_{l(k)} - f(x_k + s_k^j)}{\text{pred}_k(s_k^j)} \geq \frac{f_k - f(x_k + s_k^j)}{\text{pred}_k(s_k^j)}. \quad (3.14)$$

On the other hand, from Lemma 3.2 and (3.13),

$$\begin{aligned} \left| \frac{f_k - f(x_k + s_k^j)}{\text{pred}_k(s_k^j)} - 1 \right| &= \frac{|f_k - f(x_k + s_k^j) - \text{pred}_k(s_k^j)|}{\text{pred}_k(s_k^j)} \\ &\leq \frac{\delta_2 \|s_k^j\|^2}{\delta_3 \Delta_k^j} \\ &\leq \frac{\delta_2}{\delta_3} \Delta_k^j \end{aligned} \quad (3.15)$$

holds for all $j \geq j_1$. By (3.12) and (3.15),

$$\frac{f_k - f(x_k + s_k^j)}{\text{pred}_k(s_k^j)} > c_0$$

holds for all sufficiently large j , which contradicts (3.14). This completes the proof. \square

Now we prove the global convergence of Algorithm NCTR.

Theorem 3.4. Under the same conditions as Theorem 3.3, assume that $\{x_k\}$ is an infinite sequence generated by Algorithm NCTR, then every limit point of $\{x_{l(k)-1}\}$ is a stationary point of (1.1)

Proof. Let x^* be any limit point of $\{x_{l(k)-1}\}$. Then there exists an infinite set $K \subset \{l(k) - 1 : k = 1, 2, \dots\}$ of indices such that $\lim_{k \in K} x_k = x^*$ and $\lim_{k \in K} g_k = g^*$. Suppose that x^* is not a stationary point of (1.1). Then

$$\|g^*\| > 0. \quad (3.16)$$

Next we consider two possibilities:

$$\liminf_{k \in K} \Delta_k = 0, \quad (3.17)$$

$$\liminf_{k \in K} \Delta_k > 0. \quad (3.18)$$

Assume first that (3.17) holds. Then there exists K_1 , an infinite subset of K , such that

$$\lim_{k \in K_1} \Delta_k = 0. \quad (3.19)$$

Therefore, there exists an integer k_1 such that $\Delta_k \leq \Delta_{\min}$ holds for all $k \in K_1$ and $k \geq k_1$, where Δ_{\min} is a positive constant.

By (3.16), there exists an integer $k_2 \geq k_1$ such that

$$\|g_k\| \geq \frac{1}{2} \|g^*\| \quad (3.20)$$

holds for all $k \in K_1$ and $k \geq k_2$. By (3.19), there exists an integer $k_3 \geq k_2$ such that

$$\frac{1}{c_3} \Delta_k < \|g_k\| \quad (3.21)$$

holds for all $k \in K_1$ and $k \geq k_3$. From the construction of our algorithm, we have that the trial step $s_k(\Delta'_k)$, corresponding to $\Delta'_k = (1/c_3)\Delta_k$, is an unacceptable trial step when $k \in K_1$ and $k \geq k_3$. Because $s_k(\Delta'_k)$ is an unacceptable trial step, we have

$$r_k(\Delta'_k) \leq c_0, \quad \forall k \in K_1, \quad k \geq k_3. \quad (3.22)$$

On the other hand, it follows from Theorem 3.1, (3.19) and (3.20) that

$$\text{pred}_k(s_k(\Delta'_k)) \geq \delta_1 \|g_k\| \min \left\{ \Delta'_k, \frac{\|g_k\|}{\|B_k\|} \right\} \geq \frac{\delta_1}{2} \|g^*\| \Delta'_k, \quad \forall k \in K_1, \quad k \geq k_3, \quad (3.23)$$

where k_3 is sufficiently large. From the definition of $f_{l(k)}$ we have $f_{l(k)} \geq f_k$, it follows from (3.22) that

$$c_0 \geq r_k(\Delta'_k) = \frac{f_{l(k)} - f(x_k + s_k(\Delta'_k))}{\text{pred}_k(s_k(\Delta'_k))} \geq \frac{f_k - f(x_k + s_k(\Delta'_k))}{\text{pred}_k(s_k(\Delta'_k))}, \quad (3.24)$$

where

$$\text{pred}_k(s_k(\Delta'_k)) = -\frac{g_k^T s_k(\Delta'_k)}{1 - \alpha_k^T s_k(\Delta'_k)} - \frac{s_k(\Delta'_k)^T B_k s_k(\Delta'_k)}{2(1 - \alpha_k s_k(\Delta'_k))^2}. \quad (3.25)$$

From the above relation and Lemma 3.2 we have that there exist two positive constant δ_4 and δ_5 such that

$$\begin{aligned} \left| \frac{f_k - f(x_k + s_k(\Delta'_k))}{\text{pred}_k(s_k(\Delta'_k))} - 1 \right| &= \frac{|f_k - f(x_k + s_k(\Delta'_k)) - \text{pred}_k(s_k(\Delta'_k))|}{\text{pred}_k(s_k(\Delta'_k))} \\ &\leq \frac{\delta_4 \|s_k(\Delta'_k)\|^2}{\delta_5 \Delta'_k} \leq \frac{\delta_4}{\delta_5} \Delta'_k \end{aligned} \quad (3.26)$$

holds for all $k \in K_1$ and $k \geq k_3$. Therefore by (3.19) and (3.26),

$$\frac{f_k - f(x_k + s_k(\Delta'_k))}{\text{pred}_k(s_k(\Delta'_k))} > c_0 \quad (3.27)$$

holds for all sufficiently large $k \in K_1$, which contradicts (3.24). Thus, relation (3.17) does not hold.

Now we assume that (3.18) holds. Then there exists a constant $\varepsilon > 0$ such that

$$\Delta_k \geq \varepsilon \quad (3.28)$$

holds for all $k \in K \subset \{l(k) - 1 : k = 1, 2, \dots\}$. On the other hand (3.3) holds for all Δ_k and $r_k(s_k) > c_0$. It follows that

$$f_{l(k)} - f_{k+1} \geq c_0 \text{pred}_k(s_k) \geq 0, \quad \forall k \in K. \quad (3.29)$$

Noting that $k \in K$ implies that $k + 1 \in \{l(k) : k = 1, 2, \dots\}$. Since $\{f_l(k)\}$ admits a limit, it follows that

$$\lim_{k \in K} \text{pred}_k(s_k) = 0. \quad (3.30)$$

Noting that (3.3) and (3.28) hold for all $k \in K$, similar to (3.23), we can obtain that

$$\text{pred}_k(s_k) \geq \frac{\delta_2}{2} \|g^*\| \min \left\{ \varepsilon, \frac{\|g^*\|}{A} \right\} > 0 \quad (3.31)$$

for all sufficiently large $k \in K$. The above relation contradicts (3.30). Therefore relation (3.18) does not hold either. This completes the proof of the theorem. \square

In a practical implementation, the stop criterion $\|g_k\| = 0$ in Algorithm NCTR is changed to $\|g_k\| < \varepsilon$, there $\varepsilon > 0$ is a constant. By Theorem 3.4, Algorithm NCTR stops in a finite number of iterations under Assumption 3.1 and (2.8). In order to explore the superlinear convergence we give the following assumptions.

Assumption 3.2. (i) The sequence $\{x_k\}$ generated by Algorithm NCTR converges to a stationary point x^* , i.e.,

$$\lim_{k \rightarrow \infty} x_k = x^* \quad \text{and} \quad \lim_{k \rightarrow \infty} \|g_k\| = \|g^*\| = 0. \quad (3.32)$$

(ii) If

$$\frac{\|B_k^{-1}g_k\|}{1 - g_k^T B_k^{-1} \alpha_k} \leq \Delta_k, \quad (3.33)$$

then

$$s_k = \frac{B_k^{-1}g_k}{1 - g_k^T B_k^{-1} \alpha_k}. \quad (3.34)$$

Lemma 3.5. Suppose that Assumptions 3.1 and 3.2 hold, then after finite iterations s_k must be defined as (3.34).

Proof. Define

$$K = \left\{ k \mid \frac{\|B_k^{-1}g_k\|}{1 - g_k^T B_k^{-1} \alpha_k} > \Delta_k \right\}. \quad (3.35)$$

Now we will prove that the set K is finite. If K is infinite, then by Assumption 3.2(i), we have that

$$\lim_{k \rightarrow \infty} \Delta_k = 0. \quad (3.36)$$

This together with Lemma 3.2 and the proof of Theorem 3.4, we have that

$$r_k > c_0 \quad (3.37)$$

holds for sufficiently large k . Then by Algorithm NCTR,

$$\Delta_{k+1} \geq \Delta_k \quad (3.38)$$

holds for sufficiently large k , which contradicts (3.36). Therefore the set K is finite. \square

Theorem 3.6. Suppose that Assumptions 3.1 and 3.2 hold. If $\nabla^2 f(x^*)$ is positive definite and

$$\lim_{k \rightarrow \infty} \frac{\| [B_k - \nabla^2 f(x^*)] s_k \|}{\| s_k \|} = 0, \quad (3.39)$$

then the sequence $\{x_k\}$ converges to x^* superlinearly.

Proof. By Lemma 3.5, we have that for large enough k , $\|B_k^{-1} g_k\| / (1 - g_k^T B_k^{-1} \alpha_k) \leq \Delta_k$. Then according to Assumption 3.2(ii), for large enough k , $s_k = B_k^{-1} g_k / (1 - g_k^T B_k^{-1} \alpha_k)$. So similar to the proof of Theorem 8 of [2], the theorem can be proved. \square

4. Numerical experiments

In this part, we will carry numeric experiments for the algorithm NCTR. All programs are written in C++, numerical test in PC, CPU Main Frequency 1.43GEMS 256MMrun circumstance VC++6.0, numeric type double float. The parameters in algorithm are:

$$c_0 = 0.1, \quad c_1 = 1.5, \quad c_2 = 0.7, \quad c_3 = 0.5, \quad \Delta_{\max} = 150, \quad \Delta_{\min} = \Delta_0 = 20, \quad B_0 = I, \quad \alpha_0 = 0.$$

The convergence criterion

$$\|g_k\| \leq 10^{-6} \quad \text{or} \quad f(x_{k-1}) - f(x_k) \leq 10^{-6} \max\{0.1, |f(x_{k-1})|\}$$

is used for the termination test; that is, when one of the two conditions is satisfied, computation stop. We also set a maximum iteration number, 500, to terminate calculation when this number is reached. The following four functions from [9] are presented:

1. Box three-dimensional function

$$f(x) = \sum_{j=1}^3 f_j(x)^2,$$

where $f_j(x) = \exp[-t_j x_1] - \exp[-t_j x_2] - x_3(\exp[-t_j] - \exp[-10t_j])$ and $t_j = (0.1)^j$.

2. Penalty function

$$f(x) = \sum_{i=1}^3 10^{-5}(x_i - 1)^2 + \left[\left(\sum_{j=1}^4 x_j^2 \right) - \frac{1}{4} \right]^2.$$

3. Trigonometric function

$$f(x) = \sum_{j=1}^5 \left[5 - \sum_{i=1}^5 \cos x_i + j(1 - \cos x_j) - \sin x_j \right]^2.$$

4. Kowalik and Osborne function

$$f(x) = \sum_{j=1}^{11} \left(y_j - \frac{x_1(u_j^2 + u_j x_2)}{u_j^2 + u_j x_3 + x_4} \right)^2,$$

where y_j and u_j are given in Table 1.

Algorithm NCTR is used to solve the unconstrained optimization problems (1.1) with the objective functions defined as above, respectively. 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.

Table 1
 u_j and y_j

j	y_j	u_j	j	y_j	u_j
1	0.1957	4.0000	7	0.0456	0.1250
2	0.1947	2.0000	8	0.0342	0.1000
3	0.1753	1.0000	9	0.0323	0.0833
4	0.1600	0.5000	10	0.235	0.0714
5	0.0844	0.2500	11	0.0246	0.0625
6	0.0627	0.1670			

Table 2
Results of NCTR

Pro.	M	Residual	Itr
1	0	8.564783×10^{-7}	17
	2	2.337964×10^{-8}	14
	4	5.867754×10^{-7}	8
	6	7.365741×10^{-7}	14
	8	3.899977×10^{-7}	35
	10	8.666745×10^{-7}	84
	12–14	Failed	Failed
2	0	3.75210×10^{-7}	9
	2	7.857641×10^{-6}	13
	4–14	Failed	Failed
3	0	2.422347×10^{-10}	42
	2	7.498763×10^{-9}	14
	4	1.772999×10^{-7}	34
	6	8.397511×10^{-7}	54
	8–14	Failed	Failed
4	0	4.656321×10^{-8}	67
	2	8.837662×10^{-7}	38
	4	8.561476×10^{-7}	84
	6	7.786666×10^{-7}	91
	8	1.873443×10^{-7}	88
	10	9.877541×10^{-7}	93
	12	7.114456×10^{-7}	95
	14	5.398762×10^{-7}	112

Table 2 contains the results for these experiments, where Ini, Residual and Itr stand for the initial point, $\|g_k\|$ satisfying the stop rules and the numbers of iterations, respectively. For each problem, the code runs from $M = 0$ –14, where $M = 0$ means the monotone trust-region method of conic model in [7]. Therefore, we have actually computed 32 problems and for every problem we have eight cases (from $M = 0$ to 14). Analyzing the numerical results, we have the following conclusions: for the four problems, Algorithm NCTR is good for most problems; our nonmonotone method is competitive with the monotone method in [7] and for some special optimization problems the performance of the nonmonotone method is better.

Acknowledgments

We are grateful to the editors and referees for their suggestions and comments. The first author thanks Pro. Wu for her helpful suggestions and comments.

References

- [1] K.A. Ariyawansa, Deriving collinear scaling algorithms as extensions of quasi-Newton methods and the local convergence of DFP and BFGS related collinear scaling algorithm, *Math. Programming* 49 (1990) 23–48.
- [2] Z.W. Chen, J.Y. Han, D.C. Xu, A nonmonotone trust region method for nonlinear programming with simple bound constraints, *Appl. Math. Optim.* 43 (2001) 63–85.
- [3] W.C. Davidon, Conic approximation and collinear scaling for optimizers, *SIAM J. Numer. Anal.* 17 (1980) 268–281.
- [4] S. Di, W. Sun, Trust region method for conic model to solve unconstrained optimization problems, *Optim. Meth. Software* 6 (1996) 237–263.
- [5] R. Fletcher, *Practical Method of Optimization, Unconstrained Optimization*, vol. 1, Wiley, New York, 1980.
- [6] L. Grippo, F. Lampariello, S. Lucidi, A nonmonotone line search technique for Newton's Method, *SIAM J. Numer. Anal.* 23 (1986) 707–716.
- [7] Y. Ji, S.J. Qu, Y.J. Wang, H.M. Li, A conic trust-region method for optimization with nonlinear equality and inequality 4 constrains via active-set strategy, *Appl. Math. Comput.* 183 (2006) 217–231.
- [8] J.T. Mo, K.C. Zhang, Z.X. Wei, A nonmonotone trust region method for unconstrained optimization, *Appl. Math. Comput.* 171 (2005) 371–384.
- [9] J.J. Moré, B.S. Garbow, K.E. Hillstorn, Testing unconstrained optimization software, *ACM Trans. Math. Software* 7 (1981) 17–41.
- [10] Q. Ni, S. Hu, A new derivative free algorithm based on conic interpolation model, Technical Report, Faculty of Science, Nanjing University of Aeronautics and Astronautics, 2001.
- [11] J. Nocedal, Y. Yuan, Combining trust region and line search techniques, Report NAM 07, Department of EECS, Northwestern University, 1991.
- [12] M.J.D. Powell, On the global convergence of trust region algorithms for unconstrained optimization, *Math. Programming* 29 (1984) 297–303.
- [13] G.A. Schultz, R.B. Schnabel, R.H. Byrd, A family of trust region based algorithms for unconstrained minimization with strong global convergence, *SIAM J. Numer. Anal.* 22 (1985) 47–67.
- [14] D.C. Sorensen, The q -superlinear convergence of a collinear scaling algorithm for unconstrained optimization, *SIAM J. Numer. Anal.* 17 (1980) 84–114.
- [15] Y. Yuan, On the convergence of trust region algorithms, *Math. Numer. Sinica* 16 (1996) 333–346.
- [16] Y. Yuan, W. Sun, *Optimization Theory and Methods*, Science Press, Beijing, China, 1997.
- [17] J.L. Zhang, X.S. Zhang, A nonmonotone adaptive trust region method and its convergence, *Comput. Math. Appl.* 45 (2003) 1469–1477.