Available online at http://scik.org
J. Math. Comput. Sci. 2 (2012), No. 6, 1874-1893
ISSN: 1927-5307

NON-MONOTONE ADAPTIVE FILTER CONIC TRUST REGION METHOD FOR UNCONSTRAINED OPTIMIZATION

LIJUAN ZHAO*

School of Mathematical Sciences, Jiangsu Key Laboratory for NSLSCS, Nanjing Normal University, Nanjing 210023, China

Abstract. In this paper, a new non-monotone trust region method based on new conic model is proposed for unconstrained optimization, we adjust the trust region radius by the method proposed by Hei (2003). We use filter technique to increase the chance of accepting the trial point. Global convergence and Q-superlinear convergence of the method are established. Numerical results reported to show the efficiency of the new method.

Keywords: new conic model, non-monotone, self-adaptive, filter, trust region, unconstrained optimization.

2000 AMS Subject Classification: 47H17; 47H05; 47H09

1. Introduction

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

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

*Corresponding author

*This work was supported by the Foundation for Innovative Program of Jiangsu Province (grant No. CXLX12_0387).

Received September 8, 2012

where $f(x) : R^n \to R$ is continuously differentiable. We adopt the following notations throughout the paper.

(1): The notation $\|\cdot\|$ denotes the Euclidean norm on R^n;

(2): $g(x) \in R^n$ is the gradient of the objective function $f(x)$ evaluated at x, and $B(x) \in R^{n\times n}$ is the Hessian or it's approximation;

(3): Suppose $\{x_k\}$ is a sequence of points generated by an algorithm, we denote $f_k \stackrel{def}{=} f(x_k)$, $g_k = g(x_k) \stackrel{def}{=} \nabla f(x_k)$ and $B_k \stackrel{def}{=} B(x_k)$.

Trust region method is a powerful method for solving problem (1), its main idea is solving the following trust region subproblem,

$$\min_d \quad \phi_k(d) = f_k + g_k^T d + \frac{1}{2} d^T B_k d \tag{2}$$

$$\text{s.t.} \quad \|d\| \leq \Delta_k, \tag{3}$$

where $\Delta_k > 0$ is a trust region radius. We often use a merit function to test whether a trial step is accepted or not. Because trust region method has good theory and convergence properties, many authors have studied it since 1970's (see [1, 2, 6, 10, 11, 14, 15, 20, 21, 22, 23, 25, 27, 30] and the references therein). It's worth mentioning that the book of Conn, Gould and Toint (see [1]) is an excellent and comprehensive one on trust region method. Non-monotone technique combining trust region has been studied by many authors (see [2, 6, 11, 22, 24, 26, 29]). Non-monotone technique was first proposed by Grippo, Lampariello and Lucidi (see [8]) can enhance the possibility of finding a global minimizer. Furthermore, it can improve the rate of convergence in case where a monotone technique is used to creep along the bottom of a narrow curved valley. But it still has the following drawbacks. First, a good function value generated in any iteration is essentially discarded due to the max-value choice in traditional non-monotone. Second, in some cases, the numerical performance is dependent on the choice of M (see [8, 17, 16]). We spend much time on computation, which affects the computational efficiency. Furthermore, it has been pointed out by Dai (see [3]) that although an iterative method generates R-linearly convergent iterations for a strongly convex function, the iterative may not satisfy the Wolfe condition for sufficiently large k, for any fixed bound M on the memory.

In order to increase the freedom of non-monotone line search technique, Zhang and Hager (see [28]) proposed a new non-monotone line search technique, i.e.,

$$\begin{cases} f(x_k + \alpha_k d_k) \le C_k + \beta_1 \alpha_k g_k^T d_k, \\ g(x_k + \alpha_k d_k)^T d_k \ge \sigma g_k^T d_k, \end{cases}$$

where

$$C_k = \begin{cases} f(x_k), & \text{if } \; k = 0; \\ (\eta_{k-1} Q_{k-1} C_{k-1} + f(x_k))/Q_k, & \text{if } \; k \ge 1. \end{cases} \tag{4}$$

$$Q_k = \begin{cases} 1, & \text{if } \; k = 0; \\ \eta_{k-1} Q_{k-1} + 1, & \text{if } \; k \ge 1. \end{cases} \tag{5}$$

here, $0 < \beta_1 < \sigma < 1$ are two constants, and $\eta_{k-1} \in [\eta_{min}, \eta_{max}]$, $\eta_{min} \in [0, 1)$ and $\eta_{max} \in [\eta_{min}, 1]$ are two given constants.

Conic model function was first studied by Davidon and Sorensen (see [5, 18]) in the following form,

$$\min_s \quad \varphi_k(s) = \frac{g_k^T s}{1 - h_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1 - h_k^T s)^2}.$$

Due to trust region method has good convergence properties, Di and Sun (see [4]) proposed the following conic trust region subproblem,

$$\min_s \quad \varphi_k(s) = \frac{g_k^T s}{1 - h_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1 - h_k^T s)^2} \tag{6}$$

$$\text{s.t.} \quad \|s\| \le \Delta_k, \tag{7}$$

where $\varphi_k(s)$ is an approximation to $f(x_k + s) - f(x_k)$, $g_k = \nabla f(x_k)$ is the gradient of the objective function $f(x)$ at current iterate x_k, $B_k \in R^{n \times n}$ is the Hessian matrix of $f(x)$ or it's approximation, and $h_k \in R^n$ is a horizontal vector.

If $h_k = 0$, the conic model is reduced to the quadratic model, so the conic model is a generalization of the quadratic model. They have following advantages. First, if the objective function has strong non-quadratic behavior or it's curvature changes severely, the quadratic model method often produces a poor prediction of the minimizer of the function. In this case, the conic model approximates the objective function value better than the quadratic model, 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 gradient values at current and previous points. Using the conic model may improve the performance of the algorithms. Third, the initial and limited numerical results show that the conic model method gives an improvement over the quadratic model method. Finally, the conic model method has similar global and local convergence properties as the quadratic model method.

Sheng (see [19]) pointed out that if B_k is positive definite and $1 - h_k^T B_k^{-1} g_k \neq 0$, then the strict minimizer point of the conic model (6) is

$$s_k = -\frac{B_k^{-1} g_k}{1 - h_k^T B_k^{-1} g_k}.$$

Ni (see [15]) proposed a new conic trust region subproblem for the case where the conic model function may be unbounded, and gave sufficient and necessary conditions, which is useful for the continuous research of conic model. Ni (see [15]) proposed the following new conic trust region subproblem,

$$\min_s \quad \varphi_k(s) = \frac{g_k^T s}{1 - h_k^T s} + \frac{1}{2} \frac{s^T B_k s}{(1 - h_k^T s)^2} \tag{8}$$

$$\text{s.t.} \quad \|s\| \leq \Delta_k \tag{9}$$

$$|1 - h_k^T s| \geq \epsilon_0, \tag{10}$$

and in [15] they divided the problem (8)-(10) into three subproblems to consider and analyzed the optimality condition for each subproblem.

Hei (see [9]) proposed an adaptive method for adjusting the trust region radius, where he introduced an R-function, which adjusts the trust region radius by $\Delta_{k+1} = R_{c_2}(r_k)\|s_k\|$, where $R_{c_2}(r_k)$ is a R-function.

Filter technique was first introduced by Fletcher and Leyffer (see [7]) as a way to globalize SQP and SLP without using any merit function that would require a troublesome parameter to be provided by user for weighting the relative merits of improving feasibility and optimality. Filter technique introduces a function which aggregates constraint violation, and then deals with the resulting bi-objective problem. A step is accepted if it either reduces the objective function or the constraint violation. Han, Sun and Han (see [10]) proposed an adaptive conic trust region method, if the ratio between the predicted reduction and the actual reduction approximates 1, the method turns to the quasi-Newton method, otherwise it still is a conic model method.

This paper is organized as follows. In Section 2, we describe a new non-monotone adaptive filter trust region method based on new conic model, and in Section 3, the global convergence and the Q-superlinear convergence are established under some mild conditions. The numerical results of a computational experiment performed on a set of standard test problems are reported in Section 4. Finally, we give some concluding remarks in Section 5.

2. Motivation and New Algorithm

In this paper, we propose a non-monotone adaptive filter trust region method based on new conic model. In order to increase the possibility of accepting the trial point, we add filter technique into the algorithm. When the trial point isn't accepted by trust region, we judge whether it is accepted by filter or not. If it is accepted by filter, we add its gradient to filter, and remove the pair whose gradient dominates the other. So we reduce the computational time and increase the chance of accepting the trial point.

2.1. Ideal Trust Region and R-Function

Now, let us reconsider the idea of trust region method. At current iterate x_k, if the trial step s_k is successful (include it is accepted by filter) and the ratio is satisfactory, we accept the trial step and enlarge the trust region radius. On the contrary, s_k is rejected and Δ_k is shrunk. ρ_k reflects the extent to which we are satisfied with the solution of the subproblem (8)-(10), or to say, the extent to which the model function approximates the objective function $f(x)$.

Considering two extreme cases. The first case is when ρ_k is $+\infty$, which means the computed step s_k is very successful, then we enlarge the trust region radius Δ_k greatly, even to $+\infty$. The second case is when ρ_k is $-\infty$, which implies the trial step s_k is so bad that the objective function increases rapidly, then we reduce the trust region radius Δ_k to a small value, even to 0. The above two ideal cases, which was called *ideal trust region*, inspires us to study the following type of function, named as R-function (see [9]).

Definition 2.1. *Any one-dimensional function $R_{c_2}(t)$, which is defined in $R = (-\infty, +\infty)$ with parameter $c_2 \in (0, 1)$ is an R-function if and only if it satisfies,*

(1). $R_{c_2}(t)$ is non-decreasing in $(-\infty, +\infty)$.

(2).

$$\lim_{t \to -\infty} R_{c_2}(t) = \beta,$$

where $\beta \in [0, 1)$ is a small constant.

(3).

$$R_{c_2}(t) \leq 1 - \gamma_1,$$

for all $t < c_2$, where $\gamma_1 \in (0, 1 - \beta)$ is a constant.

(4).

$$R_{c_2}(c_2) = 1 + \gamma_2,$$

where $\gamma_2 \in (0, +\infty)$ is a constant.

(5).

$$\lim_{t \to +\infty} R_{c_2}(t) = M,$$

where $M \in (1 + \gamma_2, +\infty)$ is a constant.

2.2. The Multi-dimensional Filter Technique

Traditional trust region algorithm evaluates the objective function at the trial point, if the ratio between the actual reduction and the predicted reduction is satisfactory, we accept the trial point $x_k^+ = x_k + s_k$ as the next iterate point. Otherwise, we reject the trial point and shrink the trust region radius. Here, if the trial point isn't accepted by trust region, we test whether it is accepted by filter or not. If it is accepted by filter, we accept it too.

Definition 2.2. *A pair $(f^{(k)}, h^{(k)})$ is said to dominate another pair $(f^{(l)}, h^{(l)})$ if and only if both $f^{(k)} \leq f^{(l)}$ and $h^{(k)} \leq h^{(l)}$. A filter $\mathcal{F}$ is a list of pairs $(f^{(l)}, h^{(l)})$ such that no pair dominates any other.*

Definition 2.3. *A trial point x_k^+ is acceptable for the filter $\mathcal{F}$ if and only if*

$$\forall \overline{g}_l \in \mathcal{F}, \quad \exists j \in \{1, 2, \cdots, n\}, \quad s.t. |\overline{g}_j(x_k^+)| \leq |\overline{g}_j(x_l)| - \gamma_g \|\overline{g}_l\|,$$

where $\gamma_g \in (0, \frac{1}{\sqrt{n}})$ is a positive constant and $\overline{g}_{l,j} = \overline{g}_j(x_l)$.

If the trial point isn't accepted by trust region, we see whether it is accepted by filter $\mathcal{F}$ or not. So by using filter technique, we increase the chance of accepting the trial point.

2.3. A new Non-monotone Adaptive Filter Trust Region algorithm for new Conic model {NAFCTR}

Algorithm 2.3 {NAFCTR}

Step 0: **Initialization.**

Given $0 \leq \eta_{min} \leq \eta_{max} \leq 1$, $x_0 \in R^n$, $h_0 \in R^n$, $\Delta_{max} > \Delta_0 > 0$, $Q_0 = 1$, $B_0 \in R^{n\times n}$ is a positive definite matrix. Set $\epsilon_0 \in (0,1)$, $\eta \in (0,1)$, $c_1 = 1, \epsilon > 0$, $\beta \in (0,1)$, $\gamma_g \in (0, \frac{1}{\sqrt{n}})$, $\gamma_1 \in (0, 1-\beta)$, $\gamma_2 > 0$, $M > 1 + \gamma_2$, $0 < c_2 < 1$, $k = 0$. Compute $g_0 = g(x_0)$, $f(x_0), C_0 = f(x_0)$. Initialize the filter $\mathcal{F}$ to be an empty set.

Step 1: **Test for termination.**

If $\|g_k\| \leq \epsilon$, then stop, and x_k is an approximate optimal solution; otherwise go to Step 2.

Step 2: **Solve the subproblem.**

Solve the conic trust region subproblem (8)-(10) for s_k.

Step 3: **Computation.**

Set $x_k^+ = x_k + s_k$. Compute $f(x_k^+)$, set

$$Ared_k(s_k) = C_k - f(x_k^+)$$

$$Pred_k(s_k) = \varphi_k(0) - \varphi_k(s_k)$$

$$r_k = Ared_k(s_k)/Pred_k(s_k).$$

Step 4: **Determine the acceptance of the trial point.**

If $r_k \geq \eta$, set $x_{k+1} = x_k^+$, go to Step 5;

else compute $g_k^+ = g(x_k^+)$. If x_k^+ is accepted by filter $\mathcal{F}$, we accept s_k, set $x_{k+1} = x_k^+$, add g_k^+ into filter $\mathcal{F}$, go to Step 5;

else set $c_1 := \frac{1}{4}c_1$, $\Delta_{k+1} = c_1\Delta_k$, go to Step 2.

Step 5: **Update the trust region radius.**

Update the trust region radius by $\Delta_{k+1} = R_{c_2}(r_k)\Delta_k$.

Step 6: **Update the parameters.**

Generate B_{k+1}, h_{k+1}, choose $\eta_k \in [\eta_{min}, \eta_{max}]$, set $Q_{k+1} = \eta_k Q_k + 1$, $C_{k+1} = (\eta_k Q_k C_k + f(x_{k+1}))/Q_{k+1}$, set $k := k+1$, go to Step 1.

Remark 2.3.

(1): In Algorithm 2.3, the subproblem (8)-(10) is solved by Ni and Lu's dogleg method (see [12]);

(2): The Step 2 - Step 4 - Step 2 is called an inner cycle;

(3): From Algorithm 2.3, we can see our iterations are divided into two parts: filter iterations and trust region iterations. The first one contains the iterations which are added into filter $\mathcal{F}$ and the iterations which are acceptable for filter $\mathcal{F}$ but not be added into filter $\mathcal{F}$. Both of them are successful iterations. It plays an important role in the proof of convergence.

(4): For the convenience of convergence, we assume that there exists a constant $\delta \in (0,1)$ such that

$$\|h_k\|\Delta_k < \delta, \quad \forall k. \tag{11}$$

(5): In Algorithm 2.3, the horizontal vector h_k is updated by

$$h_k = \min\left(\frac{\beta_k - 1}{g_{k-1}^T s_{k-1}} g_{k-1}, \overline{\alpha}\right), \tag{12}$$

where

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

$$\beta_k = \begin{cases} \frac{(f_{k-1}-f_k)+\sqrt{\rho_k}}{-g_{k-1}^T s_{k-1}}, & \text{if} \quad \rho_k > 0; \\ 1, & \text{otherwise.} \end{cases}$$

3. Convergence Analysis

In this section, we give the global convergence of Algorithm 2.3, considering the following conic trust region subproblem,

$$\min_s \quad \varphi_k(s) = \frac{g_k^T s}{1 - h_k^T s} + \frac{1}{2}\frac{s^T B_k s}{(1 - h_k^T s)^2} \tag{13}$$

$$\text{s.t.} \quad \|s\| \le \Delta_k \tag{14}$$

$$|1 - h_k^T s| \ge \epsilon_0. \tag{15}$$

First, we give some assumptions as follows.

A1. The objective function $f(x)$ is twice continuously differentiable and bounded below.

A2. The level set $L(x_0) = \{x \in R^n | f(x) \le f(x_0)\}$ is compact.

A3. The gradient $g(x)$ is Lipschitz continuous with Lipschitz constant L.

Note that Assumptions A1 and A2 imply that there exist two positive constants G and M_1 such that

$$\|g(x)\| \le G, \quad \|\nabla^2 f(x)\| \le M_1, \quad \forall x \in L(x_0). \tag{16}$$

Suppose $K = 1 + \max_i \|B_i\|$.

For the purpose of convergence analysis, we define

$$A = \{k | g_k \text{ is added to filter} \mathcal{F}\},$$

the set of *filter iterations*;

$$S = \{k | x_{k+1} = x_k + s_k\},$$

the set of *successful iterations*;

$$F = \{k | g_k \text{ is accepted by filter} \mathcal{F}\},$$

the set of *filter accepted iterations*;

$$T = \{k | r_k \geq \eta\},$$

the set of *sufficient descent iterations.*

Lemma 3.1. *An R-function $R_{c_2}(t)$ (where $c_2 \in (0,1)$) satisfies:*

$$0 < \beta \leq R_{c_2}(t) \leq 1 - \gamma_1 < 1, \quad \forall t \in (-\infty, c_2),$$

$$1 < 1 + \gamma_2 \leq R_{c_2}(t) \leq M < +\infty, \quad \forall t \in [c_2, +\infty).$$

Lemma 3.2. *Suppose the sequence $\{x_k\}$ is generated by Algorithm 2.3, then for all k, the following inequality holds,*

$$f_{k+1} \leq C_{k+1} \leq C_k. \tag{17}$$

Proof. The proof is similar to Lemma 3.1 (see [28]), so we omit the proof here.

Lemma 3.3. *Suppose that s_k is the solution of subproblem (13)-(15), then we have*

$$Pred_k(s_k) \geq \frac{1}{2}\|g_k\| \min\{\frac{\Delta_k}{1 + \Delta_k\|h_k\|}, \frac{\|g_k\|}{\|B_k\|}, \frac{1 - \epsilon_0}{\|h_k\|\epsilon_0}\}.$$

Proof. The proof is similar to Theorem 4.1 (see [12]), so we omit the proof here.

Lemma 3.4. *If Assumptions A1 and A2 hold, then we have*

$$|[f(x_k) - f(x_k + s_k)] - [\varphi_k(0) - \varphi_k(s_k)]| \leq M_2\|s_k\|^2.$$

where $M_2 \overset{def}{=} \frac{G\overline{\alpha}}{\epsilon_0} + \frac{M_1}{2} + \frac{K}{2\epsilon_0^2}$.

Proof. From (12), (14), (15), (16) and Taylor's expansion, we have

$$\begin{aligned}
& |[f(x_k) - f(x_k + s_k)] - [\varphi_k(0) - \varphi_k(s_k)]| \\
= & |g_k^T s_k + \frac{1}{2} s_k^T \nabla^2 f(x_k + \theta s_k) s_k - \frac{g_k^T s_k}{1 - h_k^T s_k} - \frac{1}{2} \frac{s_k^T B_k s_k}{(1 - h_k^T s_k)^2}| \\
= & |\frac{g_k^T s_k h_k^T s_k}{1 - h_k^T s_k} - \frac{1}{2} s_k^T \nabla^2 f(x_k + \theta s_k) s_k + \frac{1}{2} \frac{s_k^T B_k s_k}{(1 - h_k^T s_k)^2}| \\
\leq & (\frac{G\overline{\alpha}}{\epsilon_0} + \frac{M_1}{2} + \frac{K}{2\epsilon_0^2}) \Delta_k^2 \\
\stackrel{def}{=} & M_2 \Delta_k^2,
\end{aligned}$$

where $\theta \in (0, 1)$ is a constant.

Lemma 3.5. *Suppose Assumptions A1 and A2 hold,* $\|g_k\| \neq 0$ *and* $\Delta_k = \min\{\Delta_{max}, \frac{(1-c_2)\epsilon}{2M_2(1+\delta)}\}$, *then*

(1): *the Algorithm 2.3 is well-defined. That is to say, the Algorithm 2.3 doesn't cycle infinitely in the inner cycle.*

(2): $r_k \geq c_2$, $\Delta_{k+1} \geq \Delta_k$.

Proof. Suppose Algorithm 2.3 cycles infinitely between Step 2 - Step 4 - Step 2, then $\lim_{k\to\infty} \Delta_k = 0$. Let $k(i)$ be the cycle index of the current iterate x_k, then $r_{k(i)} \leq c_2$, and

(18) $$\|g_{k(i)}\| > \epsilon \quad \text{and} \quad r_{k(i)} = \frac{C_{k(i)} - f(x_{k(i)} + s_{k(i)})}{Pred_{k(i)}(s_{k(i)})} \leq c_2.$$

From (11), Lemma 3.3 and Lemma 3.4 we know that

$$\begin{aligned}
& |\frac{f(x_{k(i)}) - f(x_{k(i)} + s_{k(i)})}{Pred_{k(i)}(s_{k(i)})} - 1| \\
= & |\frac{[f(x_{k(i)}) - f(x_{k(i)} + s_{k(i)})] - [\varphi_{k(i)}(0) - \varphi_{k(i)}(s_{k(i)})]}{Pred_{k(i)}(s_{k(i)})}| \\
\leq & \frac{M_2 \Delta_{k(i)}^2}{\frac{1}{2}\|g_{k(i)}\| \min\{\frac{\Delta_{k(i)}}{1+\Delta_{k(i)}\|h_{k(i)}\|}, \frac{\|g_{k(i)}\|}{\|B_{k(i)}\|}, \frac{1-\epsilon_0}{\|h_{k(i)}\|\epsilon_0}\}} \\
\leq & \frac{M_2 \Delta_{k(i)}^2}{\frac{1}{2}\|g_{k(i)}\| \min\{\frac{\Delta_{k(i)}}{1+\delta}, \frac{\epsilon}{K}, \frac{1-\epsilon_0}{\overline{\alpha}\epsilon_0}\}} \\
< & \frac{2M_2(1+\delta)\Delta_{k(i)}}{\epsilon} \\
< & 1 - c_2.
\end{aligned}$$

So we have

$$\frac{f(x_{k(i)}) - f(x_{k(i)} + s_{k(i)})}{Pred_{k(i)}(s_{k(i)})} > c_2.$$

From (17) we have

$$\begin{aligned} r_{k(i)} &= \frac{C_{k(i)} - f(x_{k(i)} + s_{k(i)})}{Pred_{k(i)}(s_{k(i)})} \\ &\geq \frac{f(x_{k(i)}) - f(x_{k(i)} + s_{k(i)})}{Pred_{k(i)}(s_{k(i)})} \\ &> c_2, \end{aligned} \tag{19}$$

which contradicts with (18). So we have

(1): the Algorithm 2.3 doesn't cycle infinitely in the inner cycle.

(2): $r_k \geq c_2$, $\Delta_{k+1} \geq \Delta_k$. $\square$

Lemma 3.6. *If Assumptions A1 and A2 hold, and there exists a positive constant ϵ such that for all k, the inequality $\|g_k\| \geq \epsilon$ holds, then there must exist a constant Δ_{bld} such that $\Delta_k \geq \Delta_{bld}$, where $\Delta_{bld} \stackrel{def}{=} \beta \min\{\Delta_{max}, \frac{(1-c_2)\epsilon}{2M_2(1+\delta)}\}$.*

Proof. Suppose that k is the first index satisfying

$$\Delta_{k+1} \leq \beta \min\{\Delta_{max}, \frac{(1-c_2)\epsilon}{2M_2(1+\delta)}\} \stackrel{def}{=} \beta\delta_0. \tag{20}$$

From Lemma 3.1 we have $\beta\Delta_k \leq \Delta_{k+1}$, so

$$\Delta_k \leq \delta_0 = \min\{\Delta_{max}, \frac{(1-c_2)\epsilon}{2M_2(1+\delta)}\}. \tag{21}$$

Due to (21), $\|g_k\| \geq \epsilon$ and Lemma 3.5, we have $\Delta_{k+1} \geq \Delta_k$, so

$$\Delta_k \leq \beta \min\{\Delta_{max}, \frac{(1-c_2)\epsilon}{2M_2(1+\delta)}\} \stackrel{def}{=} \beta\delta_0,$$

which contradicts the fact that k is the first index such that (20) holds. Set $\Delta_{lbd} \stackrel{def}{=} \beta \min\{\Delta_{max}, \frac{(1-c_2)\epsilon}{2M_2(1+\delta)}\}$. This completes the proof.

Lemma 3.7. *Suppose Assumptions A1-A3 hold, and that $|A| = |S| = +\infty$, there exists a positive constant ϵ such that the inequality $\|g_k\| \geq \epsilon$ holds for all k. Then there only have finite sufficient descent iteration, i.e., $|T| < \infty$.*

Proof. Suppose that there exists infinite sufficient descent iteration, i.e., $|T| = \infty$, set $T = \{k_i\}$, then for all $k_i \in T$, we have $r_{k_i} \geq \eta$. From (11), (12), Lemma 3.3 and Lemma 3.6 we have

$$\begin{aligned} C_{k_i} - f(x_{k_i+1}) &\geq \eta[\varphi_{k_i}(0) - \varphi_{k_i}(s_{k_i})] \\ &\geq \frac{\eta}{2}\|g_{k_i}\| \min\{\frac{\Delta_{k_i}}{1+\Delta_{k_i}\|h_{k_i}\|}, \frac{\|g_{k_i}\|}{\|B_{k_i}\|}, \frac{1-\epsilon_0}{\|h_{k_i}\|\epsilon_0}\} \\ &\geq \frac{\eta\epsilon}{2}\min\{\frac{\Delta_{lbd}}{1+\delta}, \frac{\epsilon}{K}, \frac{1-\epsilon_0}{\overline{\alpha}\epsilon_0}\}, \end{aligned}$$

then

$$f(x_{k_i+1}) \leq C_{k_i} - \frac{\eta\epsilon}{2}\min\{\frac{\Delta_{lbd}}{1+\delta}, \frac{\epsilon}{K}, \frac{1-\epsilon_0}{\overline{\alpha}\epsilon_0}\}. \tag{22}$$

From (5) we have $Q_k \leq k+1$, combining (4) we get

$$\begin{aligned} C_{k_i+1} &= \frac{\eta_{k_i}Q_{k_i}C_{k_i} + f(x_{k_i+1})}{Q_{k_i+1}} \\ &\leq \frac{\eta_{k_i}Q_{k_i}C_{k_i} + C_{k_i} - \frac{\eta\epsilon}{2}\min\{\frac{\Delta_{lbd}}{1+\delta}, \frac{\epsilon}{K}, \frac{1-\epsilon_0}{\overline{\alpha}\epsilon_0}\}}{Q_{k_i+1}} \\ &= C_{k_i} - \frac{\eta\epsilon\min\{\frac{\Delta_{lbd}}{1+\delta}, \frac{\epsilon}{K}, \frac{1-\epsilon_0}{\overline{\alpha}\epsilon_0}\}}{2(k_i+2)}, \end{aligned}$$

so

$$C_{k_i} - C_{k_i+1} \geq \frac{\eta\epsilon\min\{\frac{\Delta_{lbd}}{1+\delta}, \frac{\epsilon}{K}, \frac{1-\epsilon_0}{\overline{\alpha}\epsilon_0}\}}{2(k_i+2)}.$$

Thus,

$$\begin{aligned} C_0 - C_{k_i+1} &\geq \sum_{j=0}^{k_i}(C_j - C_{j+1}) \\ &\geq \sum_{j=0}^{k_i}\frac{\eta\epsilon\min\{\frac{\Delta_{lbd}}{1+\delta}, \frac{\epsilon}{K}, \frac{1-\epsilon_0}{\overline{\alpha}\epsilon_0}\}}{2(j+2)}. \end{aligned}$$

From Assumption A1 we know that f_k is bounded below, without loss of generality, we assume the lower bound is f_{lbd}, from Lemma 3.2 we have $C_k \geq f_k \geq f_{lbd}$. So

$$f(x_0) - f_{lbd} = C_0 - f_{lbd} \geq C_0 - C_{k_i+1} \geq \sum_{j=0}^{k_i}\frac{\eta\epsilon\min\{\frac{\Delta_{lbd}}{1+\delta}, \frac{\epsilon}{K}, \frac{1-\epsilon_0}{\overline{\alpha}\epsilon_0}\}}{2(j+2)},$$

thus

$$\sum_{j=0}^{k_i} \frac{1}{j+2} \leq \frac{2(f(x_0) - f_{lbd})}{\eta\epsilon \min\{\frac{\Delta_{lbd}}{1+\delta}, \frac{\epsilon}{K}, \frac{1-\epsilon_0}{\overline{\alpha}\epsilon_9}\}},$$

which contradicts with the fact that the series $\sum_{j=0}^{\infty} \frac{1}{j+2}$ diverges when $k_i \to \infty$. This completes the proof.

Theorem 3.8. *Suppose Assumptions A1 and A2 hold, and $|T| = +\infty$, then for sufficiently large k, we have $x_k = x^*$, where x^* is a first order critical point.*

Proof. Suppose that k_0 is the last index of successful iteration, then $x^* = x_{k_0+1} = x_{k_0+j}$, and

$$r_{k_0+j} < \eta, \quad \forall j > 0.$$

Due to the strategy of updating the trust region radius and Lemma 3.1 we have

$$\begin{aligned} \Delta_{k_0+j+1} &= R_{c_2}(r_{k_0+j})\Delta_{k_0+j} \\ &\leq (1-\gamma_1)\Delta_{k_0+j}, \end{aligned}$$

so

$$\lim_{k\to\infty} \Delta_k = 0. \tag{23}$$

If $\|g_{k_0+j}\| > 0$, from Lemma 3.6, we know Δ_k is bounded below, which contradicts with (23). So $\|g_{k_0+j}\| = 0$ for sufficiently large j, that is to say, x^* is a first order critical point. This completes the proof.

Theorem 3.9. *Suppose Assumptions A1-A3 hold, and $|A| = +\infty, |T| < +\infty$, then*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{24}$$

Proof. $|A| = \infty$ indicates that for sufficiently large $k_i \in A$, there exists an index $j \in \{1, 2, \cdots, n\}$ satisfying

$$|g_j(x_{k_i})| - |g_j(x_{k_{i-1}})| \leq -\gamma_g \|g_{k_{i-1}}\|. \tag{25}$$

From Assumptions A2 and A3, we know that $g(x)$ is continuous in the bounded compact set, so there must exist a convergent subsequence of $\{g_j(x_k)\}$, without loss of generality, we assume the convergent subsequence is $\{g_j(x_k)\}$, set limit to (25), we have

$$\liminf_{k\to\infty} \|g_k\| = 0.$$

This completes the proof.

Next, we prove the local convergence of Algorithm 2.3. The following assumptions are needed.

A4. The sequence $\{x_k\}$ generated by Algorithm 2.3 converges to a critical point x^*, i.e.,

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

A5. If

$$\frac{\|B_k^{-1} g_k\|}{1 - h_k^T B_k^{-1} g_k} \leq \Delta_k,$$

then

$$s_k = \frac{-B_k^{-1} g_k}{1 - h_k^T B_k^{-1} g_k}.$$

The local convergence is similar to Theorem 4.2 (see [11]), so we omit the proof here.

Theorem 3.10. *Suppose Assumptions A1-A5 hold, $\nabla^2 f(x^*)$ is positive definite and $\nabla^2 f(x)$ is Lipschitz continuous in the neighborhood of x^*. If*

$$\lim_{k\to\infty} \frac{\|(B_k - \nabla^2 f(x_k))s_k\|}{\|s_k\|} = 0,$$

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

4. Numerical Results

In this section, we test the performance of Algorithm 2.3, denoted by NAFCTR, on a set of standard test problems which appeared in [13]. A MATLAB program is coded to perform the experiments.

In order to compute the optimal solution by Algorithm 2.3, we set

$$s_{k-1} = x_k - x_{k-1},$$

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

$$\beta_k = \begin{cases} \frac{(f_{k-1}-f_k)+\sqrt{\rho_k}}{-g_{k-1}^T s_{k-1}}, & \text{if} \quad \rho_k > 0; \\ 1, & \text{otherwise.} \end{cases}$$

$$h_k = \min\left(\frac{\beta_k - 1}{g_{k-1}^T s_{k-1}} g_{k-1}, \overline{\alpha}\right),$$

$$y_{k-1} = \beta_k g_k - \beta_k^3 g_{k-1},$$

$$B_{k+1} = \begin{cases} B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k y_k}, & \text{if} \quad s_k^T y_k > 0; \\ B_k, & \text{otherwise.} \end{cases}$$

For each test problem, we set

$$R_{c_2}(r_k) = \begin{cases} \frac{2}{\pi}(M - 1 - \gamma_2)\arctan(r_k - c_2) + (1 + \gamma_2), & \text{if} \quad r_k \geq c_2; \\ (1 - \gamma_1 - \beta)(\exp^{r_k - c_2} + \frac{\beta}{1-\gamma_1-\beta}), & \text{otherwise.} \end{cases}$$

The initial trust region radius $\Delta_0 = 1.5$, the filter constant $\gamma_g = \min\{0.001, \frac{1}{2\sqrt{n}}\}$, where n is the dimension of the problem. The initial horizontal vector $h_0 = 0$, $c_1 = 1$, $\epsilon_0 = 0.1$, $\eta = 0.25$, $\Delta_{max} = 10000\Delta_0$, $B_0 = \|f_0\| I$, the non-monotone constant $\eta_k = 0.85$ for all k, the adaptive trust region parameter $\gamma_1 = 0.01$, $\gamma_2 = 0.01$, $\beta = 0.1$, $M = 5$, $c_2 = 0.25$.

For each test problem, the convergence criterion

$$\|g_k\| \leq \epsilon,$$

is used for termination, where $\epsilon = 10^{-4}$. Another stopping criterion is $k \leq k_{max}$, where k is the number of iteration, k_{max} is the max number of iteration, $k_{max} = 5000$.

Table 1 lists the function names in the tests. The numerical results are listed in Table 2. We denote the number of iteration by n_i, the number of function evaluations by n_f, the number of gradient evaluations by n_g, and the final objective function value by f_{min}. The sign $m(n)$ stands for $m \times 10^n$. The sign $'-'$ means that when the iteration reaches 5000, the algorithm fails to reach a minimum.

Table 1. Test Functions

No.	Function Name	No.	Function Name
1	Helical Valley	2	Biggs Exp6
3	Gaussian	4	Powell Badly Scaled
5	Box 3-Dimensional I	6	Variably Dimensioned
7	Watson	8	Penalty I
9	Penalty II	10	Brown and Dennis
11	Gulf	12	Trigonometric
13	Extended Rosenbeock	14	Extended Powell
15	Beal	16	Wood
17	Scaled Cube ($c = 100$)	18	Conic

From Table 2 we can see that the new non-monotone adaptive filter trust region method based on new conic model (abbreviated as NAFCTR) is efficient and robust than traditional conic trust region method (abbreviated as CTR) in 9 test problems, especially for some ill-conditioned or hard-solved problems, the same as CTR in 1 problem, almost the same or a little worse than the trust region method in 8 problems. We find that for most of the problems, our new method performs better than traditional conic trust region method. It should be pointed out that, because we use filter technique, the cost may be a little higher. However, the new method can deal with some ill-conditioned or hard-solved problems, and it is suitable to deal with a lot of practical problems by this method.

5. Conclusions

In this paper, we propose a new non-monotone adaptive filter trust region method based on new conic model for unconstrain optimization problem, the new method is simple to implement. The new method can deal with some ill-conditioned or hard-solved problems, and it has preferable convergence properties. The new algorithm in this paper can be extended to nonlinear constrained optimization problem, which is our next work.

Table 2. Numerical Results

$No.$	CTR				$NAFCTR$			
	n_i	n_f	n_g	f_{min}	n_i	n_f	n_g	f_{min}
1	54	119	56	2.825934(-12)	31	64	33	4.585821(-14)
2	37	79	39	5.655650(-3)	34	70	36	5.655650(-3)
3	14	30	16	1.130391(-8)	18	38	20	1.128871(-8)
4	57	135	59	2.889883(-6)	-	-	-	-
5	25	57	27	6.458912(-14)	38	78	40	5.095572(-9)
6	6	18	8	4.337054(-15)	5	12	7	6.212270(-12)
7	49	109	51	1.570059(-7)	63	128	65	1.561882(-7)
8	5	12	7	9.083185(-6)	5	12	7	9.083498(-6)
9	9	22	11	8.066390(-7)	20	42	22	8.068379(-7)
10	74	167	76	8.582220(+4)	43	88	45	8.582220(+4)
11	49	107	51	2.404750(-11)	1	4	3	3.283500(+1)
12	6	14	8	5.737001(-11)	69	140	71	1.072368(-9)
13	39	89	41	1.613514(-11)	22	46	24	2.766003(-10)
14	52	115	54	5.553763(-8)	27	56	29	3.650872(-10)
15	14	35	16	1.100584(-14)	16	34	18	2.595504(-11)
16	108	232	110	4.285270(-15)	96	194	98	9.883138(-12)
17	33	76	35	4.586391(-13)	67	136	69	9.014771(-14)
18	27	59	29	2.512936(-10)	23	48	25	4.834008(-13)

References

[1] A.R. Conn, N.I.M. Gould and Ph.L. Toint, Trust Region Methods, SIAM, Philadelphia, USA, 2000.

[2] N. Deng, Y. Xiao and F. Zhou, Nonmonotonic trust region algorithm, J. Optimiz. Theory. App, 76 (1993), 259-285.

[3] Y. Dai, On the nonmonotone line search, J. Optimiz. Theory. App, 112 (2002), 315-330.

[4] S. Di and W. Sun, A trust region method for conic model to solve unconstrained optimizations, Optim. Method. Softw, 6 (1996), 237-263.

[5] W.C. Davidon, Conic approximations and collinear scalings for optimizers, SIAM. J. Numer. Anal, 17 (1980), 268-281.

[6] J. Fu and W. Sun, Nonmonotone adaptive trust-region method for unconstrained optimization problems, Appl. Math. Comput, 163 (2005), 489-504.

[7] R. Fletcher and S. Leyffer, Nonlinear programming without a penalty function, Math. Program, 91 (2002), 239-269.

[8] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's method, SIAM. J. Numer. Anal, 23 (1986), 707-716.

[9] L. Hei, A self adaptive trust region algorithm, J. Comput. Math, 21 (2003), 229-236.

[10] Q. Han, W. Sun and J. Han, An adaptive conic trust-region method for unconstrained optimization, Optim. Method. Softw, 20 (2005), 665-677.

[11] Y. Ji, Y.J. Li, K.C. Zhang and X.L. Zhang, A new nonmonotone trust-region method of conic model for solving unconstrained optimization, J. Comput. Appl. Math, 233 (2010), 1746-1754.

[12] X. Lu and Q. Ni, A dogleg method for solving new trust region subproblems of conic model, Acta. Math. Appl. Sin - E, 30 (1997), 1-17.

[13] J.J. Moré, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software, ACM. Math. Software, 7 (1981), 17-41.

[14] J.J. Moré, Recent development in algorithm and software for trust region methods, Math. Program, The State of the Art, Edited by A. Bachem, M. Grotschel and B. Kortz, Springer-Verlag, Berlin, (1983), 258-287, Philadelphia, USA, 2000.

[15] Q. Ni, Optimality conditions for trust region subproblems involving a conic model, SIAM. J. Optimiz, 15 (2005), 828-837.

[16] M. Powell, Convergence properties of a class of minimization algorithms, O. L. Mangasarian, R. Meyer, and S.Robinson, eds, Nonlinear Programming 2, Academic Press, New York, (1975), 1-27.

[17] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problems, SIAM. J. Optimiz, 7 (1997), 26-33.

[18] D.C. Sorensen, The Q-superlinear convergence of a collinear scaling algorithm for unconstrained optimization, SIAM. J. Numer. Anal, 17 (1980), 84-114.

[19] S. Sheng, Interpolation by conic model for unconstrained optimization, Computing, 54 (1995), 83-98.

[20] T. Steihaug, The conjugate gradient method and trust region in large scale optimization, SIAM. J. Numer. Anal, 20 (1983), 626-637.

[21] W. Sun, R. Sampaio and J. Yuan, Trust region methods with quasi-Newton update for nonsmooth least-squares problems, Appl. Math. Comput, 105 (1999), 183-194.

[22] W. Sun, Nonmonotone trust region methods for optimization, Appl. Math. Comput, 156 (2004), 159-174.

[23] W. Sun and D. Xu, A filter-trust region method based on conic model for unconstrained optimization (in Chinese), Sci. Sin. Math, 42 (2012), 1-16.

[24] Ph.L. Toint, An assessment of nonmontone line search techniques for unconstrained optimization, SIAM. J. Sci. Comput, 17 (1996), 725-739.

[25] H. Wu and Q. Ni , A new trust region algorithm with conic model, Numerical Mathematics A Journal of Chinese Universities, 30 (2008), 57-63.

[26] Q. Wang and Z. Huang, A nonmonotone adaptive trust region method based on new conic model, Journal of tai yuan university of science and technology, 31 (2010), 235-238.

[27] Y. Yuan and W. Sun, Optimization Theory and Methods, Science Press, Beijing, 1997.

[28] H. Zhang and W. Hager, A nonmontone line search technique and its application to unconstrained optimization, SIAM. J. Optimiz, 14 (2004), 1043-1056.

[29] X. Zhang and X. Wang, A nonmonotone self-adaptive trust region algorithm with line search based on the new conic model, Journal of Tai Yuan University of Science and Technology, 31 (2010), 1-4.

[30] X. Zhang, Z. Chen and L. Liao, A self-adaptive trust region method for unconstrained optimization, Operations Research Transactions, 5 (2001), 53-62.