

A Retrospective Filter Trust Region Algorithm for Unconstrained Optimization*

Yue Lu, Zhongwen Chen

School of Mathematics Science, Suzhou University, Suzhou, China

E-mail: yueyue403@gmail.com, zwchen@suda.edu.cn

Received June 8, 2010; revised July 18, 2010; accepted July 21, 2010

Abstract

In this paper, we propose a retrospective filter trust region algorithm for unconstrained optimization, which is based on the framework of the retrospective trust region method and associated with the technique of the multi-dimensional filter. The new algorithm gives a good estimation of trust region radius, relaxes the condition of accepting a trial step for the usual trust region methods. Under reasonable assumptions, we analyze the global convergence of the new method and report the preliminary results of numerical tests. We compare the results with those of the basic trust region algorithm, the filter trust region algorithm and the retrospective trust region algorithm, which shows the effectiveness of the new algorithm.

Keywords: Unconstrained Optimization, Retrospective, Trust Region Method, Multi-Dimensional Filter Technique

1. Introduction

Consider the following unconstrained optimization problem

$$\min f(x) \quad (1)$$

where $x \in R^n$, $f: R^n \rightarrow R$ is a twice continuously differentiable function.

The trust region method for unconstrained optimization is first presented by Powell [1], which, in some sense, is equivalent to the Levenberg-Marquardt method which is used to solve the least square problems and which was given by Levenberg [2] and Marquardt [3]. The basic idea of trust region methods works as follows. In the neighborhood of the current iterate (which is called the trust region), we define a model function that approximates the objective function in the trust region and compute a trial step within the trust region for which we obtain a sufficient model decrease. Then we compare the achieved reduction in $f(x)$ to the predicted reduction in the model for the trial step. If the ratio of achieved versus predicted reduction is sufficiently positive, we define our next guess to be the trial point. If this ratio is not sufficiently positive, we decrease the trust region radius in order to make the trust region smaller. Otherwise, we may increase it or possibly keep it unchanged.

Since the trust region method is of the naturalness, the strong convergence and robustness, it has been concerned by many people, such as Powell [1,4,5], Schultz *et al.* [6], Sorensen [7], Moře [8], Yuan [9] and so on. In recent years, the trust region method has been applied to the optimization problems with equality constraints [10], simple bound constraints [11], convex constraints [12] and so on. Many of convergence results have been obtained, which can be seen in [13].

In Fletcher and Leyffer [14] a new technique for globalizing methods for nonlinear programming (NLP) is presented. The idea is referred to as an NLP filter, motivated by the aim of avoiding the need to choose penalty parameters, and considered the relationship between the objective function and the constraint violation in the view of multi-objective optimization. They make the values of the objective function and the constraint violation to be a pair (which is called the filter), construct a sophisticated filter mechanism by comparing the relationship between the pairs, and control the algorithm to converge to the stationary point of the problem (1). The results of numerical tests show that the filter methods are very effective. Fletcher *et al.* [14,15], Toint *et al.* [16], Ulbrich *et al.* [17], Wächter *et al.* [18,19] have combined the idea with SQP method, trust region method, interior-point method, line search methods, respectively, and obtained some interesting results about the filter method.

*This work was supported by Chinese NSF Grant 60873116.

Fletcher, Leyffer and Toint [20] review the ideas above and mention the application of the filter method in practice. In [14], they study the problem of the following form

$$\begin{aligned} \min & f(x), \\ \text{s.t.} & c(x) \leq 0, \end{aligned}$$

where $c: R^n \rightarrow R^m$ is continuously differentiable function. Define the measure of the constraint violation

$$h(c(x)) = \sum_{j=1}^m \max(0, c_j(x)).$$

We use $(f^{(k)}, h^{(k)})$ to denote values of $f(x)$ and $h(c(x))$ evaluated at x_k . Now, we give the following definitions about the filter methods.

Definition 1.1 A pair $(f^{(k)}, h^{(k)})$ obtained on iteration k is said to dominate another pair $(f^{(l)}, h^{(l)})$ if and only if

$$f^{(k)} \leq f^{(l)} \text{ and } h^{(k)} \leq h^{(l)}.$$

Definition 1.2 A filter is a list of pairs $(f^{(l)}, h^{(l)})$ such that no pair dominates any other.

We use F_k to denote the set of iteration indices $j (j \leq k)$ such that $(f^{(j)}, h^{(j)})$ is an entry in the current filter.

Definition 1.3 A pair $(f^{(l)}, h^{(l)})$ is said to be acceptable for inclusion in the filter if it is not dominated by any pair in the filter, that is, for any pair $(f^{(l)}, h^{(l)}) \in F_k$, $(f^{(k)}, h^{(k)})$ satisfies

$$f^{(k)} \leq f^{(l)} \text{ or } h^{(k)} \leq h^{(l)} \quad (2)$$

In order to obtain the global convergence of the algorithm, we should make f, h satisfy the sufficient reduction condition, so we strengthen the acceptable rule (2) as

$$f^{(k)} \leq f^{(l)} - \gamma_h h^{(l)} \text{ or } h^{(k)} \leq (1 - \gamma_h) h^{(l)} \quad (3)$$

where $\gamma_h \in (0, 1)$. When $\gamma_h \in (0, 1)$ is very small, there is negligible difference in practice between (2) and (3).

Definition 1.4 When the pair $(f^{(k)}, h^{(k)})$ is added to the list of pairs in the filter, any pairs in the filter that are dominated by the new pair are removed, that is, we remove the pair $(f^{(l)}, h^{(l)}) \in F_k$ which satisfies

$$f^{(k)} < f^{(l)} \text{ and } h^{(k)} < h^{(l)}.$$

This is called the modification of the filter.

Gould *et al.* [21] and Miao *et al.* [22] applies the filter technique to unconstrained optimization, whose characteristic is to relax the condition of accepting a trial step for the usual trust region method, which improves the effectiveness of the algorithm in some sense. The nonmonotonic algorithm also has the algorithm in some

sense. The nonmonotonic algorithm also has the characteristic [23,24].

Recently, Bastin *et al.* [25] presents a retrospective trust region method for unconstrained optimization. Comparing their algorithm with the basic trust region algorithm, the updating way of the trust region radius is different, and the retrospective ratio

$$\begin{aligned} \tilde{\rho}_{k+1} &= \frac{f(x_{k+1}) - f(x_{k+1} - s_k)}{m_{k+1}(x_{k+1}) - m_{k+1}(x_{k+1} - s_k)} \\ &= \frac{f(x_k) - f(x_k + s_k)}{m_{k+1}(x_k) - m_{k+1}(x_k + s_k)} \end{aligned}$$

is mentioned, where $m_k(x_k + s)$ is the approximated quadratic model of the objective function $f(x)$ at x_k . s_k is a solution of the following trust region subproblem

$$\begin{aligned} \min & m_k(x_k + s), \\ \text{s.t.} & \|s\| \leq \Delta_k. \end{aligned} \quad (4)$$

Δ_k is the trust region radius at the current iterate point. In the basic trust region algorithm, the ratio ρ_k

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}$$

plays the following two roles.

- 1) Determine the trial step to be accepted by the algorithm or not.
- 2) Adjust the trust region radius correspondingly.

In the retrospective trust region method, the two roles are played by the ratio ρ and $\tilde{\rho}$, respectively. In the basic trust region algorithm, the determination of trust region radius is an important and difficult problem. Sartenaer [26] and Zhang *et al.* [27] present the self-adaptive trust region methods and give some discussions about the determination of trust region radius. Bastin *et al.* [25] presents a retrospective trust region method for unconstrained optimization. The retrospective ratio in this method uses the information at the current iterate and the last iterate point, which can give the more effective estimation of trust region radius. Hence, the number of solving trust region subproblem may be decreased, which improves the effectiveness of the method.

In this paper we present a new algorithm for unconstrained optimization, which is based on the framework of the retrospective trust region method [25] and associated with the technique of the multi-dimensional filter [21,22]. Under reasonable assumptions, we analyze the global convergence of the new method and report the preliminary results of numerical tests. We compare the results with those of the basic trust region algorithm, the filter trust region algorithm and the retrospective trust region algorithm, which shows the effectiveness of the

new algorithm. This paper is organized as follows. The new algorithm is described in Section 2. Basic assumptions and some lemmas are given in Section 3. The analysis of the first order and second order convergence is given in Section 4 and Section 5, respectively. Section 6 reports the numerical results. Finally, we give some final remarks on this approach.

2. Algorithm

In this paper, we define $g(x) = \nabla_x f(x)$. At the current iterate x_k , $f_k = f(x_k)$, $g_k = g(x_k)$, g_{ki} denotes the i th component of the vector g_k . Throughout this paper, $\|\cdot\|$ denotes the Euclidean norm. Now, we review the basic-trust region algorithm as follows.

2.1. Algorithm BTR (Basic Trust Region Algorithm)

Step 0. (Initialization) Given an initial point $x_0 \in R^n$ and an initial trust-region radius $\Delta_0 > 0$, $0 < \eta_1 \leq \eta_2 < 1$ and $0 < \gamma_1 \leq \gamma_2 < 1$. Set $k := 0$.

Step 1. (Model definition) Define a model function m_k in \mathfrak{S}_k , where $\mathfrak{S}_k = \{x \in R^n \mid \|x - x_k\| \leq \Delta_k\}$.

Step 2. (Step calculation) Compute a trial step s_k for solving trust region subproblem (4).

Step 3. (Updating iterate point) Compute $f(x_k + s_k)$ and the ratio

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(x_k) - m_k(x_k + s_k)}.$$

then,

$$x_{k+1} = \begin{cases} x_k + s_k, & \text{if } \rho_k \geq \eta_1, \\ x_k, & \text{if } \rho_k < \eta_1. \end{cases}$$

Step 4. (Updating trust-region radius) Set

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty), & \text{if } \rho_k > \eta_2, \\ [\gamma_2 \Delta_k, \Delta_k), & \text{if } \rho_k \in [\eta_1, \eta_2], \\ [\gamma_1 \Delta_k, \gamma_2 \Delta_k), & \text{if } \rho_k < \eta_1. \end{cases}$$

Set $k := k + 1$, and go to Step 1.

In the algorithm BTR, we do not give a formal stopping criterion. In practice, the stopping criterion can be installed in Step 1, such as

$$\|g_k\| \leq \text{eps} \text{ or } k > k_{\max},$$

where eps denotes the precision, and k_{\max} denotes the maximal number of iterations.

If x^* is a local minimizer of the problem (1), then $g(x^*) = 0$. Motivated by the filter method, we set $\|g(x)\|$ to be the measure of the iterate. Now we intro-

duce some definitions about the multi-dimensional filter.

Definition 2.1 We say that a point x_k dominates another point x_l , if

$$|g_{kj}| \leq |g_{lj}| \text{ for all } j = 1, 2, \dots, n.$$

Definition 2.2 A multi-dimensional filter F is a list of n -tuples of the form (g_{k1}, \dots, g_{kn}) and if $g_k, g_l \in F$, then there exist two indices $j_1, j_2 \in \{1, 2, \dots, n\}$, $j_1 \neq j_2$ such that

$$|g_{lj_1}| < |g_{kj_1}| \text{ and } |g_{kj_2}| < |g_{lj_2}|.$$

Definition 2.3 The iterate point x_k is acceptable for the filter F if and only if for all $g_l \in F$, there exists $j \in \{1, 2, \dots, n\}$ such that

$$|g_{kj}| \leq |g_{lj}| - \gamma_g \|g_l\|, \gamma_g \in (0, 1/\sqrt{n}).$$

Definition 2.4 If the iterate point x_k is acceptable, then it is added to the filter and any iterates in the filter that are dominated by the new iterate are removed, which is called the modification of the filter.

Combining with the filter technique and the retrospective idea, we describe our algorithm as follows.

2.2. Algorithm RFTR (Retrospective Filter Trust-Region Algorithm)

Step 0. (Initialization) An initial point $x_0 \in R^n$ and an initial trust-region radius $\Delta_0 > 0$ are given.

$$\gamma_g \in (0, 1/\sqrt{n}), 0 < \eta_1 < 1, 0 < \tilde{\eta}_1 \leq \tilde{\eta}_2 < 1,$$

$$0 < \gamma_1 \leq \gamma_2 < 1 \leq \gamma_3.$$

Set the initial filter F to be the empty set and set $k := 0$.

Step 1. (Model definition) Define a model function m_k in \mathfrak{S}_k , where $\mathfrak{S}_k = \{x \in R^n \mid \|x - x_k\| \leq \Delta_k\}$.

Step 2. (Updating retrospective trust-region radius) If $k = 0$, then go to Step 3. If $x_k = x_{k-1}$, then choose $\Delta_k \in [\gamma_1 \Delta_{k-1}, \gamma_2 \Delta_{k-1})$. Otherwise, compute the retrospective ratio

$$\tilde{\rho}_k = \frac{f(x_{k-1}) - f(x_k)}{m_k(x_{k-1}) - m_k(x_k)}.$$

Choose

$$\Delta_k \in \begin{cases} [\gamma_1 \Delta_{k-1}, \gamma_2 \Delta_{k-1}), & \text{if } \tilde{\rho}_k < \tilde{\eta}_1, \\ [\gamma_2 \Delta_{k-1}, \Delta_{k-1}), & \text{if } \tilde{\rho}_k \in [\tilde{\eta}_1, \tilde{\eta}_2], \\ [\Delta_{k-1}, \gamma_3 \Delta_{k-1}), & \text{if } \tilde{\rho}_k > \tilde{\eta}_2. \end{cases}$$

Step 3. (Step calculation) Compute a trial step s_k for solving trust region subproblem (4) and $x_k^+ = x_k + s_k$.

Step 4. (Updating iterate point) Compute

$$\rho_k = \frac{f(x_k) - f(x_k^+)}{m_k(x_k) - m_k(x_k^+)}.$$

Case 1: If $\rho_k \geq \eta_1$, then $x_{k+1} = x_k^+$.

Case 2: If $\rho_k < \eta_1$ and x_k^+ is accepted by the filter F , then $x_{k+1} = x_k^+$ and add $g_k^+ = g(x_k^+)$ into the filter F .

Case 3: If $\rho_k < \eta_1$ and x_k^+ is not accepted by the filter F , then $x_{k+1} = x_k$.

Step 5. Set $k := k + 1$, go to Step 1.

Similar to the algorithm BTR, the stopping criterion can be installed in Step 1, such as

$$\|g_k\| \leq eps \text{ or } k > k_{\max},$$

where eps denotes the precision, and k_{\max} denotes the maximal number of iterations. The Hessian matrix in the model function m_k can be obtained by BFGS updating formula or set $\nabla_{xx}m_k(x_k + s) = \nabla_{xx}f(x_k)$ for all s such that $x_k + s \in \mathfrak{S}_k$.

In the algorithm RFTR, the retrospective idea and the filter technique are two important characteristics. The retrospective ratio uses the information at the current iterate and the last iterate to adjust the trust-region radius, which can give the more effective estimation of trust region radius. The filter technique relaxes the condition of accepting a trial step comparing with the usual trust region method, which improves the effectiveness of the algorithm in some sense. From the algorithm RFTR, if the trial point is not accepted (Case 3 in Step 5 occurs), then the algorithm is similar to the basic trust-region algorithm, whose difference is just that we use the retrospective idea in the algorithm RFTR. However, if the trial point is accepted by the algorithm (Case 1 or Case 2 in Step 5 occurs), the retrospective idea and the filter technique all play the roles.

At the iterate x_k , if $x_{k+1} = x_k + s_k \neq x_k$, then the iterate is called the successful iterate and the iteration index x_k is called successful iteration.

3. Basic Assumptions and Lemmas

In this section, we present the global convergence analysis of the algorithm RFTR. We make the following assumptions.

A1 The all iterates x_k remain in a closed and bounded convex set Ω .

A2 $f: R^n \rightarrow R$ is a twice continuously differentiable function.

A3 The model function m_k is first-order coherent with the function f at the iterate x_k , *i.e.*, their values and gradients are equal at x_k for all k ,

$$m_k(x_k) = f(x_k), \nabla_x m_k(x_k) = \nabla_x f(x_k).$$

A4 The Hessian matrix of the model function $\nabla_{xx}m_k$ is uniformly bounded, *i.e.*, there exists a constant $k_{umh} \geq 1$ such that

$$\|\nabla_{xx}m_k(x)\| \leq k_{umh} - 1,$$

holds for all $x \in R^n$ and all k .

Generally speaking, we do not need the global solution of the trust region subproblem. We only expect to decrease the model at least as much as at the Cauchy point. Therefore, we make the following assumption on the solution s_k of the trust region subproblem (4).

A5 There exists a constant k_{mdc} , for all k ,

$$\begin{aligned} & m_k(x_k) - m_k(x_k + s_k) \\ & \geq k_{mdc} \max \left\{ \|g_k\| \min \left\{ \frac{\|g_k\|}{\beta_k}, \Delta_k \right\}, |\tau_k| \min \{ \tau_k^2, \Delta_k^2 \} \right\}, \end{aligned}$$

where

$$\begin{aligned} \beta_k &= 1 + \max_{x \in \mathfrak{S}_k} \|\nabla_{xx}m_k\|, \\ \mathfrak{S}_k &= \{x \in R^n \mid \|x - x_k\| \leq \Delta_k\}, \\ \tau_k &= \min \{0, \lambda_{\min}(\nabla_{xx}m_k(x_k))\}. \end{aligned}$$

By the assumptions A1 and A2, the Hessian matrix $\nabla_{xx}f(x)$ is uniformly bounded on Ω , *i.e.*, there exists a positive constant k_{yfh} such that, for all $x \in \Omega$,

$$\|\nabla_{xx}f(x)\| \leq k_{yfh}.$$

Now we study the global convergence of the algorithm RFTR. First we give a bound on the difference between the objective function and the model function m_k at the iterate x_{k-1} and x_k . The proof of the following result is similar to Theorem 3.1 in [25].

Lemma 3.1 [25] Suppose A1-A4 hold, then exists a positive constant k_{ubh} ,

$$|f(x_k) - m_{k-1}(x_{k-1})| \leq k_{ubh} \Delta_{k-1}^2 \tag{5}$$

and if iteration $k - 1$ is successful, that

$$|f(x_{k-1}) - m_k(x_{k-1})| \leq k_{ubh} \Delta_{k-1}^2 \tag{6}$$

where $k_{ubh} = \max \{k_{yfh}, k_{umh}\}$.

As the retrospective ratio $\tilde{\rho}_k$ uses the reduction in m_k instead of the reduction in m_{k-1} , we need to compare their difference, which is provided by the next Lemma.

Lemma 3.2 [25] Suppose A1-A4 hold, then for every successful iteration $k - 1$,

$$\begin{aligned} & |(m_{k-1}(x_{k-1}) - m_{k-1}(x_k)) \\ & - (m_k(x_{k-1}) - m_k(x_k))| \leq 2k_{ubh} \Delta_{k-1}^2 \end{aligned}$$

We conclude from this result that the denominators in the expression of $\tilde{\rho}_k$ and ρ_k are the same order as the error between the objective function and the model function. Similar to Theorem 6.4.2 in [13], we obtain the next result.

Lemma 3.3 Suppose A1-A5 hold, $g_{k-1} \neq 0$ and Δ_{k-1} satisfies that

$$\Delta_{k-1} \leq \min \left\{ 1 - \eta_1, \frac{1 - \tilde{\eta}_2}{3 - 2\tilde{\eta}_2} \right\} \frac{k_{mdc}}{k_{ubh}} \|g_{k-1}\| \quad (7)$$

Then iteration $k-1$ is successful and

$$\Delta_k \geq \Delta_{k-1}.$$

Proof. It follows from $k_{mdc}, \eta_1 \in (0, 1)$ and the assumptions A3 and A5 that $\beta_{k-1} \leq k_{ubh}$. By (7),

$$\Delta_{k-1} < \frac{\|g_{k-1}\|}{\beta_{k-1}}.$$

By the assumption A5, we have that

$$\begin{aligned} m_{k-1}(x_{k-1}) - m_{k-1}(x_k) &\geq k_{mdc} \|g_{k-1}\| \min \left\{ \frac{\|g_{k-1}\|}{\beta_{k-1}}, \Delta_{k-1} \right\} \\ &\geq k_{mdc} \|g_{k-1}\| \Delta_{k-1} \end{aligned} \quad (8)$$

On the other hand, it follows from (5) and (7) that

$$\begin{aligned} |\rho_{k-1} - 1| &= \left| \frac{f(x_k) - m_{k-1}(x_k)}{m_{k-1}(x_{k-1}) - m_{k-1}(x_k)} \right| \\ &\leq \frac{k_{ubh}}{k_{mdc} \|g_{k-1}\|} \Delta_{k-1} \leq 1 - \eta_1. \end{aligned}$$

Thus, $\rho_{k-1} \geq \eta_1$, *i.e.*, the iteration $k-1$ is successful. Next we proof the second part of the conclusion.

By $k_{mdc}, \tilde{\eta}_2 \in (0, 1)$, we have

$$\frac{1 - \tilde{\eta}_2}{3 - 2\tilde{\eta}_2} < \frac{1}{2} \text{ and } k_{mdc} \frac{1 - \tilde{\eta}_2}{3 - 2\tilde{\eta}_2} < 1 \quad (9)$$

The conditions (7), (9) and the definition of β_{k-1} in the assumption A5 imply that

$$\Delta_{k-1} < \frac{1}{2} \frac{k_{mdc}}{k_{ubh}} \|g_{k-1}\| \text{ and } \Delta_{k-1} < \frac{\|g_{k-1}\|}{\beta_{k-1}}.$$

Combining (8) and Lemma 3.2, we can conclude that

$$\begin{aligned} |m_k(x_{k-1}) - m_k(x_k)| &\geq |m_{k-1}(x_{k-1}) - m_{k-1}(x_k)| - 2k_{ubh} \Delta_{k-1}^2 \\ &\geq k_{mdc} \|g_{k-1}\| \Delta_{k-1} - 2k_{ubh} \Delta_{k-1}^2 \end{aligned} \quad (10)$$

By (6) and (10),

$$|\tilde{\rho}_k - 1| = \left| \frac{f(x_{k-1}) - m_k(x_{k-1})}{m_k(x_{k-1}) - m_k(x_k)} \right| \leq \frac{k_{ubh} \Delta_{k-1}}{k_{mdc} \|g_{k-1}\| - 2k_{ubh} \Delta_{k-1}}.$$

(7) implies that

$$(3 - 2\tilde{\eta}_2) k_{ubh} \Delta_{k-1} \leq (1 - \tilde{\eta}_2) k_{mdc} \|g_{k-1}\|.$$

i.e.,

$$k_{ubh} \Delta_{k-1} \leq (1 - \tilde{\eta}_2) (k_{mdc} \|g_{k-1}\| - 2k_{ubh} \Delta_{k-1}).$$

Therefore, $|\tilde{\rho}_k - 1| \leq (1 - \tilde{\eta}_2)$, *i.e.*, $\tilde{\rho}_k \geq \tilde{\eta}_2$.

As a consequence of this property, we may now prove that the trust region radius cannot become too small as long as a first-order critical point is not approached. The technique of the proof is similar to Theorem 3.4 in [25] and Theorem 6.4.3 in [13].

Lemma 3.4 [13,25] Suppose A1-A5 hold. Suppose, furthermore, that there exists a constant $k_{lbg} > 0$ such that $\|g_k\| \geq k_{lbg}$ for all k . Then there is a constant k_{lbd} such that $\Delta_k \geq k_{lbd}$ for all k .

4. First Order Convergence

Assume that $\{x_k\}$ is an infinite sequence generated by Algorithm RFTR. Under the assumptions (A1)-(A5), we discuss the first order convergence of the sequence $\{x_k\}$. At first, we define the following sets.

The set of successful iteration index

$$S = \{k \mid x_{k+1} = x_k + s_k\}.$$

The set of the iteration index which is added to the filter

$$\begin{aligned} A = \{k \mid g_k^+ \text{ or the iterate } x_k^+ = x_k + s_k \\ \text{is added to the filter } F\}. \end{aligned}$$

The set of the iteration index which satisfies sufficient descent condition

$$D = \{k \mid \rho_k \geq \eta_1\}.$$

$|S|$ denotes the cardinal number of the set S . We now establish the criticality of the limit point of the sequence of the iterates when there are only finitely many successful iterations.

Theorem 4.1 Suppose A1-A5 hold and $|S| < +\infty$, then there exists an index k_0 such that $x_{k_0} \equiv x^*$ and x^* is a first-order critical point.

Proof. Let k_0 be the last successful iteration index, then

$$x_{k_0} = x_{k_0+j} \text{ and } \rho_{k_0+j} < \eta_1, \quad \forall j = 1, 2, \dots$$

From Step 2 of the algorithm RFTR, we have that

$$\Delta_{k_0+j} \leq \gamma_2 \Delta_{k_0+j-1} \leq \dots \leq \gamma_2^j \Delta_{k_0}.$$

Thus,

$$\lim_{j \rightarrow \infty} \Delta_{k_0+j} = 0.$$

It follows from Lemma 3.4 that $\|g_{k_0}\| = 0$.

Next, we consider the case that there are infinitely many successful iteration. From the algorithm RFTR, we know that $A \subseteq S$. Therefore we consider the following two cases.

1) There are infinitely many filter iterations, *i.e.*, $|A| = +\infty$.

2) There are finitely many filter iterations, *i.e.*, $|A| < +\infty$.

First, we have the following result.

Theorem 4.2 Suppose A1-A5 hold and $|A| = |S| = +\infty$, then

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

Proof. Suppose, by contradiction, that the result is not true, then there exists a positive constant k_{lbg} such that

$$\|g_k\| \geq k_{lbg} > 0 \tag{11}$$

holds for all k . Denote the index set $A = \{k_i\}$. It follows from the assumption A1-A5 that $\{\|g_k\|\}$ is bounded. So there exists a subsequence $\{k_{i_l} + 1\}$ of $\{k_i + 1\}$ which satisfies

$$\lim_{l \rightarrow \infty} \|g_{k_{i_l}+1}\| = \|g_\infty\| \geq k_{lbg}.$$

By the definition of k_{i_l} , the iterate point $x_{k_{i_l}+1}$ is accepted by the filter F , and for every $l > 1$ there exists $j_l \in \{1, 2, \dots, n\}$ such that

$$\left|g_{k_{i_l}+1, j_l}\right| - \left|g_{k_{i_{l-1}}+1, j_l}\right| \leq -\gamma_g \|g_{k_{i_{l-1}}+1}\|. \tag{12}$$

Since there is only finite choices of j_l , without loss of generality, we set $j_l = j$. In (12), we follows from $l \rightarrow +\infty$ and (11) that

$$0 \leftarrow \left|g_{k_{i_l}+1, j}\right| - \left|g_{k_{i_{l-1}}+1, j}\right| \leq -\gamma_g k_{lbg} < 0,$$

which is a contradiction. Thus the result is proved.

Now, we give the result when $|A| < +\infty$.

Theorem 4.3 Suppose A1-A5 hold and $|S| = +\infty$, $|A| < +\infty$, then

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

Proof. Suppose, by contradiction, that the result is not true, then there exists a positive constant k_{lbg} such that (11) holds. Since $|A| < +\infty$, by the algorithm RFTR, we have that $\rho_k \geq \eta_1$ holds for all sufficiently large index

$k \in S$. Denote

$$\sigma_k = |\{p, p+1, \dots, k\} \cap S|.$$

It follows from the assumption A5, Lemma 3.4 and $\rho_k \geq \eta_1$ that

$$\begin{aligned} f(x_p) - f(x_{k+1}) &= \sum_{\substack{j=p \\ j \in S}}^k (f(x_j) - f(x_{j+1})) \\ &\geq \sigma_k \eta_1 k_{mde} k_{lbg} \min \left\{ \frac{k_{lbg}}{k_{umh}}, k_{lbd} \right\}. \end{aligned}$$

as long as p, k is sufficiently large. $|S| = +\infty$ and $|A| < +\infty$ imply that σ_k may be large arbitrarily, which contradicts with the fact that $\{f(x_k)\}$ is bounded.

By Theorem 4.1, Theorem 4.2 and Theorem 4.3, there exists at least one limit point of the sequence $\{x_k\}$ of iterates generated by the algorithm RFTR which is a first-order critical point.

5. Second Order Convergence

We now exploit second-order information on the objective function to discuss the second order convergence of the sequence. We therefore introduce the following additional assumptions.

A6 The model is asymptotically second-order coherent with the objective function near first-order critical points, *i.e.*,

$$\lim_{k \rightarrow \infty} \|\nabla_{xx} f(x_k) - \nabla_{xx} m_k(x_k)\| = 0 \text{ where } \lim_{k \rightarrow \infty} \|g_k\| = 0.$$

A7 There exists a constant k_{lch} such that, for all k ,

$$\|\nabla_{xx} m_k(x) - \nabla_{xx} m_k(y)\| \leq k_{lch} \|x - y\|.$$

for all $x, y \in \mathfrak{S}_k$.

Lemma 5.1 Suppose that A1-A7 hold. Suppose also that there exists a sequence $\{k_i\}$ and a constant $k_{mqd} > 0$ such that

$$m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i} + s_{k_i}) \geq k_{mqd} \|s_{k_i}\|^2 > 0 \tag{13}$$

for all i sufficiently large. Finally, suppose that $\lim_{i \rightarrow \infty} \|s_{k_i}\| = 0$, then iteration k_i is successful and

$$\tilde{\rho}_{k_i+1} \geq \tilde{\eta}_2 \text{ and } \Delta_{k_i+1} \geq \Delta_{k_i} \tag{14}$$

for all i sufficiently large.

Proof. We first deduce that every iterations k_i is successful for i sufficiently large. By the mean value theorem and (13), for some ξ_k and ς_k in the segment $[x_{k_i}, x_{k_i+1}]$,

$$\begin{aligned} |\rho_{k_i} - 1| &= \left| \frac{f(x_{k_i+1}) - m_{k_i}(x_{k_i+1})}{m_{k_i}(x_{k_i}) - m_{k_i}(x_{k_i+1})} \right| \\ &\leq \frac{1}{2k_{mqd} \|s_{k_i}\|^2} \left| s_{k_i}^T (\nabla_{xx} f(\xi_{k_i}) - \nabla_{xx} m_{k_i}(\varsigma_{k_i})) s_{k_i} \right| \\ &\leq \frac{1}{2k_{mqd}} \left(\|\nabla_{xx} f(\xi_{k_i}) - \nabla_{xx} f(x_{k_i})\| \right. \\ &\quad \left. + \|\nabla_{xx} f(x_{k_i}) - \nabla_{xx} m_{k_i}(x_{k_i})\| \right. \\ &\quad \left. + \|\nabla_{xx} m_{k_i}(x_{k_i}) - \nabla_{xx} m_{k_i}(\varsigma_{k_i})\| \right), \end{aligned}$$

When i goes to infinity, by our assumption that $\|s_{k_i}\|$ tends to 0, and the bounds

$$\|\xi_{k_i} - x_{k_i}\| \leq \|s_{k_i}\| \text{ and } \|\varsigma_{k_i} - x_{k_i}\| \leq \|s_{k_i}\|.$$

Combining the assumption A2 and A7, the first and third terms of the last right-hand side tend to 0. Meanwhile, the second tends to 0 because of the assumption A6 and Theorem 4.1, 4.2, 4.3. As a consequence, ρ_{k_i} tends to 1. when i goes to infinity, and thus larger than η_1 for i sufficiently large.

The residual proof is similar to Lemma 3.8 in [25].

Theorem 5.2 Suppose that A1-A7 hold and that the complete sequence of iterates $\{x_k\}$ converges to the unique limit point x^* . Then x^* is a second order critical point of (1).

Proof. By Theorem 4.1, 4.2, 4.3, $g(x^*) = 0$. We suppose, by contradiction, that $\tau_* = \lambda_{\min}(\nabla_{xx} f(x^*)) < 0$, by the assumption A6, there exists k_0 such that, for all $k \geq k_0, \tau_k = \lambda_{\min}(\nabla_{xx} m_k(x_k)) < \frac{1}{2} \tau_*$. It follows from the assumption A5 that

$$m_k(x_k) - m_k(x_k + s_k) \geq \frac{1}{2} k_{mdc} |\tau_*| \min \left\{ \frac{1}{4} |\tau_*^2|, \Delta_k^2 \right\}$$

hold for all $k \geq k_0$. Meanwhile, by the assumption we know that $s_k = x_{k+1} - x_k$ tends to 0. Thus, there exists $k_1 \geq k_0$ such that, for all $k \geq k_1, \|s_k\| \leq \min \{1/2|\tau_*|, \Delta_k\}$. By Lemma 5.1, there exists $k_2 \geq k_1$ such that, for all $k \geq k_2$, we deduce that $\rho_k \geq \eta_1, \tilde{\rho}_{k+1} \geq \tilde{\eta}_2$ and $\Delta_{k+1} \geq \Delta_k$. Thus,

$$\begin{aligned} f(x_k) - f(x_{k+1}) &\geq \eta_1 (m_k(x_k) - m_k(x_k + s_k)) \\ &\geq \frac{1}{2} \eta_1 k_{mdc} |\tau_*| \min \left\{ \frac{1}{4} |\tau_*^2|, \Delta_k^2 \right\}, \end{aligned}$$

which follows from $\lim_{k \rightarrow \infty} x_k = x^*$ that $\lim_{k \rightarrow \infty} \Delta_k = 0$. This contradicts with $\Delta_{k+1} \geq \Delta_k$ for all $k \geq k_2$. Thus, $\tau_* \geq 0$ and therefore x^* is a second-order critical point of (1).

6. Numerical Experiments

In this section, a preliminary numerical test of the algorithm BTR, the algorithm FTR [22], the algorithm RTR [25] and the algorithm RFTR are given. The Matlab codes (Version 7.4.0.287 R2007a) were written corresponding to those algorithms. For the numerical tests, we use the following trust-region radius updated form which is proposed in Conn *et al.* [13].

$$\Delta_{k+1} = \begin{cases} \gamma_1 \|s_k\|, & \text{if } \tilde{\rho}_{k+1} < \tilde{\eta}_1, \\ \Delta_k, & \text{if } \tilde{\rho}_{k+1} \in [\tilde{\eta}_1, \tilde{\eta}_2], \\ \max(\gamma_3 \|s_k\|, \Delta_k), & \text{if } \tilde{\rho}_{k+1} > \tilde{\eta}_2. \end{cases}$$

and the following parameter settings [21,28].

$$\Delta \gamma_1 = 0.25, \gamma_3 = 3.5, \tilde{\eta}_1 = \eta_1 = 0.0001, \tilde{\eta}_2 = \eta_2 = 0.99,$$

$$\Delta_0 = 1, \gamma_g = \min \left\{ 0.001, 1 / (2\sqrt{n}) \right\}.$$

The Hessian matrix of the model function is $\nabla_{xx} m_k(x) = \nabla_{xx} f(x_k)$. The termination criterion is as following,

$$\left\{ \|g_k\| \leq 10^{-6} \sqrt{n} \text{ or } k > k_{\max} \right\}, k_{\max} = 1000,$$

where k_{\max} denotes the maximal iteration number.

We choose 24 test problems from [29], where ‘‘S201’’ means problem 201 in Schittkowski (1987) collection [29], 12 test problems from CUTE [25,30] and the famous Extended Rosenbrock test problem. In the following tables, ‘‘n’’ means the test problem’s dimension, ‘‘nBTR, nFTR, nRTR, nRFTR’’ mean the number of iterations of the algorithm BTR, the algorithm FTR, the algorithm RTR and the algorithm RFTR, respectively. ‘‘ng1, ng2, ng3, ng4’’ mean the number of gradient evaluations of the algorithm BTR, the algorithm FTR, the algorithm RTR and the algorithm RFTR, respectively.

‘‘r’’ means the rank of the number of iterations of the algorithm RFTR among the four algorithms, whose values is in $\{1, 2, 3, 4\}$, where ‘‘1’’ means that the number of iterations of the algorithm RFTR is the smallest among the four algorithms, so the algorithm RFTR is the best one among the four algorithms. ‘‘4’’ means that the number of iterations of the algorithm RFTR is the largest among the four algorithms, so the algorithm RFTR is the worst one among the four algorithms. ‘‘F’’ means that the algorithm does not stop when the maximal iteration number is achieved.

In **Table 1**, there are 20 test problems whose iteration number is the smallest, 2 test problems whose rank is second, 2 test problems whose iteration number is the largest among 24 test problems. The numerical results

Table 1. Reports the numerical results on 24 test problems from [29].

Problem	n	nBTR	nFTR	nRTR	nRFTR	ng1	ng2	ng3	ng4	r
S201	2	34	34	34	34	35	35	35	35	1
S202	2	6	6	6	6	7	7	7	7	1
S203	2	6	6	6	6	7	7	7	7	1
S205	2	8	8	8	8	9	9	9	9	1
S206	2	4	4	4	4	5	5	5	5	1
S207	2	9	7	11	7	8	9	10	9	1
S208	2	39	17	27	17	27	23	26	23	1
S209	2	108	45	99	53	86	55	93	61	2
S210	2	424	169	460	176	365	195	450	185	2
S211	2	37	13	37	12	29	17	32	16	1
S212	2	13	14	12	14	11	16	11	16	4
S213	2	27	27	27	27	28	28	28	28	1
S240	3	5	5	5	5	6	6	6	6	1
S241	3	13	13	13	13	14	14	14	14	1
S246	3	10	7	10	7	10	9	10	9	1
S256	4	15	15	15	15	16	16	16	16	1
S260	4	69	33	53	33	47	37	48	37	1
S261	4	13	15	13	15	13	17	13	17	4
S271	6	2	2	2	2	3	3	3	3	1
S273	6	10	10	10	10	11	11	11	11	1
S274	2	2	2	2	2	3	3	3	3	1
S275	4	2	2	2	2	3	3	3	3	1
S276	6	2	2	2	2	3	3	3	3	1
S308	2	11	9	11	9	10	11	10	11	1

Table 2. Reports the numerical results on the famous Extended Rosenbrock test problem.

n	nBTR	nFTR	nRTR	nRFTR	ng1	ng2	ng3	ng4	r
2	39	17	27	17	27	23	26	23	1
10	36	22	31	22	28	29	29	29	1
20	67	36	68	36	49	43	67	43	1
30	83	41	60	42	62	56	58	55	2
40	111	52	100	52	81	70	98	70	1
50	150	69	130	69	405	93	127	91	1
60	170	82	147	82	122	109	144	107	1
70	199	102	162	101	141	140	159	135	1
80	218	121	171	120	157	155	170	147	1
90	248	128	209	121	177	171	207	150	1
100	278	141	220	141	179	182	218	180	1
150	397	207	296	213	284	276	293	274	2
200	532	302	427	283	378	391	423	361	1
250	647	374	507	373	464	478	504	474	1
300	769	419	722	419	551	542	719	537	1
350	900	507	F	501	644	653	996	627	1
400	F	585	F	500	715	748	997	536	1
450	F	640	878	630	712	834	873	798	1
500	F	719	F	710	712	938	994	900	1

Table 3. Reports the numerical results on 12 test problems from CUTE [25,30].

Problem	n	nBTR	nFTR	nRTR	nRFTR	ng1	ng2	ng3	ng4	r
ARHEAD	100	5	5	5	5	6	6	6	6	1
ROSNSCHN	50	66	39	100	39	51	46	98	46	1
COSINE	100	F	F	34	36	986	1008	32	40	2
DQDR TIC	100	4	4	4	4	5	5	5	5	1
ERRINROS	50	52	67	48	27	39	78	44	33	1
FLERCHCR	100	29	29	29	29	30	30	30	30	1
LIARWHD	300	13	13	13	13	14	14	14	14	1
LOGHAIRY	2	61	64	55	31	51	66	50	33	1
NONDIA	100	24	24	24	24	25	25	25	25	1
PFIT4LS	3	271	291	309	233	217	323	300	245	1
POWELLS	4	15	15	15	15	16	16	16	16	1
WOOD	4	69	33	53	33	47	37	48	37	1

show that the number for the algorithm RFTR to solve trust region subproblem is the smallest in total.

In **Table 2**, There are only 2 cases whose rank is second, the others all are the best. Moreover, the algorithm RFTR is more and more effective as the increase of the problem's dimension.

In **Table 3**, There are only 1 case whose rank is second, the others all are the best. Moreover, The retrospective idea takes effects on the Problems COSINE, ERRINROS, LOGHAIRY clearly.

7. Conclusions and Perspectives

Trust region method is very reliable and robust and has very strong convergence properties. It is a class of very effective algorithms for solving unconstrained optimization now. The basic trust region algorithm is the monotone descent algorithm, *i.e.*, the value of the object function in the iterate sequence $\{x_k\}$ strictly decreases monotonically. If the iterates follow the bottom of curved narrow valleys, then the monotone descent algorithm converges very slowly. The idea of non-monotone method [23,24] abandons the restriction of the descent property of the value of the object function, which allows the sequence of iterates to follow the bottom of curved narrow valleys much more loosely, which hopefully results in longer and more efficient steps.

Trust region method combines with the filter technique, which, in some sense, relaxes the monotonicity condition which accepts the trial step. The filter technique improves the numerical effect for some problems.

The new algorithm RFTR presented in this paper combines with the filter technique and the retrospective idea, which the number of the algorithm RFTR to solve trust

region subproblem is decreased in total. On the other hand, our algorithm also looks like a self-adaptive method based on the trust-region framework. Meanwhile, our algorithm is not like the other algorithms about self-adaptive method [26,27] which need to compute the gradient value and function value at the auxiliary point, but may measure the acceptance of the previous iterate and the current iterate for the new and old model function, respectively, which keep the robustness property of the trust-region method.

8. References

- [1] M. J. D. Powell, "A New Algorithm for Unconstrained Optimization," In: J. B. Rosen, O. L. Mangasarian and K. Ritter, Eds., *Nonlinear Programming*, Academic Press, New York, 1970.
- [2] K. Levenberg, "A Method for The Solution of Certain Nonlinear Problems in Least Squares," *The Quarterly of Applied Mathematics*, Vol. 2, No. 2, 1944, pp. 164-168.
- [3] D. W. Marquardt, "An Algorithm for Least Squares Estimation of Nonlinear Inequalities," *SIAM Journal on Applied Mathematics*, Vol. 11, No. 2, 1963, pp. 431-441.
- [4] M. J. D. Powell, "Convergence Properties of a Class of Minimization Algorithms," In: O. L. Mangasarian, R. R. Meyer and S. M. Robinson, Eds., *Nonlinear Programming*, Academic Press, New York, 1975, pp. 1-27.
- [5] M. J. D. Powell, "On the Global Convergence of Trust-Region Algorithms for Unconstrained Optimization," *Mathematical Programming*, Vol. 29, No. 3, 1984, pp. 297-303.
- [6] G. A. Schultz, R. B. Schnabei and R. H. Byrd, "A Family of Trust-Region-Based Algorithms for Unconstrained Minimization with Strong Global Convergence," *SIAM Journal on Numerical Analysis*, Vol. 22, No. 1, 1985, pp. 47-67.

- [7] D. C. Sorensen, "Newton's Method with a Model Trust Region Modifications," *SIAM Journal on Numerical Analysis*, Vol. 19, No. 2, 1982, pp. 409-426.
- [8] J. J. Moré, "Recent Developments in Algorithms and Software for Trust Region Methods," In A. R. Bachem, M. Grotshel and B. Korte, Eds., *Mathematical Programming: The State of the Art*, Springer-Verlag, Berlin, 1983, pp. 258-287.
- [9] Y. X. Yuan, "On the Convergence of Trust Region Algorithm," *Mathematica Numerica Sinica*, Vol. 16, No. 3, 1996, pp. 333-346.
- [10] M. Lalee, J. Nocedal and T. Plantenga, "On the Implementation of an Algorithm for Large-Scale Equality Constrained Optimization," *SIAM Journal on Optimization*, Vol. 8, No. 3, 1998, pp. 682-706.
- [11] A. Friedlander, J. M. Martinez and S. A. Santos, "A New Trust Region Algorithm for Bound Constrained Minimization," *Applied Mathematics and Optimization*, Vol. 30, No. 3, 1994, pp. 235-266.
- [12] A. R. Conn, N. I. M. Gould and P. L. Toint, "Convergence Properties of Minimization Algorithms for Convex Constraints Using a Structured Trust Region," *SIAM Journal on Optimization*, Vol. 6, No. 4, 1996, pp. 1059-1086.
- [13] A. R. Conn, N. I. M. Gould and P. L. Toint, "Trust Region Methods," MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2000.
- [14] R. Fletcher and S. Leyffer, "Nonlinear Programming without a Penalty Function," *Mathematical Programming*, Vol. 91, No. 2, 2002, pp. 239-269.
- [15] R. Fletcher, N. I. M. Gould, S. Leyffer, P. L. Toint and A. Wächter, "Global Convergence of a Trust-Region SQP-Filter Algorithm for General Nonlinear Programming," *SIAM Journal on Optimization*, Vol. 13, No. 3, 2002, pp. 635-659.
- [16] R. Fletcher, S. Leyffer and P. L. Toint, "On the Global Convergence of a Filter-SQP Algorithm," *SIAM Journal on Optimization*, Vol. 13, No. 1, 2002, pp. 44-59.
- [17] M. Ulbrich, S. Ulbrich and L. N. Vicente, "A Globally Convergent Primal Dual Interior-Point Filter Method for Nonconvex Nonlinear Programming," *Mathematical Programming*, Vol. 100, No. 2, 2003, pp. 379-410.
- [18] A. Wächter and L. T. Biegler, "Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence," *SIAM Journal on Optimization*, Vol. 16, No. 1, 2005, pp. 1-31.
- [19] A. Wächter and L. T. Biegler, "Line Search Filter Methods for Nonlinear Programming: Local Convergence," *SIAM Journal on Optimization*, Vol. 16, No. 1, 2005, pp. 32-48.
- [20] R. Fletcher, S. Leyffer and P. L. Toint, "A Brief History of Filter Methods," *SIAG/OPT Views and News*, Vol. 18, No. 1, 2006, pp. 2-12.
- [21] N. I. M. Gould, C. Sainvitu and P. L. Toint, "A Filter-Trust-Region Method for Unconstrained Optimization," *SIAM Journal on Optimization*, Vol. 16, No. 2, 2005, pp. 341-357.
- [22] W. H. Miao and W. Y. Sun, "A Filter Trust-Region Method for Unconstrained Optimization," *Numerical Mathematics - A Journal of Chinese Universities. Gaodeng Xuexiao Jisuan Shuxue Xuebao*, Vol. 29, No. 1, 2007, pp. 88-96.
- [23] L. Grippo, F. Lampariello and S. Lucidi, "A Non-monotone Line Search Technique for Newton's Methods," *SIAM Journal on Numerical Analysis*, Vol. 23, No. 4, 1986, pp. 707-716.
- [24] P. L. Toint, "Non-Monotone Trust-Region Algorithms for Nonlinear Optimization Subject to Convex Constraints," *Mathematical Programming*, Vol. 77, No. 1, 1997, pp. 69-94.
- [25] F. Bastin, V. Malmedy, M. Mouffe, P. L. Toint and D. Tomanos, "A Retrospective Trust-Region Method for Unconstrained Optimization," *Mathematical Programming*, Vol. 123, No. 2, 2010, pp. 395-418.
- [26] A. Sartenaer, "Automatic Determination of an Initial Trust Region in Nonlinear Programming," *SIAM Journal on Scientific Computing*, Vol. 18, No. 6, 1997, pp. 1788-1803.
- [27] X. S. Zhang, Z. W. Chen and J. L. Zhang, "A Self-Adaptive Trust Region Method Unconstrained Optimization," *Operations Research Transactions*, Vol. 5, No. 1, 2001, pp. 53-62.
- [28] N. I. M. Gould, D. Orban, A. Sartenaer and P. L. Toint, "Sensitivity of the Trust-Region Algorithms to Their Parameters," *4OR: A Quarterly Journal of Operations Research*, Vol. 3, No. 3, 2005, pp. 227-241.
- [29] K. Schittkowski, "More Test Examples for Nonlinear Programming Codes," *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin, Heidelberg, Vol. 282, 1987.
- [30] H. Y. Benson, "Nonlinear Optimization Models by AMPL: Cute Set." <http://www.princeton.edu/~rvdb/ampl/nlmodels>