An Adjustable Algorithm Based on Non-Monotone Strategy for Optimization Problems

Shaojing Lian, Qiang Hua

Key Lab. of Machine Learning and Computational Intelligence, College of Mathematics and Information Science, Hebei University, Baoding, China

*Corresponding Author:
Shaojing Lian
Email: 48458272@qq.com

Abstract: This paper devotes to incorporating a non-monotone strategy with an adjusted trust region radius to propose a more efficient trust region approach for unconstrained optimization. The primary objective of the paper is to introduce a more relaxed trust region approach based on a novel extension in trust region ratio and radius. The global convergence is proved under some reasonable conditions.

Keywords: unconstrained optimization; trust region; non-monotone strategy; adjustable radius; global convergence.

INTRODUCTION

In this paper, we consider the following unconstrained optimization problem

$$\min f(x), \quad x \in \mathbb{R}^n$$

(1)

where $f(x) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a real-valued twice continuously differentiable function. Two important methods have been developed for solving this problem, namely, line search and trust region methods [1, 2]. Trust region methods are a prominent class of methods for unconstrained optimization problems defining an area around the current step $x_k$ in which the quadratic model has a good agreement with the objective function. In these methods, in each iterate, a trial step $d_k$ is obtained by solving the following quadratic sub-problem:

$$\begin{align*}
\min & \quad q_k(d) = f_k + g_k^T d + \frac{1}{2} d^T B_k d \\
\text{s.t.} & \quad \|d\| \leq \Delta_k
\end{align*}$$

(2)

where $f_k = f(x_k)$, $g_k = \nabla f(x_k)$, and $B_k \in \mathbb{R}^{n \times n}$ is a symmetric matrix which is the Hessian matrix or its approximation of $f(x)$ at the current point $x_k$. $\Delta_k$ is called the trust radius and $\|\|\|$ refers to the 2-norm. A crucial point in trust region methods is a strategy of choosing the trust region radius $\Delta_k$, at every iterate. In the standard trust region method, based on agreement between the model and the objective function, the radius of trust region is updated by the following ratio

$$r_k^2 = \frac{f(x_k) - f(x_k + d_k)}{q_k(0) - q_k(d_k)}$$

(3)

The numerator and the denominator of (3) have been called the actual reduction and the predicted reduction, respectively. It can be concluded that there will be a good agreement between the model and the objective function over current region of trust whenever $r_k^2$ be close to 1. In this case, it is safe to increase the trust region radius in the next iterate. Otherwise, the trust region radius must be shrunk.

It is well known that the standard trust region method is very sensitive on the initial trust region radius [3-5]. In other word, we know that the standard trust region radius $\Delta_k$ is independent from $g_k$ and $B_k$, so we do not know the radius $\Delta_k$ is suitable to the whole of implementation. This situation possibly increases the number of solving sub-problems in the inner steps of the method and so decreases the efficiency of the method. It is obvious that if we decrease the number of ineffective iterates, we can decline the number of solving sub-problems in each step. In [4], Sartenaer
proposed an approach to determine the initial radius monitoring agreement between the model and the objective function along the steepest descent direction computed at the starting point. The first adjustable strategy to determine the trust region radius, for decreasing the number of solving sub-problems, was proposed by Zhang et al. in [6].

This strategy used the information of gradient and Hessian in current iterate to construct the trust region radius without requiring any initial trust region radius. Inspired by Zhang’s strategy, Shi and Guo in [5] proposed a automatically adjustable radius for trust region methods. They proved that the new method preserves the global, the super-linear and the quadratic convergence properties of the standard method. The numerical experiments have been showed that this method is more efficient than Zhang’s method and standard trust region method. We describe these trust region radii in the next section.

On the other hand, Grippo et al. in [7, 8] provided a non-monotone strategy to line search methods for unconstrained optimization problems. In their non-monotone line search, step-length \( \alpha_k \) is accepted if it satisfies the following Armijo-type condition

\[
 f(x_k + \alpha_k d_k) \leq f(x_{k-j}) + \phi \alpha_k \nabla f(x_k)^T d_k 
\]

where \( \phi \in (0,1) \), \( m_0 = 0 \) and \( 0 \leq m_k \leq \min\{m_{k-1} + 1, M\} \) with an integer constant \( M > 0 \). Theoretical analysis and numerical experiments have been indicated the efficiency of this strategy to improve both the possibility of finding the global optimum and the rate of convergence of algorithm [8]. Motivated by these outstanding results, many researchers have interested to work on combination the non-monotone strategy with the trust region methods [9-12]. Gu and Mo in [13] proposed a new non-monotone strategy. The method substitutes the ratio (3) changed as

\[
\left( \frac{D_k - f(x_k + d_k)}{m_k (0) - m_k (d_k)} \right) 
\]

In their proposal, the ratio (3) changed as

\[
\left( \frac{D_k - f(x_k + d_k)}{m_k (0) - m_k (d_k)} \right) 
\]

The investigation have been proved that the combination of the non-monotone strategy with trust region made a new method which has been inherited the strong theoretical properties of trust region as well as the computational robustness of the non-monotone strategy.

The rest of this paper organized as follows. In Section 2, we describe a novel hybrid of non-monotone trust region methods with an adjustable radius. Some properties and the global convergence of the new method are investigated in Section 3. Finally, some concluding remarks are delivered in Section 4.

NOVEL TRUST REGION ALGORITHM

In this section, we describe the trust region radius of Shi and Guo [5]. Then we introduce a new non-monotone trust region algorithm with automatically adjustable radius based on the adjustable radius of Shi and Guo and the idea of the non-monotone strategy of Gu and Mo. We also establish some properties of the new algorithm.

In 2008, Shi and Guo in [5] proposed a variant adjustable radius for trust region method. They selected parameters \( \mu, \rho \in (0,1) \), \( \tau \in (0,1) \) and \( q_k \) to be satisfy in

\[
 - \frac{g_k^T q_k}{\|g_k\|q_k} \geq \tau 
\]

The method provides a new trust region radius by

\[
\Delta_k = \sigma_k \|q_k\| 
\]

where \( \sigma_k = \rho^p s_k \), and \( p_k \) is the smallest positive integer number \( p \) such that \( r_k \geq \mu \). They also determine the term \( s_k \) by

Available Online:  http://scholarsbulletin.com/
\[ S_k = - \frac{g_k^T q_k}{q_k^T \hat{B}_k q_k} \]  \hspace{1cm} (9)

in which \( \hat{B}_k \) is generated by the procedure: \( \hat{B}_k = B_k + iI \), where \( i \) is the smallest nonnegative integer such that the condition \( q_k^T \hat{B}_k q_k > 0 \) holds. It is obvious that if the matrix \( B_k \) be a positive definite matrix, then there is no need to substitute \( \hat{B}_k \) by \( B_k \). To avoid this substitution, we take advantage of a positive definite version of the BFGS quasi-Newton formula as follows

\[ B_{k+1} = B_k + \frac{s_k y_k}{s_k^T y_k} - \frac{s_k s_k^T y_k y_k^T s_k}{s_k^T y_k} \]  \hspace{1cm} (10)

where \( s_k = x_{k+1} - x_k \) and \( y_k = g_{k+1} - g_k \). In order to preserve positive definite property, we do not update \( B_k \) whenever the curvature condition, i.e. \( s_k^T y_k > 0 \), does not hold, i.e. \( \hat{B}_k = B_k \). We now can outline our non-monotone trust region algorithm with adjustable radius as follows:

**Algorithm 1: non-monotone adjustable trust region algorithm**

Input: An initial point \( x_0 \in \mathbb{R}^n \), a symmetric positive definite matrix \( B_0 \in \mathbb{R}^{n \times n} \), \( k_{max} \in \mathbb{N} \), \( \rho, \mu \in (0,1) \), \( \eta_0 \in [\eta_{max}, \eta_{min}] \) and \( \varepsilon > 0 \).

Begin

1. \( \Delta_0 = \|g_0\|, \quad R_0 = f_0, \quad p = 0, \quad r_0 = 0, \quad k = 0. \)

2. While \( (\|g_k\| \geq \varepsilon \) and \( k \leq k_{max}) \) \{Start of outer loop\}

3. While \( (r_k < \mu) \) \{Start of inner loop\}

4. Specify the trial point \( d_k \) by solving the sub-problem (2).

5. Determine the trust-region ratio \( r_k \) using (6).

6. If \( r_k > \mu \), set \( p = p + 1 \) and update the trust region radius \( \Delta_k \) with (8).

    Else \( x_{k+1} = x_k + d_k \).

7. Break;

8. End if

9. End while \{End of inner loop\}

10. \( p = 0; \) Determine \( \Delta_k \) using (8); update \( B_{k+1} \) if \( s_k^T y_k > 0 \); generate \( \eta_{k+1} \) by an adaptive formula; calculate \( D_{k+1} \) by (5); \( k = k + 1 \)

11. End while \{End of outer loop\}

End

**CONVERGENCE ANALYSIS**

In this section, we prove the some properties of the new algorithm that are prominent to prove its convergence analysis. Throughout the paper, we consider the following assumptions in order to analyze the convergence of the new algorithm: (H1) the objective function \( f(x) \) has a lower bound on \( \mathbb{R}^n \) and \( g(x) = \nabla f(x) \) is uniformly continuous on open convex set \( \Omega \) that contains the level set \( L(x_0) = \{ x \in \mathbb{R}^n \mid f(x) \leq f(x_0) \} \). (H2) \( B_k \) is uniformly bounded, i.e., there exists a constant \( M > 0 \) such that \( \|B_k\| \leq M \), for all \( k \).

**Remark 1:** To establish strong theoretical results, it is supposed that the model \( m_k(d) \) decreases at least as much as a fraction of that obtained in Cauchy point, i.e. there exists \( 0 < \beta < 1 \) such that, for all \( k \),

\[ m_k(0) - m_k(d_k) \geq \beta \|g_k\| \min \{\Delta_k, \frac{1}{\|g_k\|}\} \]  \hspace{1cm} (11)
Remark 2: If \( f(x) \) is a twice continuously differentiable function and the level set \( L(x_0) \) is bounded, then (H1) implies that \( \| \nabla^2 f(x) \| \) is uniformly continuous and bounded on the open bounded convex set \( \Omega \) that contains \( L(x_0) \).

Hence, there exists a constant \( M_1 \) such that \( \| \nabla f(x) \| \leq M_1 \) and by using mean value theorem we have
\[
\| g(x) - g(y) \| \leq M_1 \| x - y \| \quad \forall x, y \in \Omega
\]

Lemma 1: Suppose that the sequence \( \{ x_k \} \) be generated by algorithm 1. Then, for all \( k \in N \), we have
\[
m_k(0) - m_k(d_k) \geq m_k(0) - m_k(\sigma_k d_k) \geq -\frac{1}{2} \sigma_k g_k^T q_k
\]
where \( d_k \) is the optimal solution of the sub-problem (2) with respect to \( \sigma_k \).

Proof. A proof of this lemma can be observed in [5].

Lemma 2: Suppose that the sequence \( \{ x_k \} \) be generated by Algorithm 1, then we have
\[
\| m_k(d_k) - f(x_k + d_k) \| \leq O(\| d_k \|^2)
\]
Proof. See Coon, Gould and Toint [14].

Lemma 3: Suppose that the sequence \( \{ x_k \} \) be generated by Algorithm 1, then we have
\[
f^*_{k+1} \leq D^*_{k+1} \leq D_k
\]
Proof. Let iterate \( k \) be a successive iterate. From \( r_k \geq \mu \) and (11), we have
\[
D_k - f_{k+1} \geq \mu \text{Pr}ed_k \geq \mu \beta \| g_k \| \min \left\{ \Delta_k, \frac{\| \theta \|}{\| k \|} \right\} \geq 0
\]
\[
D_k - f_{k+1} = \eta_{k+1}(D_k - f_{k+1}) \geq \mu \eta_{k+1} \| g_k \| \min \left\{ \Delta_k, \frac{\| \theta \|}{\| k \|} \right\} \geq 0
\]
From inequalities (13) and (14), we have
\[
D_{k+1} = \eta_{k+1}D_k + (1 - \eta_{k+1})f_{k+1} \leq \eta_{k+1}D_k + (1 - \eta_{k+1})D_{k+1}
\]
Thus, \( \eta_{k+1}(D_k - D_{k+1}) \geq 0 \).

Hence, (12) holds and the proof is completed.

Lemma 4: Suppose that (H1) and (H2) hold and the sequence \( \{ x_k \} \) be generated by Algorithm 1. Then inner loop is well-defined.

Proof. We prove \( r_k \geq \mu \), for sufficiently large \( p_k \). Let \( d_k^p \) is a solution of the sub-problem (2) corresponding to \( p \)-th inner loop execution in \( x_k \). From Lemma 3.1, it obtains that
\[
m_k(0) - m_k(d_k) \geq -\frac{1}{2} \sigma_k g_k^T q_k
\]
This fact together with Lemma 3.2 imply
\[
\frac{f(x_k) - f(x_k + d_k^p)}{m_k(0) - m_k(d_k^p)} - 1 = \frac{f(x_k) - f(x_k + d_k^p) - (m_k(0) - m_k(d_k^p))}{m_k(0) - m_k(d_k^p)} \leq O(\| d_k^p \|^2) - \frac{1}{2} \sigma_k g_k^T q_k
\]
where the last inequality is obtained using (2) and (8). If \( f \to \infty \), then \( \sigma_k(p) \to 0 \) and using (8), right hand side of the preceding inequality tends to zero. Thus, using (12), we obtain
\[
\frac{C_k - f(x_k + d_k^p)}{m_k(0) - m_k(d_k^p)} \geq \frac{f(x_k) - f(x_k + d_k^p)}{m_k(0) - m_k(d_k^p)} \geq \mu
\]
Therefore, for sufficiently large \( p_k \), we get \( r_k \geq \mu \). This straightforwardly implies that inner loop of the algorithm is well-defined.
Lemma 5: Suppose that (H1) and (H2) hold and the sequence \( \{x_k\} \) be generated by Algorithm 1. Then we have
\[
\lim_{k \to \infty} D_k = \lim_{k \to \infty} f(x_k)
\] (16)

Proof. Due to the definition of \( D_k \), we have
\[
D_{k+1} = \eta_{k+1}D_k + (1-\eta_{k+1})f_{k+1}
\]
By (12), we obtain
\[
f_{k+1} - D_{k+1} = \eta_{k+1}(f_{k+1} - D_k) \leq \eta_{k+1}(D_{k+1} - D_k)
\] (17)
From lemma 3.3, we know that \( \{D_k\} \) is convergent. We notice that \( \eta_{\min} \in [0,1) \), \( \eta_{\max} \in [\eta_{\min},1) \), \( \eta_{k+1} \in [\eta_{\max},\eta_{\min}] \). And as \( k \to \infty \), we have
\[
\lim_{k \to \infty} f_{k+1} - D_{k+1} \leq \lim_{k \to \infty} \eta_{k+1}(D_{k+1} - D_k) = 0
\] (18)
Therefore, the lemma is true.

Theorem 6: Suppose that (H1) and (H2) hold, then Algorithm 1 either stops at a stationary point of (1) or generates an infinite sequence \( \{x_k\} \) such that
\[
\lim_{k \to \infty} -\frac{\varepsilon^T q_k}{\|q_k\|} = 0
\] (19)

Proof. If Algorithm 1 does not stop at a stationary point, we prove that (19) holds. Suppose that Algorithm 1 generates the sequence \( \{x_k\} \) and
\[
\lim_{k \to \infty} -\frac{\varepsilon^T q_k}{\|q_k\|} \neq 0
\]
Thus, there exists \( \epsilon_0 \) and an infinite subset \( K \subseteq \{0,1,2,\cdots\} \) such that
\[
-\frac{\varepsilon^T q_k}{\|q_k\|} \geq \epsilon_0 \quad \forall k \in K
\] (20)
From (H2), we know that there exists a constant \( M > 0 \) such that \( \|B_k\| \leq M \), for all \( k \in N \). Hence we have that
\[
q_k^TB_kq_k \leq M \|q_k\|^2 \quad \forall k \in N
\] (21)
By defining \( K_1 = \{k \in K | \sigma_k = s_k \} \) and \( K_2 = \{k \in K | \sigma_k < s_k \} \), it is obvious that \( K = K_1 \cup K_2 \) is an infinite subset of \( \{0,1,2,\cdots\} \). We now prove that neither \( K_1 \) nor \( K_2 \) can be an infinite set contradicting with (20).

First, we assume that \( K_1 \) is an infinite subset of \( K \). Lemma 3.1 and (21) lead us to
\[
D_k - f(x_k + d_k) \geq \mu (m_k(0) - m_k(d_k))
\]
\[
\geq -\frac{1}{2} \mu \sigma_k s_k^T q_k \geq -\frac{1}{2} \mu \sigma_k \|s_k\| q_k
\]
\[
= \frac{1}{2} \mu \frac{(s_k^T q_k)^2}{\|s_k\| q_k} \geq \frac{\mu}{2M} \frac{(s_k^T q_k)^2}{\|q_k\|} \geq \frac{\mu}{2M} \epsilon_0^2
\]
The previous inequality together with Lemma 3.5, as \( k \to \infty \), suggest
\[
\frac{\mu}{2M} \epsilon_0^2 \leq 0
\]
This is a contradiction. Thus \( K_1 \) can not be an infinite subset of \( K \).

Now, we let that \( K_2 \) be an infinite subset of \( K \). From Lemma 3.1, we get
\[
D_k - f(x_k + d_k) \geq \mu (m_k(0) - m_k(d_k))
\]
\[
\geq -\frac{1}{2} \mu \sigma_k \|q_k\| \|s_k^T q_k\| \geq \frac{1}{2} \mu \sigma_k \|q_k\| \epsilon_0
\]
This inequality along with Lemma 3.5, as \( k \to \infty \), imply that
\[
\lim_{k \to \infty} \Delta_k = \lim_{k \to \infty} \sigma_k \|q_k\| = 0, \quad k \in K_2
\]  
(22)

Now, suppose that \( \tilde{d}_k \) is an optimal solution of the following sub-problem
\[
\min g_k^T d + \frac{1}{2} d^T B_k d \quad \text{s.t.} \quad \|d\| \leq \bar{\Delta}_k, \quad \bar{\Delta}_k = \frac{\Delta}{\rho}
\]

From the definition of \( \Delta_k \), it is clear that
\[
\frac{D_k - f(x_k + \tilde{d}_k)}{m_k(0) - m_k(d_k)} < \mu, \quad \|\tilde{d}_k\| \leq \bar{\Delta}_k, \quad \forall k \in K_2
\]
(23)

On the other hand, (22) suggests that
\[
\lim_{k \to \infty} \|\tilde{d}_k\| = 0, \quad k \in K_2
\]
(24)

Using Lemma 3.1, (20) and (24), for \( k \in K_2 \), we can write
\[
\frac{f(x_k) - f(x_k + \tilde{d}_k)}{m_k(0) - m_k(d_k)} - 1 \leq \frac{f(x_k) - f(x_k + \tilde{d}_k) - (m_k(0) - m_k(d_k))}{m_k(0) - m_k(d_k)} \leq \frac{O(\sigma_k^2)}{-\Delta_0 \|d_k\|} \to 0
\]

Thus, for sufficiently large \( k \in K_2 \), we get
\[
\frac{C_k - f(x_k + \tilde{d}_k)}{m_k(0) - m_k(d_k)} \geq \frac{f(x_k) - f(x_k + \tilde{d}_k)}{m_k(0) - m_k(d_k)} \geq \mu
\]
(25)

This is a contradiction with (23). Hence, there exists no infinite subset of \( K \) such that (20) holds. Therefore, the proof is completed.

**Theorem 7**: Suppose that all conditions of Theorem 3.6 hold and \( q_k \) satisfies (7). Then Algorithm 1 either stops finitely or generates an infinite sequence \( \{x_k\} \) such that
\[
\lim_{k \to \infty} \|g_k\| = 0
\]

Proof. If Algorithm 1 stops finitely, the proof is obvious. Otherwise, Theorem 3.6 indicates that Algorithm 1 generates an infinite sequence \( \{x_k\} \) satisfying in (19). Since \( q_k \) satisfies (7), we have
\[
0 \leq \tau \|s_k\| \leq -\frac{\tau}{\|w_k\|} \|s_k\| = -\frac{\tau}{\|w_k\|} \to 0
\]
Therefore, we have \( \lim_{k \to \infty} \|s_k\| = 0 \).

**CONCLUDING REMARKS**

Trust region methods are generally considered to be reliable and effective methods for nonlinear unconstrained optimization, and thus it is worth to improve their structures. In this paper, we combine an adjustable trust region radius with an effective non-monotone technique to propose a novel hybrid non-monotone adjustable trust region method. The radius can be adjusted automatically according to the current iterative information to reduce the total number of iterates and function evaluations.

**ACKNOWLEDGEMENT**

The author would be grateful to the anonymous referees for their valuable comments and suggestions.

**REFERENCES**

China, 45:620-631.