Scholars Bulletin
(A Multidisciplinary Bi-weekly Journal)
An Official Publication of "Scholars Middle East Publishers",
Dubai, United Arab Emirates
Website: http://scholarsbulletin.com/

ISSN 2412-9771 (Print)
ISSN 2412-897X (Online)

A non-monotone self-adaptive trust region method with line search for unconstrained optimization

***Liran Yang*[1], *Qinghua Zhou*[2]**
Key Lab. of Machine Learning and Computational Intelligence,
College of Mathematics and Information Science, Hebei University,
Baoding, 071002, Hebei Province, China

***Corresponding Author:**
Qinghua Zhou
Email: qinghua.zhou@gmail.com

Abstract: In this paper, we propose a new non-monotone self-adaptive trust region method with line search for solving unconstrained optimization problem. Different from the usual trust region methods, our algorithm does not only use a non-monotone technique, but also use a new rule to update the trust region radius. We prove the global convergence of the new algorithm under some reasonable assumptions.
Keywords: non-monotone strategy; self-adaptive trust region method; unconstrained optimization; line search; global convergence.

INTRODUCTION

Consider the following large unconstrained optimization problem:

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

where $f(x): R^n \to R$ is a twice continuously differentiable function. For a given iteration point x_k, line search method has the form defined by computing a step-size α_k in the specific direction d_k and derives a new point as $x_{k+1} = x_k + \alpha_k d_k$. For example, in the Armijo-rule line search method, given $s > 0$, $\beta \in (0,1)$ and $\zeta \in (0,1)$, α_k is the largest α in $\{s, s\beta, s\beta^2, \ldots\}$ such that

$$f(x_k + \alpha_k d_k) \le f_k + \zeta \alpha_k g_k^T d_k \tag{1.2}$$

On the other hand, trust region methods compute a trial step d_k by solving the following quadratic sub-problem:

$$\begin{aligned} \min \quad & q_k(d) = f_k + g_k^T d + \tfrac{1}{2} d^T B_k d, \\ s.t. \quad & \|d\| \le \Delta_k, \end{aligned} \tag{1.3}$$

where $f_k = f(x_k)$, $g_k = \nabla f(x_k)$, $B_k \in R^{n\times n}$ is a symmetric matrix which is the Hessian matrix or its approximation of $f(x)$ at the current point x_k, $\Delta_k > 0$ is the trust radius and $\|\cdot\|$ denotes to the Euclidean norm. The ratio

$$r_k = \frac{f(x_k) - f(x_{k+1})}{q_k(0) - q_k(d_k)}$$

plays a key role to decide whether d_k is acceptable or not and how to adjust the trust region radius. The iteration is said to be successful if $r_k \ge \eta_1$. This leads us to the new point x_{k+1}, and the trust region radius is updated. If not, the iteration is unsuccessful, and the trail point is rejected. Generally, trust region radius update rule can be summarized as follow:

$$\Delta_{k+1} = \begin{cases} [\gamma_1 \Delta_k, \gamma_2 \Delta_k], & if \ r_k < \eta_1; \\ [\gamma_2 \Delta_k, \Delta_k], & if \ \eta_1 \le r_k \le \eta_2; \\ [\Delta_k, \infty), & if \ r_k \ge \eta_2. \end{cases} \tag{1.4}$$

Where the constants γ_1, γ_2, η_1 and η_2 satisfy

$$0 \le \eta_1 < \eta_2 < 1,\ 0 < \gamma_1 \le \gamma_2 < 1 \tag{1.5}$$

In the trust region method, the difficulty is how to adjust the trust region radius. In order to choose an adaptive trust region radius at each iteration, many adaptive trust region methods have been studied in [1, 2].

In 1997, Sartenaer [3] introduced a strategy that can automatically determine the initial trust region radius. The strategy requires additional evaluations of the objective function. Zhang et al. [4] presented another strategy of updating the trust region radius. Their basic idea is originated from the following sub-problem in [5].

$$\begin{aligned} \min \quad & q_k(d) = g_k^T d + \tfrac{1}{2} d^T B_k d, \\ s.t. \quad & -\Delta_k \le d_i \le \Delta_k, \quad i = 1, 2, \ldots, n, \end{aligned} \tag{1.6}$$

where $\Delta_k = c^p (\|g_k\| / \gamma)$, $0 < c < 1$, $\gamma = \min(\|B_k\|, 1)$ and p is a nonnegative integer.

Hei [6] and Walmg et al. [7] proposed self-adaptive update methods, respectively. The trust region radius are product of so-called R-function and the Λ-function, that is, $\Delta_{k+1} = R(r_k)\Delta_k$ and $\Delta_{k+1} = \Lambda(r_k)\Delta_k$. Recently, the L-function is introduced by Lu et al. in [8]. They presented a new self-adaptive trust region method, in which the radius is $\Delta_{k+1} = L(r_k)\Delta_k$.

Grippo et al. [9] firstly proposed a non-monotone line search for Newton's method. This algorithm accepts the step-size α_k whether

$$f(x_k + \alpha_k d_k) \le f(x_{l(k)}) + \beta\alpha_k \nabla f(x_k)^T d, \tag{1.7}$$

where $\beta \in (0, \frac{1}{2})$, $f(x_{l(k)}) = \max_{0 \le j \le m_k} f(x_{k-j}), m_0 = 0, 0 \le m_k \le \min\{m_{k-1} + 1, M\} (k \ge 1),$ and $M \ge 0$ is an integer. It has been proved that the sequence $\{f(x_k)\}$ is not increasing. Since then, many researchers [4-5] have exploited the non-monotone technique. In 1993, Deng et al. in [10] made some changes and applied it to the trust region method, and proposed a non-monotone trust region method for unconstrained optimization. Theoretical analysis and numerical results show that algorithms with non-monotone strategy are more effective than algorithms without it.

Zhang and Hager [11] proposed another non-monotone line search method, they replaced the maximum function value with an average of function values. In detail, their method finds a step-size α_k satisfying the following condition:

$$f(x_k + \alpha_k d_k) \le C_k + \beta\alpha_k \nabla f(x_k)^T d, \tag{1.8}$$

where

$$C_k = \begin{cases} f(x_k), & k = 0, \\ \frac{\eta_{k-1} Q_{k-1} C_{k-1} + f(x_k)}{Q_k}, & k \ge 1, \end{cases} \quad Q_k = \begin{cases} 1, & k = 0, \\ \eta_{k-1} Q_{k-1} + 1, & k \ge 1, \end{cases} \tag{1.9}$$

and

$\eta_{k-1} \in [\eta_{\min}, \eta_{\max}]$, $\eta_{\min} \in [0, 1)$ and $\eta_{\max} \in [\eta_{\min}, 1)$ are two chosen parameters. Numerical results showed that this non-monotone technique was superior to (1.7). Then, this non-monotone was applied to the trust region methods [12, 13].

Inspired by the ideas introduced above, we use the L-function to update the radius, then applied it to the trust region method with non-monotone strategy proposed by Zhang and Hager [11]. The purpose of this paper is to present a new non-monotone adaptive trust region method with line search.

The rest of the paper is organized as follows. In Section 2, we describe L-function and our new non-monotone self-adaptive trust region algorithm. The global convergence properties of this novel algorithm are given in Section 3. Finally, some conclusions are summarized in Section 4.

ALGORITHM DESCRIPTION

The L-function rule proposed by Lu [8] can be described as follows:

$$\Delta_{k+1} = L(r_k)\Delta_k \tag{2.1}$$

where the L -function $L(r_k)$ is chosen as

$$L(r_k)=\begin{cases} c_1+(c_2-c_1)\exp(r_k), & if\ r_k\le 0, \\ \frac{1-\beta_1\exp(\eta_2)}{1-\exp(\eta_2)}-\frac{(1-\beta_1)\exp(\eta_2)}{1-\exp(\eta_2)}\exp((r_k-\eta_2)), & if\ 0<r_k<\eta_2, \\ \beta_2, & if\ \eta_2\le r_k<2-\eta_2, \\ \beta_3+(\beta_2-\beta_3)\exp(-(\frac{r_k+\eta_2-2}{\eta_2-2})^2), & if\ r_k>2-\eta_2. \end{cases} \tag{2.2}$$

where β_1, β_2, β_3, c_1, c_2 and η_2 are constants.

Now describe the new non-monotone self-adaptive trust region algorithm with the new radius update rule. After we obtain d_k, then the ratio r_k is defined by

$$r_k=\frac{Ared_k}{\Pr ed_k}=\frac{C_k-f(x_k+d_k)}{q_k(0)-q_k(d_k)}, \tag{2.3}$$

Algorithm 2.1

Step 1. Given $x_0\in R^n, \Delta_0>0, B_0\in R^{n\times n},\ 0<\eta_1<\eta_2<1,\ 0<c_1<c_2<1,\ \varepsilon\ge 0$, $0<\beta_1\le\beta_3<1\le\beta_2$, set $k:=0$.

Step 2. Compute g_k. If $\|g_k\|\le\varepsilon$, stop. Otherwise, go to Step 3.

Step 3. Solve the sub-problem (1.3) for d_k. Compute C_k, $Ared_k$, $Pred_k$ and r_k.

Step 4. If $r_k>\eta_1$, set $x_{k+1}=x_k+d_k$, go to the Step 6; otherwise, go to Step 5.

Step 5. Select α_k, which is the largest number in $\{1,\beta,\beta^2,\ldots\}$ such that

$$f(x_k+\alpha_k d_k)\le C_k+\zeta\alpha_k g_k^T d_k \tag{2.4}$$

Set $x_{k+1}=x_k+\alpha_k d_k$.

Step 6. Update the trust region radius Δ_{k+1} by (2.1) and (2.2).

Step 7. Compute g_{k+1} and B_{k+1}, and set $k:=k+1$, go to Step 2.

Remark: In order to ease of reference, we define two index sets as below:

$I=\{k|r_k\ge\eta_1\}$ and $J=\{k|r_k<\eta_1\}$.

CONVERGENCE ANALYSIS

In this section, we discuss the global convergence properties of Algorithm 2.1. Before we address some theoretical issues, we would like to give the following assumptions.

(H1) The level set $L(x_0)=\{x\in R^n\,|\,f(x)\le f(x_0)\}$ is bounded for any given $x_0\in R^n$.

(H2) The matrix B_k is uniformly bounded, i.e., there exists a constant $M_0>0$, such that, for all k, $\|B_k\|\le M_0$.

(H3) There exists a constant $M_1>0$, such that $\|\nabla^2 f(x)\|\le M_1$ for all $x\in L(x_0)$.

Lemma 3.1. If d_k is the solution to sub-problem (1.2), then

$$Pred_k=q_k(0)-q_k(d_k)\ge\tfrac{1}{2}\|g_k\|\min\{\Delta_k,\tfrac{\|g_k\|}{\|B_k\|}\}. \tag{3.1}$$

$$g_k^T d_k\le-\tfrac{1}{2}\|g_k\|\min\{\Delta_k,\tfrac{\|g_k\|}{\|B_k\|}\}. \tag{3.2}$$

Proof. From Lemma 13.3.1 in [14], we know (3.1) holds. And from (3.1), we can see

$$Pred_k=-g_k^T d_k-\tfrac{1}{2}d_k^T B_k d_k\ge\tfrac{1}{2}\|g_k\|\min\{\Delta_k,\tfrac{\|g_k\|}{\|B_k\|}\}$$

Consider the above inequality and the fact $d_k^T B_k d_k>0$, (3.2) obviously holds. Therefore, the lemma is true.

Lemma 3.2. Let $\{x_k\}$ be the sequence generated by Algorithm 2.1. For any fixed $k\ge 0$, we have

$$f_{k+1}\le C_{k+1}\le C_k \tag{3.3}$$

Proof. Let $k \geq 0$ be an arbitrary fixed integer.

If $k \in I$, i.e., $r_k \geq \eta_1$. By the definition of r_k and $r_k > \eta_1$, we have

$$C_k - f_{k+1} \geq \eta_1 Pred_k = \eta_1 (q_k(0) - q_k(d_k))$$
$$\geq \tfrac{\eta_1}{2} \|g_k\| \min\{\Delta_k, \tfrac{\|g_k\|}{\|B_k\|}\} \geq 0$$

If $k \in J$, i.e., $r_k < \eta_1$. From (2.4), we have

$$C_k - f_{k+1} \geq -\zeta \alpha_k g_k^T d_k \tag{3.4}$$

The inequality together with the fact that $g_k^T d_k \leq 0$ yields $C_k - f_{k+1} \geq 0$

Thus, $C_k \geq f_{k+1}$. Then, by the definition of C_k, we obtain that

$$C_k = \tfrac{\eta_{k-1} Q_{k-1} C_{k-1} + f_k}{Q_k} \geq \tfrac{\eta_{k-1} Q_{k-1} f_k + f_k}{Q_k} = f_k .$$

So

$$C_k \geq f_k . \tag{3.5}$$

On the other hand, we have

$$C_{k+1} = \tfrac{\eta_k Q_k C_k + f_{k+1}}{Q_{k+1}} \leq \tfrac{\eta_k Q_k C_k + C_k}{Q_{k+1}} = C_k . \tag{3.6}$$

From (3.5) and (3.6), Lemma 3.2 holds.

Lemma 3.3. Suppose that the sequence $\{x_k\}$ is generated by Algorithm 2.1. The algorithm is well defined.

Proof . The process is similar to Lemma 2.3 in [16].

Lemma 3.4. Suppose that (H1)-(H2) hold. Then

$$|f(x_k + d_k) - q(d_k)| = O(\|d_k\|^2) \tag{3.7}$$

Proof. From the Taylor expansion, the lemma is true.

Lemma 3.5. (See Lemma 3.6 in [15]) Suppose that Assumption 1 holds, and there is a positive number $\tau > 0$ such that $\|g_k\| \geq \tau$ for all k, then there exists a $\overline{\Delta} > 0$, such that for all k, we have $\Delta_k \geq \overline{\Delta}$.

Lemma 3.6. Suppose that the sequence $\{x_k\}$ is generated by Algorithm 2.1. Then, for all $k \in J$, the step-size α_k satisfies

$$\alpha_k > \min\{\frac{\beta}{2}, \frac{\beta(1-\zeta)M_0}{M_1}\} \tag{3.8}$$

Proof. Let $\alpha = \frac{\alpha_k}{\beta}$. If $\alpha_k > \frac{\beta}{2}$, (3.8) is obvious. We only consider the situation when $\alpha_k \leq \frac{\beta}{2}$. Then, from Step 5 of Algorithm 2.1, we have

$$C_k + \zeta \alpha g_k^T d_k < f(x_k + \alpha d_k) \tag{3.9}$$

From Taylor expansion, we get

$$f(x_k + \alpha d_k) = f_k + \alpha g_k^T d_k + \tfrac{1}{2} \alpha^2 d_k^T \nabla^2 f(\xi_k) d_k \tag{3.10}$$

From (3.9), (3.10) and (H3), we obtain

$$f_k + \zeta \alpha g_k^T d_k \leq C_k + \zeta \alpha g_k^T d_k \leq f_k + \alpha g_k^T d_k + \tfrac{1}{2} \alpha^2 M_1 \|d_k\|^2 \tag{3.11}$$

where $\xi_k \in (x_k, x_k + \frac{\alpha_k}{\beta} d_k)$. Therefore, we have

$$-(1-\zeta) g_k^T d_k < \tfrac{1}{2} \alpha M_1 \|d_k\|^2 \tag{3.12}$$

From the definition of the $Pred_k$, we get

$$\tfrac{1}{2} d_k^T B_k d \leq -g_k^T d_k \tag{3.13}$$

Considering (3.12) and (3.13), we obtain

$$(1-\zeta) d_k^T B_k d < \frac{M_1}{\beta} \alpha_k \|d_k\|^2 \tag{3.14}$$

Thus,

$$\alpha_k > \frac{\beta(1-\zeta)d_k^T B_k d_k}{\|d_k\|^2 M_1} \geq \frac{\beta(1-\zeta)M_0}{M_1}$$

The proof is completed.

Theorem 3.7. Suppose that (H1)-(H2) hold. Let the sequence $\{x_k\}$ generated by Algorithm 2.1, then we have

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{3.15}$$

Proof. The proof is analogous to Theorem 3.7 in [15].

CONCLUSIONS

In this paper, we propose a new non-monotone self-adaptive trust region method with line search. It is useful to take advantage of new adaptive strategy and non-monotone technique which can be implemented easily. Under some mild conditions, we proved the global convergence result of the proposed method.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61473111) and the Natural Science Foundation of Hebei Province (Grant No. A2014201003, A2014201100).

REFERENCES

1. Fan, J. Y., Ai, W. B. & Zhang Q. Y. (2004). A line search and trust region algorithm with trust region radius converging to zero. *Journal of Computational Mathematics*, 22(6), 865–872.
2. Gertz, E. M. (2004). A quasi-Newton trust-region method, *Mathematical Programming Series A*, 100, 447–470.
3. Sartenaer (1997). Automatic determination of an initial trust region in nonlinear programming, *SIAM Journal on Scientific Computing*, 18(6), 1788–1803.
4. Zhang, X. J, Zhang, L. Liao, (2002). An adaptive trust region method and its convergence, *Science in China A.* 45(5), 620–631.
5. Zhang, X. (1997).Trust region method in neural network, *Acta Mathematicae Applicatae Sinica*, 12, 1–10.
6. Hei, L. (2003). A self-adaptive trust region algorithm, *Journal of Computational Mathematics*, 21(2), 229–236.
7. Walmag, J. M. B & E. J. M. Delhez. (2004). A note on trust-region radius update, *SIAM Journal on Optimization*, 16(2), 548–562.
8. Lu, Y., Li, W., Cao, M., & Yang, Y. (2014). A novel self-adaptive trust region algorithm for unconstrained optimization. *Journal of Applied Mathematics*, *2014*.
9. Grippo, L., Lampariello, F., & Lucidi, S. (1986). A nonmonotone line search technique for Newton's method. *SIAM Journal on Numerical Analysis*, *23*(4), 707-716.
10. Deng, N, Xiao, Y., & Zhou, F. (1993). Nonmontonic trust region algorithm. *Journal of Optimization Theory and Application*, 76(2), 259-285.
11. Zhang, H., & Hager, W. W. (2004). A nonmonotone line search technique and its application to unconstrained optimization. *SIAM Journal on Optimization*, *14*(4), 1043-1056.
12. Mo, J., Liu, C., & Yan, S. (2007). A nonmonotone trust region method based on nonincreasing technique of weighted average of the successive function values. *Journal of Computational and Applied Mathematics*, *209*(1), 97-108.
13. Qing-Jun, W. (2010). Nonmonotone trust region algorithm for unconstrained optimization problems. *Applied Mathematics and Computation*, *217*(8), 4274-4281.
14. Yuan, Y., & Sun, W. (1997). Optimization theory and methods.
15. Zhou, Q., Chen, J., & Xie, Z. (2014). A nonmonotone trust region method based on simple quadratic models. *Journal of Computational and Applied Mathematics*, *272*, 107-115.
16. Ahookhosh, M., Amini, K., & Peyghami, M. (2012). A non-monotone adaptive trust region line search method for large-scale unconstrained optimization. *Applied Mathematical Modelling*, 36(1), 478-487.