International Journal of Recent Engineering Science (IJRES),
ISSN: 2349-7157, *Volume 2 Issue 3 May to June 2015*

A Non-monotone Self-adaptive Trust Region Method with Line Search Based on Simple Quadratic Models

Liran Yang, Qinghua Zhou*
College of Mathematics and Information Science, Hebei University, China

Abstract:

In this paper, we propose a new non-monotone adaptive trust region method with line search for solving unconstrained optimization problem. Unlike the traditional trust region methods, our new algorithm combine non-monotone adaptive trust region strategy with a scale approximation of the objective function's Hessian. Theoretical analysis indicates that the new method preserves the global convergence under some mild conditions.

Keywords- *Armijo-type line search, global convergence, non-monotone strategy, trust region method, unconstrained optimization*

I. INTRODUCTION

Consider the following large unconstrained optimization problem:

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

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

$$f(x_k + \alpha_k d_k) \le f_k + \zeta \alpha_k g_k^T d_k \qquad (2)$$

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

$$\begin{aligned} \min \quad & q_k(d) = f_k + g_k^T d + \tfrac{1}{2} d^T B_k d \\ s.t. \quad & \|d\| \le \Delta_k \end{aligned} \qquad (3)$$

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 called the trust radius and $\|\cdot\|$ refers to the Euclidean norm. The ratio ρ_k between the actual reduction $f(x_k) - f(x_{k+1})$ and the predicted reduction $q_k(0) - q_k(d_k)$ plays a key role to decide whether d_k is acceptable or not and how to adjust the trust region radius. The trial step is accepted whenever ρ_k is greater than a positive constant μ_1. This leads us to the new point $x_{k+1} = x_k + \alpha_k d_k$, and the trust region radius is updated. Otherwise, the trust region radius must be diminished and the sub-problem (3) be solved again. Many authors have studied the self-adaptive trust region method [2, 7, 19]. In [17], a new self-adaptive adjustment strategy for updating the trust region radius was proposed. That is, given $0 \le \mu_1 \le \mu_2 < 1$, $c_1 > 1$, $0 < c_2 < 1$, set

$$\Delta_{k+1} = \theta_{k+1} \|g_{k+1}\| \|B_{k+1}^{-1}\| \quad (4)$$

where

$$\theta_{k+1} = \begin{cases} c_1\theta_k, & \text{if } r_k > \mu_2; \\ c_2\theta_k, & \text{if } r_k < \mu_2; \\ \theta_k, & \text{if } \mu_1 \le r_k \le \mu_2. \end{cases}$$

In 1986, Grippo et al. [3] proposed a non-monotone line search for Newton's method. This algorithm accepts the step-size α_k whether

$$f(x_k + \alpha_k d_k) \le f(x_{l(k)}) + \beta\alpha_k \nabla f(x_k)^T d, \quad (5)$$

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 [4-6] have exploited the non-monotone technique and a lot of numerical tests have showed that the non-monotone technique proposed by Grippo et al. [3] is efficient at some extent. In 1993, Deng et al. in [1] 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. From then on a variety of the non-monotone trust region methods have been presented [7, 9].

Although the non-monotone technique has many advantages, however, it has some disadvantages too. The iterations may not satisfy the condition (5) for sufficiently large k, for any fixed bound M on the memory. Zhang and Hager [8] also pointed out that the numerical results are dependent on the choice of parameter M in some cases. In order to overcome these disadvantages, Zhang and Hager [8] 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:

$$f(x_k + \alpha_k d_k) \le C_k + \beta\alpha_k \nabla f(x_k)^T d \quad (6)$$

where

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

$$Q_k = \begin{cases} 1, & k = 0, \\ \eta_{k-1} Q_{k-1} + 1, & k \ge 1, \end{cases}$$

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 (5). Then, this non-monotone was applied to the trust region methods [9, 10]. In 2012, M. Ahookhosh et al. [16] introduced another non-monotone strategy. They replaced C_k in (6) with D_k

$$D_k = \eta_k f_{l(k)} + (1 - \eta_k) f_k \quad (7)$$

for $\eta_{k-1} \in [\eta_{\min}, \eta_{\max}]$. This non-monotone technique is efficient and robust which is showed by numerical experiments in [16].

The key problem is how to solve the trust region sub-problem (3) for the trust region method. Many efficient methods for sub-problem (3) have been proposed [1, 2, 7]. However, when the scale of problem (1) is large, these methods may be too slow because all these methods have to store a symmetric matrix B_k and the algorithms are complicated relatively.

A diagonal-sparse quasi-Newton method, which replaces the scalar matrix with the diagonal matrix, was proposed in [12]. Based on the diagonal-sparse quasi-Newton method, Sun et al. [13] developed a non-monotone trust region algorithm with simple quadratic models, in which the approximation of Hessian matrix in the sub-problem is a diagonal positive definite matrix. It is obvious that the memory requirements and computational complexity for estimating B_k are low.

Inspired by the ideas introduced above, we use the new scale approximation of the minimizing function's Hessian in the trust region sub-problem, and then combine it with the non-monotone strategy proposed by M. Ahookhosh et al. [16]. The purpose of this paper is to present a new non-monotone adaptive trust region method with line search based on simple quadratic models.

This paper is organized as follows. In Section 2, we describe our new non-monotone self-adaptive trust region method with line search. The properties of this new algorithm and the global convergence theory are given in Section 3. Finally, some concluding remarks are given in Section 4.

II. NEW NON-MONOTONE ADAPTIVE TRUST REGION METHOD WITH LINE SEARCH

If we give the initial point x_0, then f_0 and g_0 can be computed. Suppose I is the $n\times n$ identity matrix and set $B_0 = I$. We can get the next iteration point $x_1 = x_0 + d_0$. Suppose that $x_k\ (k\geq 1)$ have been obtained. We compute the approximation of the Hessian of the function f at x_k. From $x_k = x_{k-1} + d_{k-1}$ we have $x_{k-1} = x_k - d_{k-1}$. By the Taylor's theorem, we can obtain

$$\begin{aligned} & f(x_{k-1}) \\ &= f(x_k - d_{k-1}) \\ &\approx f(x_k) - g_k^T d_{k-1} + \tfrac{1}{2} d_{k-1}^T \nabla^2 f(x_k) d_{k-1}, \end{aligned} \tag{8}$$

We consider $\gamma(x_k)I$ as an approximation of $\nabla^2 f(x_k)$, where $\gamma(x_k)\in R$. And the $\gamma(x_k)$ can be expressed as

$$\gamma(x_k) = \begin{cases} \dfrac{2\varphi}{d_{k-1}^T d_{k-1}}, & if\ \varphi > 0, \\ \dfrac{2\delta}{d_{k-1}^T d_{k-1}}, & otherwise. \end{cases} \tag{9}$$

where $\varphi = f(x_{k-1}) - f(x_k) + g_k^T d_{k-1}$.

So, the sub-problem (3) can be modified as

$$\begin{aligned} \min\ & q_k(d) = f_k + g_k^T d + \tfrac{1}{2}\gamma(x_k) d^T d \\ s.t.\ & \|d\| \leq \Delta_k \end{aligned} \tag{10}$$

The sub-problem (10) can be solved easily. In fact, if $\left\|-\frac{1}{\gamma(x_k)} g_k\right\| \leq \Delta_k$, set $d_k = -\frac{1}{\gamma(x_k)} g_k$; otherwise d_k of sub-problem (10) is the solution of the following problem [15]:

$$\begin{aligned} \min\ & q_k(d) = f_k + g_k^T d + \tfrac{1}{2}\gamma(x_k) d^T d \\ s.t.\ & \|d\| = \Delta_k \end{aligned} \tag{11}$$

By solving (11), we can compute the solution $d_k = -\frac{\Delta_k}{\|g_k\|} g_k$.

After obtaining d_k, then the ratio ρ_k is computed by

$$\rho_k = \frac{Ared_k}{\Pr ed_k} = \frac{D_k - f(x_k + d_k)}{q_k(0) - q_k(d_k)} \tag{12}$$

Algorithm 2.1

Step 1. Given $x_0 \in R^n$, $\Delta_0 > 0, 0 < \zeta < 1$,

$0 < c_2 < 1 < c_1$, $0 \le u_1 < u_2 < 1, 0 < \beta < 1$,

$\delta > 0$, $0 < \varepsilon < 1$, $0 \le \eta_{\min} < \eta_{\max} < 1$,

$\sigma > 0$, set $k = 0$, $B_0 = I$, $\theta_0 = 1$. Choose parameters $\eta_{\min} \in [0,1)$ and $\eta_{\max} \in [\eta_{\min}, 1)$.

Step 2. Compute g_k. If $\|g_k\| = 0$, stop. Otherwise, go to Step 3.

Step 3. Solve the sub-problem (10) for d_k. Compute D_k, $Ared_k$, $Pred_k$ and ρ_k.

Step 4. If $\rho_k \ge \mu_1$, set $s_k = d_k$, $x_{k+1} = x_k + s_k$ and

$$\theta_{k+1} = \begin{cases} c_1\theta_k, & if\ \rho_k > \mu_2; \\ \theta_k, & if\ \mu_1 \le \rho_k \le \mu_2, \end{cases}$$

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

$$f(x_k + \alpha_k d_k) \le D_k + \zeta \alpha_k g_k^T d_k \qquad (13)$$

Set $s_k = \alpha_k d_k$, $x_{k+1} = x_k + s_k$, $\theta_{k+1} = c_2\theta_k$.

Step 6. Compute $\gamma(x_{k+1})$. If $\gamma(x_{k+1}) \le \varepsilon$ or $\gamma(x_{k+1}) \ge \frac{1}{\varepsilon}$, set $\gamma(x_{k+1}) = \sigma$. Let

$\Delta_{k+1} = \dfrac{\theta_{k+1}}{\gamma(x_{k+1})}\|g_{k+1}\|$, and set $k = k+1$, go to Step 2.

It is obvious that for all k,

$$0 < \min(\varepsilon, \sigma) \le \gamma(x_k) \le \max(\tfrac{1}{\varepsilon}, \sigma) \qquad (14)$$

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

$$I = \{k \mid \rho_k \ge u_1\} \text{ and } J = \{k \mid \rho_k < u_1\}.$$

III. CONVERGENCE ANALYSIS

In this section, we will prove the global convergence property of Algorithm 2.1. The following assumptions are necessary to analyze the convergence property.

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

(H2) There exists a constant $M_0 > 0$, such that $\|\nabla^2 f(x)\| \le M_0$ for all $x \in L(x_0)$.

(H3) The matrix $\gamma(x_k)I$ is uniformly bounded, i.e., there exists a constant $M_1 > 0$, such that, for all k, $\|\gamma(x_k)I\| \le M_1$.

Lemma 3.1. If d_k is the solution to sub-problem (10), then

$$\begin{aligned} Pred_k &= q_k(0) - q_k(d_k) \\ &\ge \tfrac{1}{2}\|g_k\|\min\{\Delta_k, \tfrac{\|g_k\|}{\gamma(x_k)}\} \end{aligned} \qquad (15)$$

$$g_k^T d_k \le -\tfrac{1}{2}\|g_k\|\min\{\Delta_k, \tfrac{\|g_k\|}{\gamma(x_k)}\} \qquad (16)$$

Proof. From Lemma 3.2 in [14], we know (15) holds. And from (15), we can see

$$\begin{aligned} Pred_k &= -g_k^T d_k - \tfrac{1}{2}\gamma(x_k) d_k^T d \\ &\ge \tfrac{1}{2}\|g_k\|\min\{\Delta_k, \tfrac{\|g_k\|}{\gamma(x_k)}\} \end{aligned}$$

Consider the above inequality and the fact $\gamma(x_k) d_k^T d > 0$, (16) 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

$$f_{k+1} \le D_{k+1} \tag{17}$$

Proof. Let $k \ge 0$ be an arbitrary fixed integer. By the definition of D_k and $f_{l(k)}$, we have

$$\begin{aligned} D_{k+1} &= \eta_{k+1} f_{l(k+1)} + (1-\eta_{k+1}) f_{k+1} \\ &\ge \eta_{k+1} f_{k+1} + (1-\eta_{k+1}) f_{k+1} \\ &= f_{k+1} \end{aligned} \tag{18}$$

From (18), 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 the sequence $\{x_k\}$ is generated by Algorithm 2.1. Then, for all $k \in J$, the step-size α_k satisfies

$$\alpha_k > \min\{\frac{\beta}{2}, \frac{\beta(1-\zeta)\gamma(x_k)}{M_0}\} \tag{19}$$

Proof. Let $\alpha = \frac{\alpha_k}{\beta}$. If $\alpha_k > \frac{\beta}{2}$, (19) is obvious. We only consider the situation when $\alpha_k \le \frac{\beta}{2}$. Then, from Step 5 of Algorithm 2.1, we have

$$D_k + \zeta \alpha g_k^T d_k < f(x_k + \alpha d_k) \tag{20}$$

From Taylor expansion, we get

$$\begin{aligned} & f(x_k + \alpha d_k) \\ &= f_k + \alpha g_k^T d_k + \tfrac{1}{2}\alpha^2 d_k^T \nabla^2 f(\xi_k) d_k \end{aligned} \tag{21}$$

From (20), (21) and (H1), we obtain

$$\begin{aligned} & f_k + \zeta \alpha g_k^T d_k \\ &\le D_k + \zeta \alpha g_k^T d_k \\ &\le f_k + \alpha g_k^T d_k + \tfrac{1}{2}\alpha^2 M_0 \|d_k\|^2 \end{aligned} \tag{22}$$

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

$$-(1-\zeta) g_k^T d_k < \tfrac{1}{2}\alpha M_0 \|d_k\|^2 \tag{23}$$

From the definition of the $Pred_k$, we get

$$\tfrac{1}{2}\gamma(x_k) d_k^T d \le -g_k^T d_k \tag{24}$$

Considering (23) and (24), we obtain

$$(1-\zeta)\gamma(x_k) d_k^T d < \frac{M_0}{\beta}\alpha_k \|d_k\|^2 \tag{25}$$

Thus,

$$\begin{aligned} \alpha_k &> \frac{\beta(1-\zeta)\gamma(x_k) d_k^T d_k}{\|d_k\|^2 M_0} \\ &\ge \frac{\beta(1-\zeta)\gamma(x_k)}{M_0} \end{aligned}$$

The proof is completed.

Lemma 3.5. (See Lemma 2.1 in [16]) Suppose that (H1) holds, the sequence $\{x_k\}$ generated by Algorithm 2.1 is contained in the level set $L(x_0)$ and $\{f_{l(k)}\}$ is a decreasing sequence.

Lemma 3.6. (See Lemma 3.2 in [16]) Suppose that all conditions of Lemma 3.4 hold. Then, we have

$$\lim_{k\to\infty} f(x_{l(k)}) = \lim_{k\to\infty} f(x_k) \tag{26}$$

Corollary 3.7. Suppose that the sequence $\{x_k\}$ generated by Algorithm 2.1. Then we obtain

$$\lim_{k\to\infty} D_k = \lim_{k\to\infty} f(x_k) \tag{27}$$

Proof. By the definition of D_k and Lemma 3.2, we have

$$f_k \le D_k \le f_{l(k)} \tag{28}$$

Then, by using Lemma 3.6, we complete the proof.

Lemma 3.8. Suppose that all conditions of Lemma 3.4 hold. Assume that the sequence $\{x_k\}$ does not converge to a stationary point, i.e., there exists a constant $0 < \tau < 1$ such that for all k , we have $\|g_k\| \ge \tau$. Then, we have

$$\lim_{k\to\infty} \min\{\Delta_k \frac{\tau}{\gamma(x_k)}\} = 0 \quad (29)$$

Proof. There exists a constant φ such that

$$f_{k+1} \le D_k - \varphi \min\{\Delta_k \tfrac{\tau}{\gamma(x_k)}\} \quad (30)$$

If $k \in I$, i.e., $\rho_k \ge \mu_1$, we have

$$\begin{aligned} & f_{k+1} - D_k \\ & \le -\mu_1 \operatorname{Pr} ed_k \\ & \le -\tfrac{1}{2}\mu_1 \|g_k\| \min\{\Delta_k, \tfrac{\|g_k\|}{\gamma(x_k)}\} \\ & \le -\tfrac{1}{2}\mu_1 \|g_k\| \min\{\Delta_k, \tfrac{\tau}{\gamma(x_k)}\} \\ & = \varphi_1 \min\{\Delta_k, \tfrac{\tau}{\gamma(x_k)}\}. \end{aligned} \quad (31)$$

If $k \in J$, i.e., $\rho_k < \mu_1$. From (13) (16) and (20), we have

$$\begin{aligned} & f_{k+1} \\ & \le D_k + \zeta \alpha_k g_k^T d_k \\ & \le D_k - \zeta \alpha_k \|g_k\| \min\{\Delta_k, \tfrac{\|g_k\|}{\gamma(x_k)}\} \\ & \le D_k - \frac{\beta\zeta(1-\zeta)\gamma(x_k)}{M_0} \min\{\Delta_k, \tfrac{\tau}{\gamma(x_k)}\} \\ & = D_k - \varphi_2 \min\{\Delta_k, \tfrac{\tau}{\gamma(x_k)}\} \end{aligned} \quad (32)$$

Set $\varphi = \min\{\varphi_1, \varphi_2\}$, we can conclude that (30) holds. Combining with Corollary 3.7, it completes the proof.

Lemma 3.9. Suppose that the sequence $\{x_k\}$ generated by Algorithm 2.1. If there exists a positive constant $\tau > 0$, such that $\|g_k\| \ge \tau$ for all k hold, there exists a nonnegative integer p, such that $\rho_{k+p} \ge \mu_1$.

Proof. The process is similar to the same proof of Lemma 9 in [18].

Theorem 3.10. Suppose that (H1)-(H3) and all conditions of Lemma 3.8 hold. Let the sequence $\{x_k\}$ generated by Algorithm 2.1, then we have

$$\liminf_{k\to\infty} \|g_k\| = 0 \quad (33)$$

Proof. We assume that Formula (33) is not true, that is, there exists a positive constant $\tau > 0$, such that

$$\|g_k\| \ge \tau \quad \text{for all } k \quad (34)$$

By Lemma 3.1, Lemma 3.5, (H3) and Formula (34), we have

$$\begin{aligned} +\infty & > \sum_{k=1}^{\infty} (D_k - f(x_{k+1})) \ge \sum_{k\in I} (D_k - f(x_{k+1})) \\ & \ge \sum_{k\in I} \mu_1 \operatorname{Pr} ed_k \\ & \ge \sum_{k\in I} \tfrac{1}{2}\mu_1 \|g_k\| \min\{\Delta_k, \tfrac{\|g_k\|}{\gamma(x_k)}\} \\ & = \sum_{k\in I} \tfrac{1}{2}\mu_1 \|g_k\| \min\{\tfrac{\theta_k\|g_k\|}{\gamma(x_k)}, \tfrac{\|g_k\|}{\gamma(x_k)}\} \\ & = \sum_{k\in I} \tfrac{1}{2\gamma(x_k)} \mu_1 \|g_k\|^2 \min\{\theta_k, 1\} \\ & \ge \sum_{k\in I} \tfrac{1}{2M_1} \mu_1 \|g_k\|^2 \min\{\theta_k, 1\} \end{aligned}$$

which implies

$$\sum_{k\in I} \min\{\theta_k, 1\} < +\infty \quad (35)$$

It follows that Lemma 3.9 that I is an infinite set. Thus, by Formula (35), we have

$$\lim_{k\in I, k\to\infty} \theta_k = 0$$

(36)

On the other hand, for $k \in I$, we have $\rho_k \ge \mu_1$. Hence, there exists a constant $\theta > 0$, such that $\theta_k > \theta$ holds for sufficiently large $k \in I$, which is a contradiction with Formula (36). Theorem 3.10 has been proved.

IV. CONCLUSIONS

In this paper, we present a new non-monotone self-adaptive trust region method with Armijo-type line search strategy based on simple quadratic models. With the help of line search, new algorithm can reduce the number of the solving sub-problems. And the form of the new method is very simple. 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] N. Deng, Y. Xiao, F. Zhou, Nonmontonic trust region algorithm, Journal of Optimization Theory and Application, 76(2), 1993, 259-285.

[2] J. Fu, W. Sun, Raimundo J. B. De Sampaio, An adaptive approach of conic trust-region method for unconstrained optimization problems, Journal of Applied Mathematics and Computing, 19(1-2), 2005, 165-177.

[3] L. Grippo, F. Lamparillo, S. Lucidi, A nonmonotone line search technique for Newton's method, SIAM J. Numer. Anal., 23(4), 1986, 707-716.

[4] M. Raydan, The Barzilai and Barwein gradient method for large scale unconstrained minimization problem, SIAM Journal on Optimization, 7(1), 1997, 26-33.

[5] W. Sun, Q. Zhou, An unconstrained optimization method using nonmonotone second order Goldstein's line search, Science in China Series A, 50(10), 2007, 1389-1400.

[6] Q. Zhou, W. Sun, A nonmonotone second order line search method for unconstrained optimization, Journal of Computational Mathematics, 25(1), 2007, 104-112.

[7] Q. Zhou, W. Sun, An adaptive nonmonotone trust region method with curvilinear optimization, Journal of Computational Mathematics, 24(6), 2006, 761-770.

[8] H. Zhang, W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, SIAM Journal on Optimization, 14(4), 2004, 1043-1056.

[9] J. Mo, C. Liu, S. Yan, 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), 2007, 97-108.

[10] Q. Wu, Nonmonotone trust region algorithm for unconstrained optimization problems, Applied Mathematics and Computation, 217(8), 2010, 4274-4281.

[11] N. Gu, J. Mo, Incorporating nonmonotone strategies into the trust region method for unconstrained optimization problem, Journal of Computers& Mathematics with Applications, 55(9), 2008, 2158-2172.

[12] Z. Shi, G. Sun, A diagonal-sparse quasi-Newton for unconstrained optimization problem, J. Systems Sci. Math. Sci., 26(1), 2006, 101-112.

[13] Q. Sun, L. Duan, B. Cui, A nonmonotone trust region algorithm with simple quadratic models, J. Systems Sci. Math. Sci., 29 2009, 407-483.

[14] Q. Zhou, J. Chen, Z. Xie, A nonmonotone trust region method based on simple quadratic models, Journal of Computational and Applied Mathematics, 272(1), 2014, 107-115.

[15] W. Sun, Y. Yuan, Optimization theory and methods (Springer, New York, 2006).

[16] M. Ahookhosh, K. Amini, M. Peyghami, A non-monotone adaptive trust region line search method for large-scale unconstrained optimization, Applied Mathematical Modelling, 36(1), 2012, 478-487.

[17] Z. Sang, Q. Sun, A self-adaptive trust region method with line search based on a simple sub-problem model, Journal of Applied Mathematic and Computing, 232(2), 2009, 514-522.

[18] M. Ahookhosh, K. Amini, An efficient non-monotone adaptive trust-region method for unconstrained optimization, Journal Numerical Algorithm, 59(4), 2012, 523-540.

[19] Z. Cui, B. Wu, A new modified nonmonotone adaptive trust region method for unconstrained optimization, Computational Optimization and Application, 53(3), 2012, 795-806.