Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 9(4): 8497–8515.
DOI: 10.3934/math.2024413
Received: 09 January 2024
Revised: 10 February 2024
Accepted: 22 February 2024
Published: 28 February 2024

Research article

An adaptive simple model trust region algorithm based on new weak secant equations

Yueting Yang, Hongbo Wang, Huijuan Wei, Ziwen Gao and Mingyuan Cao*

School of Mathematics and Statistics, Beihua University, Jilin 132013, China

* **Correspondence:** Email: cmy0918@beihua.edu.cn.

Abstract: In this work, we proposed a new trust region method for solving large-scale unconstrained optimization problems. The trust region subproblem with a simple form was constructed based on new weak secant equations, which utilized both gradient and function values and available information from the three most recent points. A modified Metropolis criterion was used to determine whether to accept the trial step, and an adaptive strategy was used to update the trust region radius. The global convergence and locally superlinearly convergence of the new algorithm were established under appropriate conditions. Numerical experiments showed that the proposed algorithm was effective.

Keywords: trust region method; weak secant equation; modified Metropolis criterion; adaptive strategy
Mathematics Subject Classification: 90C06, 90C30

1. Introduction

Consider the following unconstrained optimization problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{1.1}$$

where $f(x)$: $\mathbb{R}^n \to \mathbb{R}$ is a continuous differentiable function. The problem has penetrated deeply into various fields, such as aerospace, engineering technology, economics and finance, etc [1–3]. The trust region method is an important method for solving (1.1) and it has attracted the attention of many researchers [4–6]. The trust region methods usually compute a trial step s_k by solving the following quadratic subproblem

$$\begin{aligned} \min\ & q^{(k)}(x_k + s) = f_k + g_k^T s + \frac{1}{2} s^T B_k s, \\ \text{s.t.}\ & \|s\| \le \Delta_k, \end{aligned} \tag{1.2}$$

where

$$f_k = f(x_k), \quad g_k = \nabla f(x_k),$$

$B_k \in \mathbb{R}^{n\times n}$ is the Hessian matrix of the function at the current iteration point x_k or its symmetric approximation, s is the trial step, $\Delta_k > 0$ is the trust region radius, and $\|\cdot\|$ stands for the Euclidean norm. The trust region methods take the ratio of the actual reduction to the predicted reduction

$$r_k = \frac{f(x_k) - f(x_k + s_k)}{q^{(k)}(x_k) - q^{(k)}(x_k + s_k)}$$

to decide whether to accept the trial step and how to adjust the trust region radius. If r_k is close to 1, the trial step s_k should be accepted and Δ_k can be increased. If r_k is too small, the trial step s_k should be rejected, Δ_k should be decreased, and the subproblem (1.2) should be resolved. If r_k is much larger than 1, the case of 'too successful iteration' might occur. In addition, the trust region methods have always been accepted as effective methods for dealing with small and medium scale optimization problems due to the cost of computation and storage on the matrix B_k^{-1} at each iteration. Many researchers [7–16] considered the modification of the trust region methods to adapt large scale optimization. We devote to the construction of subproblem and the adjustment of the trust region radius.

In recent years, some trust region methods [11–15] based on simple models for solving large-scale optimization problems were proposed. For example, Sun et al. [13] developed a nonmonotone trust region algorithm with simple quadratic models, in which Hessian matrix in the subproblem is a diagonal positive definite matrix. Li et al. [14] proposed a simulated annealing-based trust region Bazilai-Borwein (BB) method and [15] proposed nonmonotone trust region BB methods. They all used scalar matrix with the reciprocal of the BB-stepsize to approximate the Hessian matrix of the objective function $f(x)$. In the above methods, the amount of computation and storage is greatly reduced.

The matrix B_k in subproblem (1.2) usually satisfies the classic secant equation (see [17])

$$B_{k+1}s_k = y_k, \tag{1.3}$$

where

$$s_k = x_{k+1} - x_k, \quad y_k = g_{k+1} - g_k.$$

Ebadi et al. [18] provided two new secant equations

$$B_{k+1}\bar{s}_k = \bar{y}_k, \quad \bar{s}_k = \frac{3}{2}s_k - \frac{1}{2}s_{k-1}, \quad \bar{y}_k = y_k - \frac{1}{3}y_{k-1} + \frac{\nu_k}{\|\bar{s}_k\|^2}\bar{s}_k, \tag{1.4}$$

where

$$\nu_k = 2(f_k - f_{k+1}) + \bar{s}_k^T\left(\frac{4}{3}g_k - \frac{1}{3}g_{k-1}\right) + \frac{1}{2}(s_k + s_{k-1})^T g_{k+1},$$

so

$$B_{k+1}\bar{s}_k = z_k, \quad z_k = y_k - \frac{1}{3}y_{k-1} + \frac{\eta_k}{\|\bar{s}_k\|^2}\bar{s}_k, \tag{1.5}$$

where

$$\eta_k = 2f_k - \frac{1}{2}f_{k-1} - \frac{3}{2}f_{k+1} + \nu_k.$$

Constructing the approximation of the Hessian matrix based on the formulas (1.4) and (1.5) can make the algorithm maintain the third order curvature information of the objective function at the current iteration point, and it can make use of both gradient and function values and information from the three most recent points. It would improve the efficiency of the algorithm, which has attracted our attention. We try to introduce these two secant equations into the trust region algorithm.

It is usually difficult to satisfy the secant Eq (1.3) with a nonsingular scalar matrix. Many researchers [16, 19, 20] considered some alternative conditions that could maintain the accumulated curvature information along the negative gradient. For example, Dennis and Wolkowicz [20] introduced a weaker form by projecting the secant Eq (1.3) in the direction s_k as follows

$$s_k^T B_{k+1} s_k = s_k^T y_k. \tag{1.6}$$

Zhou et al. [16] considered some generalization of the weak secant Eq (1.6) and proposed a new simple model trust region method with generalized BB parameter. Inspired by the above work, we try to introduce two new weak secant equations with more information.

Updating strategy of the trust region radius may significantly affect the number of iterations. Many researchers proposed adaptive trust region methods [21–26] to adjust the trust region radius. For example, Zhang et al. [24] proposed an adaptive trust region radius

$$\Delta_k = c^p \left\| \widehat{B}_k^{-1} \right\| \|g_k\|,$$

where $c \in (0, 1)$, p is a nonnegative integer, and

$$\widehat{B}_k = B_k + iI$$

is a positive definite matrix, for some $i \in N$. Under the same parameters, Shi et al. [25] proposed another adaptive trust region radius

$$\Delta_k = -c^p \frac{g_k^T q_k}{q_k^T \widehat{B}_k q_k} \|q_k\|,$$

with the vector parameter $q_k \in \mathbb{R}^n$ satisfying the angle condition

$$-\frac{g_k^T q_k}{\|g_k\| \cdot \|q_k\|} \geq o,$$

where $o \in (0, 1)$. Rezaee and Babaie-Kafaki [26] proposed an adaptive choice for the trust region radius based on an eigenvalue analysis conducted on the scaled memoryless quasi-Newton updating formulas

$$\Delta_k = 2c^p \|g_k\| \begin{cases} 1, & \text{if } k = 0, \\ \frac{\|s_{k-1}\|^2}{|s_{k-1}^T y_{k-1}|}, & \text{if } k > 0, \end{cases} \tag{1.7}$$

where

$$s_{k-1} = x_k - x_{k-1} \text{ and } y_{k-1} = g_k - g_{k-1}.$$

The adaptive trust region radius does not use the Hessian matrix explicitly, and comparing the trust region method with the adaptive radius to some adaptive trust region methods, this method is low-cost to update the trust region radius.

Note that some trust region methods require monotone reduction of the objective function, which may slow the convergence rate in the presence of a narrow curved valley. Nonmonotone trust region methods [14, 27–30] were proposed. Li et al. [14] proposed a simulated annealing-based trust region BB method. Their nonmonotone strategy was defined by a modified Metropolis criterion, which can dynamically control the acceptance probability of the solution of the subproblem by introducing adaptive parameters (such as temperature parameters) into the accept-reject strategy. In the early stage of iteration, a higher acceptance probability can help the algorithm jump out of the local optimal solution by a larger extent, while in the later stage of iteration, the acceptance probability was gradually reduced to converge to a better solution.

Our research aims to propose an adaptive simple model trust region algorithm based on new weak secant equations for solving large-scale optimization problems. The contributions of our work are listed as follows:

- Two new weak secant equations are introduced, which make use of both gradient and function values and utilize information from the three most recent points. A simple trust region subproblem is also constructed.
- In order to enable the algorithm to accept more trial steps, the nonmonotone strategy is defined by a modified Metropolis criterion.
- To overcome the case of "too successful iteration", adaptive strategy is introduced to adjust the trust region radius.

The rest of this paper is organized as follows. An adaptive simple trust region algorithm based on new weak secant equations is proposed in the next section. In Section 3, the global convergence and locally superlinearly convergence of the new algorithm are established under mild assumptions. Section 4 introduces numerical experiments to prove the effectiveness of the algorithm. Conclusions are made in the last section.

2. The proposed method

In this section, two new weak secant equations are introduced based on the formulas (1.4) and (1.5). On this basis, a simple trust region subproblem is constructed and a new trust region method for solving large-scale optimization problems is proposed.

Based on the formulas (1.4) and (1.5), we introduce the following new weak secant equations

$$\bar{s}_k^T B_{k+1} \bar{s}_k = \bar{s}_k^T \bar{y}_k, \tag{2.1}$$

$$\bar{s}_k^T B_{k+1} \bar{s}_k = \bar{s}_k^T z_k, \tag{2.2}$$

where $\bar{s}_k$, $\bar{y}_k$, and z_k are the same as formulas (1.4) and (1.5). Let the matrix B_k in subproblem (1.2) be a scalar matrix $\gamma_k I$ satisfying (2.1) or (2.2), then we get

$$\gamma_k = \frac{\bar{s}_{k-1}^T \bar{y}_{k-1}}{\|\bar{s}_{k-1}\|^2} \tag{2.3}$$

or

$$\gamma_k = \frac{\bar{s}_{k-1}^T z_{k-1}}{\|\bar{s}_{k-1}\|^2}. \tag{2.4}$$

A simple trust region subproblem could be constructed as follows

$$\begin{aligned} \min\ & m^{(k)}(x_k + s) = f_k + g_k^T s + \frac{1}{2} s^T \gamma_k I s, \\ \text{s.t.}\ & \|s\| \le \Delta_k. \end{aligned} \tag{2.5}$$

Suppose $\|g_k\| \neq 0$. The solution of subproblem (2.5) is given by:

(i) If

$$\left\| -\frac{g_k}{\gamma_k} \right\| \le \Delta_k,$$

then

$$s_k = -\frac{g_k}{\gamma_k}.$$

(ii) If

$$\left\| -\frac{g_k}{\gamma_k} \right\| > \Delta_k,$$

then

$$s_k = -\frac{\Delta_k}{\|g_k\|} g_k.$$

The ratio r_k can be rewritten as

$$r_k = \frac{f(x_k) - f(x_k + s_k)}{m^{(k)}(x_k) - m^{(k)}(x_k + s_k)}. \tag{2.6}$$

In our algorithm, the following modified Metropolis criterion is used to determine whether to accept the trial step

$$p_k = \begin{cases} 1, & \text{if } r_k > \tau, \\ \exp\left\{-\frac{\tau - r_k}{T_k}\right\}, & \text{otherwise,} \end{cases} \tag{2.7}$$

where T_k is the temperature at the k-iteration, $0 < \tau < 1$ is a sufficiently small real number, and the temperature T_k decreases to 0 as $k \to \infty$. The modified Metropolis criterion is embedded into the trust region methods. Combine p_k and r_k to determine whether the algorithm is iterative. Different from the traditional trust region algorithm, when $r_k \le \tau$, the modified Metropolis criterion is used to accept more iterations with a certain probability, thereby reducing the amount of computation and improving the convergence rate of the algorithm.

Based on the above analysis, we propose an adaptive simple model trust region algorithm (ASMTR) with new weak secant equations.

ASMTR algorithm

Step 0. Set $\varepsilon > 0, \beta \in (0, 1)$, $u \in (0, 1)$, $0 < \tau < u < 1$, $v > 1$, $c \in (0, 1)$, $p = 0$, $0 < \kappa_1 \le \kappa_2$. Give $x_0 \in \mathbb{R}^n$, compute g_0, and set $s_0 = -g_0$, $x_1 = x_0 + s_0$. $T_1 > 0$, $\Delta_1 > 0$, $\gamma_1 = 1$, and $k := 1$.

Step 1. Compute g_k. If $\|g_k\| < \varepsilon$, then stop.

Step 2. If

$$\left\| -\frac{g_k}{\gamma_k} \right\| > \Delta_k,$$

then

$$s_k = -\frac{\Delta_k}{\|g_k\|} g_k,$$

otherwise

$$s_k = -\frac{g_k}{\gamma_k}.$$

Step 3. Compute r_k and p_k by (2.6) and (2.7), respectively, and let

$$l_k := e^{-v} + (e^{-1/v} - e^{-v}) \times \text{rand}(1).$$

If $p_k > l_k$, then

$$x_{k+1} = x_k + s_k, \tag{2.8}$$

otherwise

$$x_{k+1} = x_k. \tag{2.9}$$

Step 4. Compute γ_{k+1} by (2.3) or (2.4). If $\gamma_{k+1} \leq \kappa_1$, set $\gamma_{k+1} = \kappa_1$. If $\gamma_{k+1} \geq \kappa_2$, set $\gamma_{k+1} = \kappa_2$.

Step 5. If $r_k > u$, set $p = 0$; Otherwise, set $p = p + 1$. Compute Δ_{k+1} by (1.7).

Step 6. Set $T_{k+1} = \beta T_k$, $k := k + 1$, and go to Step 1.

Remark 2.1. The formula (1.7) is used to update the trust region radius, i.e.,

$$\Delta_{k+1} = 2c^p \|g_{k+1}\| \begin{cases} 1, & \text{if } k = 0, \\ \frac{\|s_k\|^2}{|s_k^T y_k|}, & \text{if } k > 0. \end{cases}$$

Remark 2.2. From Step 4 of the ASMTR algorithm, we know that the sequence $\{\gamma_k\}$ is uniformly bounded, i.e.,

$$0 < \kappa_1 \leq \gamma_k \leq \kappa_2, \quad \forall k. \tag{2.10}$$

3. Convergence analysis

In this section, we will discuss the convergence of the ASMTR algorithm. We would like to make the following assumptions:

Assumption (i) Let f: $\mathbb{R}^n \to \mathbb{R}$ be twice continuously differentiable and bounded below on the level set

$$L(x_0) = \{x | f(x) \leq f(x_0)\}$$

for $\forall x_0 \in \mathbb{R}^n$.

(ii) Let the gradient $g(x)$ be uniformly continuous on a compact convex set so that Ω contains the level set $L(x_0)$.

Assumptions (i) and (ii) mean that $\left\|\nabla^2 f(x)\right\|$ is continuous and uniformly bounded on Ω, so there exists a positive constant L such that

$$\left\|\nabla^2 f(x)\right\| \leq L, \quad \forall x \in \Omega.$$

Therefore, from the mean value theorem, we have

$$\|g(x) - g(y)\| \leq L \|x - y\|, \ \forall x, y \in \Omega,$$

which ensures that $g(x)$ is Lipschitz continuous on Ω.

Lemma 1. *Suppose that s_k is the solution of subproblem* (2.5) *and the sequence* $\{x_k\}$ *is generated by the ASMTR algorithm. If* $\|g_k\| \neq 0$*, then*

$$pred\,(s_k) = m^{(k)}(x_k) - m^{(k)}(x_k + s_k) \geq \frac{1}{2}\delta_1 \|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\gamma_k}\right\},$$

where $\delta_1 \in (0, 1]$.

Proof. The proof is similar to the proof of Lemma 4.1 in [16]. □

Lemma 2. *Suppose that Assumptions* (*i*) *and* (*ii*) *hold. The solution s_k of the simple model* (2.5) *satisfies*

$$\left|f(x_k + s_k) - m^{(k)}(x_k + s_k)\right| \leq \frac{1}{2}(\kappa_2 + L)\|s_k\|^2.$$

Proof. According to the second-order Taylor expansion, we have

$$f(x_k + s_k) = f_k + s_k^T g_k + \int_0^1 s_k^T [g(x_k + t s_k) - g(x_k)] dt.$$

By the definition of $m^{(k)}(x_k + s)$ in (2.5), we get

$$\begin{aligned}
\left|f(x_k + s_k) - m^{(k)}(x_k + s_k)\right| &= \left|f_k + s_k^T g_k + \int_0^1 s_k^T [g(x_k + t s_k) - g(x_k)] dt - f_k - g_k^T s_k - \frac{1}{2} s_k^T \gamma_k s_k\right| \\
&= \left|\frac{1}{2} s_k^T \gamma_k s_k - \int_0^1 s_k^T [g(x_k + t s_k) - g(x_k)] dt\right| \\
&\leq \frac{1}{2}\kappa_2 \|s_k\|^2 + \left|\int_0^1 s_k^T [g(x_k + t s_k) - g(x_k)] dt\right| \\
&\leq \frac{1}{2}\kappa_2 \|s_k\|^2 + \frac{1}{2} L \|s_k\|^2 = \frac{1}{2}(\kappa_2 + L)\|s_k\|^2.
\end{aligned}$$

We complete the proof. □

Lemma 3. *Suppose that Assumption* (*i*) *holds. Let the sequence* $\{x_k\}$ *be generated by the ASMTR algorithm, then there exists a sufficiently large* $N > 0$ *such that*

$$f_{k+1} \leq f_k - \frac{\tau}{4}\delta_1 \| g_k \| \max\left\{\Delta_k, \frac{\| g_k \|}{\gamma_k}\right\}$$

holds for all $k > N$.

Proof. If $r_k \geq \tau > 0$, then $p_k = 1 > l_k$ holds. By Lemma 1, we have

$$m^{(k)}(x_k) - m^{(k)}(x_k + s_k) \geq \frac{1}{2}\delta_1 \|g_k\| \min\left\{\Delta_k, \frac{\|g_k\|}{\gamma_k}\right\}$$

and

$$\begin{aligned} f(x_{k+1}) = f(x_k + s_k) &\leq f_k + \tau\left(m^{(k)}(x_k + s_k) - m^{(k)}(s_k)\right) \\ &\leq f_k - \frac{\tau}{2}\delta_1 \parallel g_k \parallel \max\left\{\Delta_k, \frac{\parallel g_k \parallel}{\gamma_k}\right\}. \end{aligned}$$

If $r_k < \tau$, then we accept $x_k + s_k$ as the new iterate point when

$$p_k = \exp\left\{-\frac{\tau - r_k}{T_k}\right\} > l_k. \tag{3.1}$$

By Step 6 of the ASMTR algorithm, it can be deduced that

$$\lim_{k\to\infty} T_k = 0.$$

Combining with $l_k \in \left[e^{-\nu}, e^{-1/\nu}\right]$, we have

$$-\nu \leq \ln l_k \leq -\frac{1}{\nu},$$

then, for a given $\forall \varepsilon_1 > 0$, there exists $N > 0$ such that

$$0 < -T_k \ln l_k < \varepsilon_1$$

holds for all $k > N$. By a simple manipulation on (3.1), we get

$$r_k > \tau + T_k \ln l_k > \tau - \varepsilon_1,$$

then by the definition of r_k, there is

$$f_{k+1} < f_k + (\tau - \varepsilon_1)\left(m^{(k)}(x_k + s_k) - m^{(k)}(x_k)\right).$$

Set $\varepsilon_1 = \frac{\tau}{2}$. According to Lemma 1, we have

$$f_{k+1} \leq f_k - \frac{\tau}{4}\delta_1 \parallel g_k \parallel \max\left\{\Delta_k, \frac{\parallel g_k \parallel}{\gamma_k}\right\}.$$

We complete the proof. □

Lemma 4. *Suppose that Assumptions (i) and (ii) hold. Let the sequence $\{x_k\}$ be generated by the ASMTR algorithm. If x_k is not the solution of the problem, i.e., $\|g_k\| \neq 0$, then the iteration (2.9) will be terminated at a finite step.*

Proof. By contradiction that there is $K_1 > 0$ such that the iteration (2.9) will cycle infinitely for $k \geq K_1$, then $\|g_k\| \geq \varepsilon$. Thus, by Step 3 of the ASMTR algorithm, we have $p_k < l_k$ for all $k > K_1$, so

$$r_k < \tau + T_k \ln l_k < \tau - \frac{T_k}{v} < \tau < u. \tag{3.2}$$

By Step 5 of the ASMTR algorithm, we have

$$\lim_{k\to\infty} \Delta_k = 0. \tag{3.3}$$

From Lemmas 1 and 2, we get

$$\begin{aligned} \left|\frac{f(x_k) - f(x_k + s_k)}{pred(s_k)} - 1\right| &= \left|\frac{[f(x_k) - f(x_k + s_k)] - \left[m^{(k)}(x_k) - m^{(k)}(x_k + s_k)\right]}{pred(s_k)}\right| \\ &\leq \frac{(\kappa_2 + L)\|s_k\|^2/2}{\delta_1 \varepsilon \min\{\Delta_k, \varepsilon/\kappa_2\}/2} = \frac{(\kappa_2 + L)\|\Delta_k\|^2}{\delta_1 \varepsilon \min\{\Delta_k, \varepsilon/\kappa_2\}}. \end{aligned} \tag{3.4}$$

Combining (3.3) and (3.4), we obtain

$$\lim_{k\to\infty} r_k = 1.$$

This implies that, for arbitrarily given $\varepsilon_2 > 0$, there exists $K_2 > 0$ such that $r_k > 1 - \varepsilon_2$ holds for $\forall k > K_2$. Since $0 < \tau < u < 1$, by letting $0 < \varepsilon_2 < 1 - u$, we get

$$r_k > u > \tau - \frac{T_k}{v}. \tag{3.5}$$

Taking $K = \max\{K_1, K_2\}$, we have that (3.2) and (3.5) simultaneously hold for $\forall k > K$, leading to a contradiction. □

Theorem 1. *Let the sequence $\{x_k\}$ be generated by the ASMTR algorithm, then we have*

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

Proof. If the ASMTR algorithm terminates in a finite step, then the conclusion is obviously valid. Consider an infinite number of successful iterations. According to Assumption (i), it is known that the sequence $\{f(x_k)\}$ is bounded, i.e., there is an $a \in \mathbb{R}$ such that $f(x_k) \geq a$ holds for all k.

It is obtained from Lemma 3 that

$$f_{k+1} \leq f_k - \frac{\tau}{4}\delta_1 \|g_k\| \max\left\{\Delta_k, \frac{\|g_k\|}{\gamma_k}\right\}.$$

By (2.10), we have

$$\max\left\{\Delta_k, \frac{\|g_k\|}{\gamma_k}\right\} \geq \frac{\|g_k\|}{\kappa_2},$$

so

$$f_{k+1} \leq f_k - \frac{\tau\delta_1}{4\kappa_2}\|g_k\|^2. \tag{3.6}$$

Adding (3.6) with respect to k yields

$$\frac{\tau\delta_1}{4\kappa_2}\sum_{k=N}^{N+K}\|g_k\|^2 \le f_N - f_{N+K+1}. \tag{3.7}$$

Noting that $f_{N+K+1} > a$ for any $K > 0$ and taking limit on both sides of (3.7) as $K \to \infty$, we have

$$\lim_{K\to\infty}\sum_{k=N}^{N+K}\|g_k\|^2 < \infty,$$

which deduces

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

We complete the proof. □

Theorem 2. *Suppose that Assumptions (i) and (ii) hold. Also, assume the ASMTR algorithm generates an infinite sequence $\{x_k\}$ converging to the optimal solution x^*, where the matrix $\nabla^2 f(x^*)$ is positive definite and $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of x^*. If the following condition holds*

$$\lim_{k\to\infty}\frac{\left\|g_k + \nabla^2 f(x^*)s_k\right\|}{\|s_k\|} = 0, \tag{3.8}$$

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

Proof. Since $\nabla^2 f(x^*)$ is positive definite and $\nabla^2 f(x)$ is continuous in a neighborhood of x^*, there exist positive scalars h and ψ such that

$$s^T\nabla^2 f(x)s \ge h\|s\|^2, \quad \forall s \in \mathbb{R}^n, \tag{3.9}$$

for all

$$x \in \widetilde{\Omega} = \{x|\,\|x - x^*\| \le \psi\}.$$

Also, there exists positive integer $\bar{k}$ such that $x_k \in \widetilde{\Omega}$, for all $k \ge \bar{k}$.

From the Taylor expansion and inequality (3.9), for sufficiently large indices k, we have

$$s_k^T y_k = s_k^T(g_{k+1} - g_k) = s_k^T\nabla^2 f(x_k + \zeta_k s_k)s_k \ge h\|s_k\|^2,$$

for some $\zeta_k \in (0,1)$. So, from (1.7), we get

$$\|s_k\| \le 2c^p\frac{\|s_{k-1}\|^2}{\left|s_{k-1}^T y_{k-1}\right|}\|g_k\| \le \frac{2c^p}{h}\|g_k\|, \tag{3.10}$$

where $c \in (0,1)$. Considering Lemma 4, p is finite in each iteration.

On the other hand, from the Taylor expansion, we have

$$\begin{aligned} g_{k+1} &= g_k + \nabla^2 f(x_k + \varsigma_k s_k)s_k \\ &= g_k + \nabla^2 f(x^*)s_k + (\nabla^2 f(x_k + \varsigma_k s_k) - \nabla^2 f(x^*))s_k, \end{aligned}$$

for some $\varsigma_k \in (0,1)$. Thus,

$$\|g_{k+1}\| \le \left\|g_k + \nabla^2 f(x^*)s_k\right\| + \left\|\nabla^2 f(x_k + \varsigma_k s_k) - \nabla^2 f(x^*)\right\| \cdot \|s_k\|.$$

Dividing both sides by $\|s_k\|$, we get

$$\frac{\|g_{k+1}\|}{\|s_k\|} \le \frac{\left\|g_k + \nabla^2 f(x^*)s_k\right\|}{\|s_k\|} + \left\|\nabla^2 f(x_k + \varsigma_k s_k) - \nabla^2 f(x^*)\right\|.$$

So, from Lipschitz continuity of $\nabla^2 f(x)$ on $\widetilde{\Omega}$ and (3.8), we have

$$\lim_{k\to\infty} \frac{\|g_{k+1}\|}{\|s_k\|} = 0,$$

which, from (3.10), yields

$$\lim_{k\to\infty} \frac{\|g_{k+1}\|}{\|g_k\|} = 0,$$

implying that the sequence $\{x_k\}$ converges to x^* superlinearly. □

4. Numerical experiments

In the current section, we show the numerical performance of the ASMTR algorithm. The test problems are unconstrained problems from CUTEr (a widely used testing environment for optimization software) library [31] and Andrei [32, 33]. All codes are written on MATLAB R2015b and run on PC with a 1.19 GHz central processing unit (CPU) processor with 8.00 GB RAM memory. We write two new algorithms as

(1) ASMTR1: the ASMTR algorithm with

$$\gamma_{k+1} = \frac{\bar{s}_k^T \bar{y}_k}{\|\bar{s}_k\|^2}.$$

(2) ASMTR2: the ASMTR algorithm with

$$\gamma_{k+1} = \frac{\bar{s}_k^T z_k}{\|\bar{s}_k\|^2}.$$

Two new algorithms are compared with the following two algorithms. The first is the simulated annealing-based trust region BB method (SATRBB) [14], whose nonmonotone technique is defined by the modified Metropolis criterion; the second is the nonmonotone trust region BB methods (NTBB) [15]. The parameters are given by: $T_1 = 200$, $v = 10, \beta = 0.99$, $\tau = 0.1$, $c_1 = 0.25$, $c_2 = 0.75$, $\Delta_1 = 1$, and $\varepsilon = 10^{-4}$ for the SATRBB algorithm, $\gamma = 0.1$, $\eta_1 = 0.25$, $\eta_2 = 0.75$, and $M = 5$ for the NTBB algorithm, and $u = 0.15$, $c = 0.5$, $\kappa_1 = 2$, $\kappa_2 = 100$, and $\gamma_1 = 1$ for the ASMTR algorithm.

All test algorithms are terminated when satisfying condition

$$\|g_k\| \le 10^{-4}.$$

In addition, the algorithm is stopped if the number of iterations exceeds 500. In such a case, we claim fail of this algorithm. The values x_0, x_0^1, x_0^2, x_0^3, x_0^4, and x_0^5 in the second column associate with starting points with x_0, $10x_0$, $-10x_0$, $100x_0$, $-100x_0$, and $-x_0$, where x_0 is the same as [32, 33]. The notations used in numerical results include the dimension of the problem (n), the initial point of the problem (x_0), the number of iterations (k), the CPU time cost in seconds, the number of function evaluations (nf), and the number of gradient evaluations (ng). The sign "–" means the algorithm fails because the number of iterations exceeds 500. Next, we present some of the numerical results in Examples 4.1–4.4.

Example 4.1. Consider the Broyden banded function

$$f(x) = \sum_{i=1}^{n} (2x_i + 5x_i^3 + 1)^2 - \sum_{j \in J_i} (x_j + x_j^2)^2,$$

where n is the variable,

$$J_i = \{j : j \neq i, \max(1, i - m_l) \leq j \leq \min(n, i + m_u)\},$$

$m_l = 5$, and $m_u = 1$. The initial point of this function is $x_0 = (-1, \cdots, -1)^T$, and the results are listed in Table 1.

Perform numerical experiments on the Broyden banded function for different initial points. Table 1 shows that ASMTR2 needs fewer iterations, function, and gradient evaluations. ASMTR1 and ASMTR2 are superior to SATRBB and NTBB.

Table 1. Numerical result of the Broyden banded function.

		SATRBB				NTBB				ASMTR1				ASMTR2			
n	x_0	k	CPU	nf	ng	k	CPU	nf	ng	k	CPU	nf	ng	k	CPU	nf	ng
10	x_0	142	0.0010	282	143	131	0.0156	260	126	161	0.0010	320	162	111	0.0010	221	112
	x_0^1	237	0.0010	472	238	242	0.0156	482	233	208	0.0156	414	209	149	0.0010	297	150
	x_0^2	146	0.0010	290	147	145	0.0156	288	146	84	0.0156	166	85	82	0.0010	163	83
	x_0^3	328	0.0156	654	328	333	0.0156	664	324	242	0.0469	464	243	228	0.0156	453	229
	x_0^4	237	0.0156	472	238	236	0.0156	470	237	129	0.0313	256	130	127	0.0156	253	128

Example 4.2. Consider the Penalty function I

$$f(x) = 10^{-5} \sum_{i=1}^{n} (x_i - 1)^2 + (\sum_{i=1}^{n} x_i{}^2 - 0.25)^2,$$

where n is the variable. The initial point of this function is $x_0 = (1, 2, \cdots, n)^T$, and the results are listed in Table 2.

Table 2. Numerical result of the Penalty function I.

		SATRBB				NTBB				ASMTR1				ASMTR2			
n	x_0	k	CPU	nf	ng	k	CPU	nf	ng	k	CPU	nf	ng	k	CPU	nf	ng
10	x_0	57	0.0010	112	58	57	0.0010	112	58	30	0.0010	58	31	27	0.0010	53	28
20		72	0.0781	142	73	72	0.0156	142	73	37	0.0010	72	38	32	0.0010	63	33
50		92	0.0010	182	93	92	0.0156	182	93	45	0.0010	88	46	41	0.0010	81	42
100		108	0.0010	214	109	108	0.0010	214	109	52	0.0010	102	53	48	0.0010	95	49

In the case of four dimensions, numerical experiments are performed from the same initial point. We find that ASMTR1 uses less iterations, function, and gradient evaluations than SATRBB and NTBB, and ASMTR2 is better than ASMTR1.

Example 4.3. Consider the Broyden tridiagonal function

$$f(x) = (3x_1 - 2x_1^2)^2 + \sum_{i=2}^{n-1} (3x_i - 2x_i^2 - x_{i-1} - 2x_{i+1} + 1)^2 + (3x_n - 2x_n^2 - x_{n-1} + 1)^2,$$

where n is the variable. The initial point of function is $x_0 = (-1, \cdots, -1)^T$, and the results are listed in Table 3.

Table 3. Numerical result of the Broyden tridiagonal function.

		SATRBB				NTBB				ASMTR1				ASMTR2			
n	x_0	k	CPU	nf	ng	k	CPU	nf	ng	k	CPU	nf	ng	k	CPU	nf	ng
10000	x_0	50	0.1094	98	51	48	0.0625	94	49	40	0.0625	78	41	37	0.0625	73	38
	x_0^1	86	0.2813	170	87	85	0.2656	168	85	56	0.1406	110	57	58	0.1719	115	59
	x_0^2	75	0.1719	148	76	75	0.1406	148	76	63	0.1563	124	64	68	0.1563	135	69
	x_0^3	116	0.2344	230	117	121	0.1563	240	122	70	0.1406	138	71	66	0.1563	131	67
	x_0^4	111	0.3281	220	112	110	0.2188	218	111	76	0.1563	150	77	70	0.1250	139	71
20000	x_0	49	0.1875	96	50	48	0.2344	94	49	51	0.1250	100	49	91	0.3125	181	89
	x_0^1	83	0.2969	164	84	87	0.4063	172	87	51	0.1563	100	52	54	0.2656	107	55
	x_0^2	77	0.3281	152	78	76	0.2500	150	77	69	0.2031	136	70	66	0.2344	131	67
	x_0^3	123	0.4063	244	124	121	0.2969	240	122	76	0.2031	150	77	66	0.2656	131	67
	x_0^4	110	0.5000	218	111	110	0.500	218	111	72	0.1719	142	73	79	0.1719	157	80
50000	x_0	44	0.3906	86	45	47	0.3750	92	48	58	0.3438	114	55	58	0.4844	115	56
	x_0^1	88	0.7500	174	89	87	0.7031	172	88	57	0.5469	112	58	52	0.4375	103	53
	x_0^2	78	0.6875	154	79	78	0.7031	154	79	99	0.7500	196	100	64	0.6719	127	65
	x_0^3	122	0.9531	242	123	121	0.9531	240	122	71	0.5938	140	72	70	0.6563	139	71
	x_0^4	112	1.3281	222	113	111	0.9219	220	112	71	0.5000	140	72	78	0.5625	155	79

For 10000, 20000, and 50000 dimensions, respectively, five initial points are selected to test the numerical results. SATRBB and NTBB win in approximately 13.4% of performed testing problems concerning the number of iterations, and ASMTR1 and ASMTR2 win in nearly 86.6% of performed testing problems. In addition, ASMTR1 and ASMTR2 need shorter CPU time for most problems. This means the new algorithm is very effective for large-scale optimization problems.

Example 4.4. Consider the nearly separable function

$$f(x) = \sum_{i=1}^{n} x_i^2 + \sum_{j=1}^{n} x_j^6 + \cos^2 x_2 + \sum_{i=2}^{n-1} \cos^2 (x_{i-1} + x_{i+1}) + \cos^2 x_{n-1},$$

where n is the variable and $1 \leq j \leq n$. The initial point of this function is

$$x_0 = \left(\frac{n}{2(n+1)}, \frac{n-1}{2(n+1)}, ..., \frac{1}{2(n+1)} \right)^T,$$

and the results are listed in Table 4.

Table 4. Numerical result of the Nearly separable function.

		SATRBB				NTBB				ASMTR1				ASMTR2			
n	x_0	k	CPU	nf	ng	k	CPU	nf	ng	k	CPU	nf	ng	k	CPU	nf	ng
5000	x_0	–	–	–	–	–	–	–	–	147	30.2656	292	132	127	26.5000	364	114
10000		–	–	–	–	–	–	–	–	130	27.5781	258	116	180	118.3750	525	169
20000		–	–	–	–	–	–	–	–	134	103.4375	266	123	142	329.2190	409	129

Table 4 shows the numerical results of the function under different dimensions of the same initial point. SATRBB and NTBB do not run results within 500 iterations, and ASMTR1 and ASMTR2 effectively solved within 180 iterations.

For more insight, we use the performance profiles introduced by Dolan and Moré [34] to illustrate the numerical performance of the four algorithms based on the testing functions in Table 5. The 20 test functions are listed in Table 5, in which the dimensions vary from 2 to 50000. In Table 5, "No." represents the number of the functions. Here, we also add the nonmonotone adaptive trust region method based on the simple conic model nonmonotone adaptive conic trust region (NACTR) method [7] to compare. This method needs less memory and computational efforts.

Table 5. The test functions.

No.	Functions
1	Brown badly scaled function
2	Generalized tridiagonal 1 function
3	Allgower function
4	Brown and Dennis function
5	Boundary value function
6	Staircase S1 function
7	Chebyquad function
8	Staircase S2 function
9	Broyden banded function
10	Broyden tridiagonal function
11	Penalty function I
12	Nearly separable function
13	Schittkowski function 302
14	Variable dimension function
15	Yang tridiagonal function
16	Generalized Rosebrock function
17	Diagonal full Border function
18	DIAG-AUP1 function
19	Separable cubic function
20	LIARWHD function

Based on the numerical results of all the test problems, we present the performance profiles (including the number of iterations, the CPU time, the number of function evaluations, and the number of gradient evaluations). In a performance profile plot, the horizontal axis gives the percentage (τ) of the test problems for which a method is the fastest (efficiency), while the vertical side gives the percentage (ψ) of the test problems that are successfully solved by each of the methods.

Figures 1–4 plot the performance profiles for the number of iterations, the CPU time, the number of function evaluations, and the number of gradient evaluations, respectively. It can be observed that ASMTR1 and ASMTR2 grow up faster than the other algorithms. In a word, they show that the performance of ASMTR1 and ASMTR2 is superior to SATRBB, NACTR, and NTBB in all aspects. In the overall trend, the performance of ASMTR2 is slightly better than ASMTR1. We believe that ASMTR2 is more competitive. Specifically for large-scale problems, the ASMTR algorithm has a strong numerical stability. From above analysis, we can conclude that the algorithm proposed in our work turns out to be quite competitive.

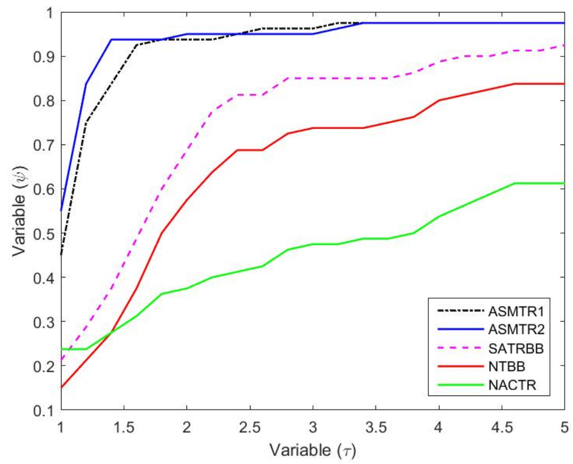

Figure 1. Performance profile of the number of iterations.

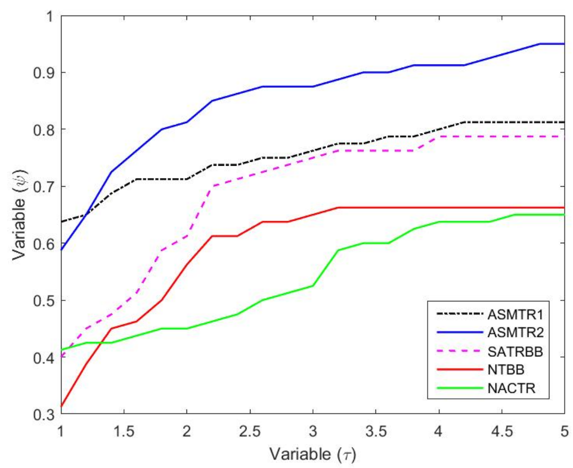

Figure 2. Performance profile of the CPU time.

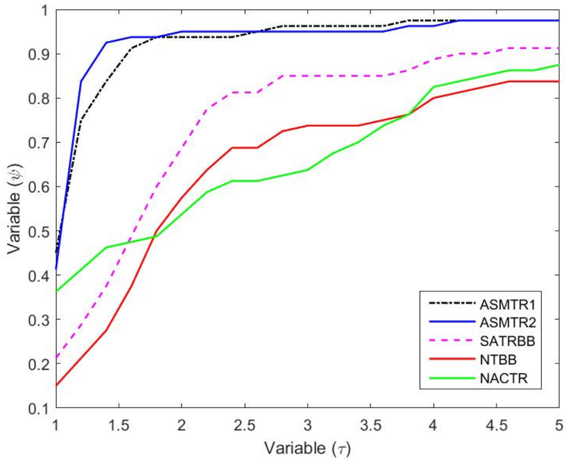

Figure 3. Performance profile of the number of function evaluations.

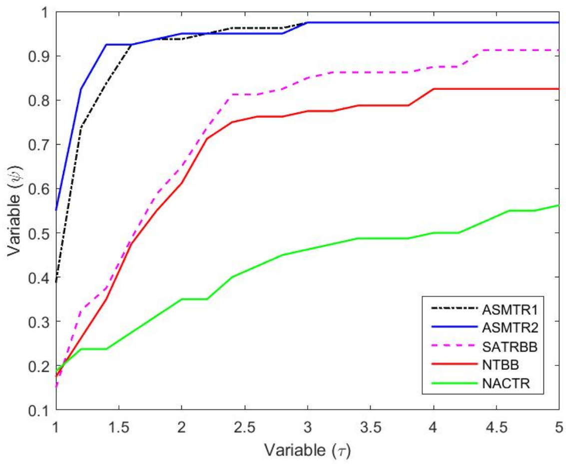

Figure 4. Performance profile of the number of gradient evaluations.

5. Conclusions

In this paper, we propose an adaptive simple model trust region algorithm based on new weak secant equations. It is worth noting that the trust region subproblem of the algorithm is solved more simply in contrast to the many other trust region methods proposed in the literature. We discuss the benefits of constructing a simple model using the last three points of information, and the algorithm combines the nonmonotone strategy defined by a modified Metropolis criterion and adaptive strategy. The global convergence and locally superlinearly convergence of the new algorithm are established under appropriate conditions. Numerical experiments show that the proposed algorithm is effective. There are still many deficiencies in our research; For example, the more efficient and widely used adaptive trust region radius is not taken into account. Therefore, we will explore a new adaptive trust region radius in the future and obtain a more effective and robust method.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The key project of natural science foundation joint fund of Jilin Province (YDZJ202101ZYTS167, YDZJ202201ZYTS303, YDZJ202201ZYTS320, YDZJ202101ZYTS156); The graduate innovation project of Beihua University (2022001, 2022038).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Y. Ji, Y. Ma, The robust maximum expert consensus model with risk aversion, *Inf. Fusion*, **99** (2023), 101866. https://doi.org/10.1016/j.inffus.2023.101866
2. S. Qu, S. Li, A supply chain finance game model with order-to-factoring under blockchain, *Syst. Eng. Theory Pract.*, **43** (2023), 3570–3586. https://doi.org/10.12011/SETP2022-2888
3. Y. Ji, Y. Yuan, Z. Peng, A novel robust flexible minimum cost consensus model with consensus granule, *Group Decis. Negot.*, 2024. https://doi.org/10.1007/s10726-023-09869-3
4. N. Eslami, B. Najafi, S. M. Vaezpour, A trust region method for solving multicriteria optimization problems on riemannian manifolds, *J. Optim. Theory Appl.*, **196** (2022), 212–239. https://doi.org/10.1007/s10957-022-02142-8
5. V. A. Ramirez, G. N. Sottosanto, Nonmonotone trust region algorithm for solving the unconstrained multiobjective optimization problems, *Comput. Optim. Appl.*, **81** (2022), 769–788. https://doi.org/10.1007/s10589-021-00346-8
6. H. H. Dwail, M. A. K. Shiker, Using a trust region method with nonmonotone technique to solve unrestricted optimization problem, *J. Phys. Conf. Ser.*, **1664** (2020), 012128. https://doi.org/10.1088/1742-6596/1664/1/012128
7. L. Zhao, W. Sun, R. J. B. de Sampaio, Nonmonotone adaptive trust region method based on simple conic model for unconstrained optimization, *Front. Math. China*, **9** (2014), 1211–1238. https://doi.org/10.1007/s11464-014-0356-8
8. M. Ahookhosh, K. Amini, M. R. Peyghami, A nonmonotone trust-region line search method for large-scale unconstrained optimization, *Appl. Math. Modell.*, **36** (2012), 478–487. https://doi.org/10.1016/j.apm.2011.07.021
9. X. T. Zhu, M. Xi, W. Y. Sun, A new nonmonotone BB-TR method based on simple conic model for large scale unconstrained optimization, *Numer. Math. A J. Chin. Univ.*, **38** (2016), 173–192.
10. H. Zhu, Q. Ni, J. Jiang, C. Dang, A new alternating direction trust region method based on conic model for solving unconstrained optimization, *Optimization*, **70** (2020), 1555–1579. https://doi.org/10.1080/02331934.2020.1745793

11. Q. Zhou, C. Zhang, A new nonmonotone adaptive trust region method based on simple quadratic models, *J. Appl. Math. Comput.*, **40** (2012), 111–123. https://doi.org/10.1007/s12190-012-0572-x

12. Q. Zhou, J. Chen, Z. Xie, A nonmonotone trust region method based on simple quadratic models, *J. Comput. Appl. Math.*, **272** (2014), 107–115. https://doi.org/10.1016/j.cam.2014.04.026

13. Q. Sun, L. Duan, B. Cui, A nomonotone trust region algorithm with simple quadratic models, *J. Syst. Sci. Math. Sci.*, **29** (2009), 470–483.

14. X. Li, W. Dong, Z. Peng, A new nonmonotone trust region Barzilai-Borwein method for unconstrained optimization problems, *Acta Math. Appl. Sin. Engl. Ser.*, **37** (2021), 166–175. https://doi.org/10.1007/s10255-021-0997-9

15. Y. Liu, X. Liu, Trust region BB methods for unconstrained optimization, *Math. Numer. Sin.*, **38** (2016), 96–112. https://doi.org/10.12286/jssx.2016.1.96

16. Q. Zhou, W. Sun, H. Zhang, A new simple model trust-region method with generalized Barzilai-Borwein parameter for large-scale optimization, *Sci. China Math.*, **59** (2016), 2265–2280. https://doi.org/10.1007/s11425-015-0734-2

17. Q. Z. Yang, *Optimization method*, Beijing: Science Press, 2015.

18. M. J. Ebadi, A. Fahs, H. Fahs, R. Dehghani, Competitive secant (BFGS) methods based on modified secant relations for unconstrained optimization, *Optimization*, **72** (2023), 1691–1701. https://doi.org/10.1080/02331934.2022.2048381

19. J. Zhang, C. Xu, Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations, *J. Comput. Appl. Math.*, **137** (2001), 269–278. https://doi.org/10.1016/S0377-0427(00)00713-5

20. J. E. Dennis, H. Wolkowicz, Sizing and least-change secant methods, *SIAM J. Numer. Anal.*, **30** (1993), 1291–1314. https://doi.org/10.1137/0730067

21. S. Zhao, T. Yan, K. Wang, Y. Zhu, Adaptive trust-region method on Riemannian manifold, *J. Sci. Comput.*, **96** (2023), 67. https://doi.org/10.1007/s10915-023-02288-1

22. S. Lior, E. Yonathan, M. Shie, Adaptive trust region policy optimization: global convergence and faster rates for regularized MDPs, *Proceedings of the AAAI Conference on Artificial Intelligence*, **34** (2020), 5668–5675. https://doi.org/10.1609/aaai.v34i04.6021

23. A. Kamandi, K. Amini, A new nonmonotone adaptive trust region algorithm, *Appl. Math.*, **67** (2022), 233–250. https://doi.org/10.21136/AM.2021.0122-20

24. X. Zhang, J. Zhang, L. Liao, An adaptive trust region method and its convergence, *Sci. China Ser. A*, **45** (2002), 620–631. https://doi.org/10.1360/02ys9067

25. Z. Shi, J. Guo, A new trust region method with adaptive radius, *Comput. Optim. Appl.*, **41** (2008), 225–242. https://doi.org/10.1007/s10589-007-9099-8

26. S. Rezaee, S. Babaie-Kafaki, An adaptive nonmonotone trust region algorithm, *Optim. Methods Software*, **34** (2017), 264–277. https://doi.org/10.1080/10556788.2017.1364738

27. N. Ghalavand, E. Khorram, V. Morovati, Two adaptive nonmonotone trust-region algorithms for solving multiobjective optimization problems, *Optimization*, 2023. https://doi.org/10.1080/02331934.2023.2234920

28. X. Ding, Q. Qu, X. Wang, A modified filter nonmonotone adaptive retrospective trust region method, *PLoS ONE*, **16** (2021), e0253016. https://doi.org/10.1371/journal.pone.0253016

29. M. Yousefi, A. M. Calomardo, A stochastic nonmonotone trust-region training algorithm for image classification, *International IEEE Conference on Signal-Image Technologies and Internet-Based System*, 2022, 522–529. https://doi.org/10.1109/SITIS57111.2022.00084

30. Q. Zhou, D. Hang, Nonmonotone adaptive trust region method with line search based on new diagonal updating, *Appl. Numer. Math.*, **91** (2015), 75–88. https://doi.org/10.1016/j.apnum.2014.12.009

31. N. I. Gould, D. Orban, P. L. Toint, UTEr and SifDec: a constrained and unconstrained testing environment, revisited, *ACM Trans. Math. Software*, **29** (2003), 373–394. https://doi.org/10.1145/962437.962439

32. N. Andrei, Introduction: overview of unconstrained optimization, In: *Nonlinear conjugate gradient methods for unconstrained optimization*, Springer, 2020. https://doi.org/10.1007/978-3-030-42950-8_1

33. N. Andrei, An unconstrained optimization test functions collection, *Adv. Model. Optim.*, **10** (2008), 147–161.

34. E. D. Dolan, J. J. Moré, Benchmarking optimization software with performance profiles, *Math. Program.*, **91** (2002), 201–213. https://doi.org/10.1007/s101070100263

AIMS Press