A modified filter nonmonotone adaptive retrospective trust region method

Xianfeng Ding¹,², Quan Qu¹*, Xinyi Wang¹

¹ School of Sciences, Southwest Petroleum University, Chengdu, China, ² School of Artificial Intelligence, Southwest Petroleum University, Chengdu, China

* 545434091@qq.com

Abstract

In this paper, aiming at the unconstrained optimization problem, a new nonmonotone adaptive retrospective trust region line search method is presented, which takes advantages of multidimensional filter technique to increase the acceptance probability of the trial step. The new nonmonotone trust region ratio is presented, which based on the convex combination of nonmonotone trust region ratio and retrospective ratio. The global convergence and the superlinear convergence of the algorithm are shown in the right circumstances. Comparative numerical experiments show the better effective and robustness.

1. Introduction

Consider the following unconstrained optimization problem

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

where \(f: \mathbb{R}^n \rightarrow \mathbb{R}\) is a twice continuously differentiable function.

Trust region method and line search method are two effective methods to solve unconstrained optimization problems. At present, the both numerical methods mentioned above for solving nonlinear programming are widely used in many applications of various engineering design, automation, transportation, economic analysis, pattern recognition, artificial intelligence, network design and many other modern high-tech research and development.

Compared with the line search method, the trust region method has novel idea, strong convergence and stable numerical performance, see [1–4]. It can not only solve the well-conditioned problem quickly, but also solve the ill-conditioned optimization problem effectively. The basic idea of trust region method as follows: at the iteration point \(x_k\), the trial step \(d_k\) is obtained by solving the subproblem.

$$\min_{d \in \mathbb{R}^n} m_k(d) = g_k^T d + \frac{1}{2} d^T B_k d$$

where \(\nabla f(x_k)\), \(B_k\) is a symmetric approximation of \(\nabla^2 f(x_k)\), \(\Delta_k\) stands for the trust region radius, and \(\|d\|\) shows any vector norm, usually the Euclidean norm.
To evaluate the consistency between the quadratic model and the objective function, the most classical ratio, denoted by $\rho^B_k$, is defined as follows,

$$
\rho^B_k = \frac{f(x_k) - f(x_k + d_k)}{m_k(0) - m_k(d_k)}
$$

The trial step $d_k$ is accepted whenever $\rho^B_k$ is closed to 1, that is to say $x_{k+1} = x_k + d_k$, and $\Delta_k$ is updated suitably. Otherwise, $\rho^B_k$ is negative or positive but not close to 1, $\Delta_k$ is decreased and the subproblem should be resolved.

It is well-known that monotone techniques may not only lead to the rate of convergence slows down, especially in the presence of the narrow curved valley, but also the objective function is required to be decreased at each iteration. Considering this fact, it is most meaningful to study nonmonotone technique for improving algorithm. Obviously, the first nonmonotone technique was the alleged watchdog technique proposed by Chamberlain et al. [5], which was designed to overcome the Maratos effect. At the same time, nonmonotone techniques in [6–8] have attracted extensive attention from scholars, for example, Deng et al. [9] proposed a nonmonotone trust region method by replacing $f(x_k)$ in (3) with $f_{l(k)}$ given by

$$
f_{l(k)}(k) = \max \{ f_{k-1}, \}
$$

where $f_i = f(x_i)$, $m(0) = 0$, $0 \leq m(k) \leq \min\{N, m(k-1)+1\}$, and $N \geq 0$ is an integer constant. Deng et al. [9] proposed another nonmonotone trust region method with the following ratio.

$$
\overline{\rho}_k = \frac{f_{l(k)} - f(x_k + d_k)}{m_k(0) - m_k(d_k)}
$$

Motivated by this, Ahookhoosh et al. [6] proposed a nonmonotone trust region method with the following ratio,

$$
\overline{\rho}_k = \frac{R_k - f(x_k + d_k)}{m_k(0) - m_k(d_k)}
$$

where

$$
R_k = \eta_l f_{l(k)} + (1 - \eta_l) f_k
$$

in which $\eta_l \in [\eta_{\text{min}}, \eta_{\text{max}}]$ with $\eta_{\text{min}} \in [0, 1]$ and $\eta_{\text{max}} \in [\eta_{\text{min}}, 1]$.

After that, Grippo et al. [10] proposed a nonmonotone line search technique for Newton’s method, that is,

$$
f(x_k + \alpha_k d_k) \leq f_{l(k)} + \sigma x_k g_k^T d_k
$$

However, Grippo’s nonmonotone technique has a drawback in that numerical performance is highly dependent on the choice of $N$. It makes more sense, for given $\sigma \in (0,1)$, the step size $\alpha_k$ is chosen so that

$$
f(x_k + \alpha_k d_k) \leq R_k + \sigma x_k g_k^T d_k
$$

As is well-known, an appropriate updating strategy of trust region radius plays a valuable role in a manner that it may prominently affect the computational efficiency. Motivated by this, the versions of trust region radius have attracted considerable attention from many scholars in [11–13]. In order to avoid the gradient or the Hessian information is not precisely employed in the standard trust region, Zhang et al. [14] proposed a new scheme, which use the
following trust region radius,

$$\Delta_k = c \|g_k\| \|B^{-1}_k\|$$  (9)

where $c$ is a constant, $p$ is an adjustment parameter, $B_k = B_k + iI$, $i \in \mathbb{N}$.

Despite effectiveness of the Zhang’s method, calculating an estimation of the inverse of the Hessian in each iteration which results in some additional computational costs. Qu et al. [15] referred to another adaptive strategy for updating the trust region radius as follows,

$$\Delta_k = c \|g_k\|^p$$  (10)

Here, the filter technique introduced by Fletcher and Leyffer [16], which avoids the difficulty of updating the penalty parameter in penalty functions. The filter is able to reject poor trial iterates and enforce global convergence from arbitrary starting points. In this case, it is worth mentioning that Gould et al. [17] proposed an algorithm by using filter technique for unconstrained optimization problems, the main idea is to accept the new iteration point as much as possible.

The filter is primarily composed of a series of gradient of iteration points, the so-called trial step is accepted by the filter, in fact, the corresponding gradient is accepted by the filter. Set $\nabla f(x_k) = g(x_k) = g_k = (g_{k,1}, g_{k,2}, \ldots, g_{k,n})$, $g_{ki} (i = 1, 2, \ldots, n)$ is the i-th component of $g_k$. We say that an iterate $x_1$ dominates $x_2$, whenever

$$|g_{1,i}| \leq |g_{2,i}| \quad \forall \ i = 1, 2, \ldots, n$$

The basic concept of multidimensional filter $\mathcal{F}$ is a list of $n$-tuples of the form $(g_{k,1}, g_{k,2}, \ldots, g_{k,n})$. Suppose that $g_k \in \mathcal{F}, g_l \in \mathcal{F}$, there exists $j \in \{1, 2, \ldots, n\}$ such that

$$|g_{k,j}| \leq |g_{l,j}|$$

Subsequently, compared with [17], in order to maximize the possibility of acceptance of the trial point, we introduced an improved multidimensional filter $\mathcal{F}$, specifically as follows,

Set $\phi_l = \min \{\|g_{l,1}\|, \|g_{l,2}\|, \ldots, \|g_{l,n}\| \in \mathcal{F}\}$, a new trial point $x_k$ is acceptable if there exists $j \in \{1, 2, 3, \ldots, n\}$, such that

$$|g_{k,j}| \leq \max_{1 \leq j \leq n} \|g_{l,j}\| - \gamma_k \phi_l,$$  (11)

When an iteration point $x_k$ is accepted, we add $g_k$ to the filter and Meanwhile and remove the points which are dominated $x_k$ by from the filter.

The rest of the paper is organized as follows. Section 2 is devoted to describe the new filter nonmonotone adaptive retrospective trust region method in details. The global convergence and superlinear convergence are established in Section 3. Some preliminary numerical results are introduced in Section 4. Finally, some conclusions are summarized in Section 5.

2. Materials and methods

In this section, we propose a new filter nonmonotone adaptive retrospective trust region algorithm for solving unconstrained optimization problems. In order to reduce the computational cost, at the iteration point $x_k$, the new nonmonotone ratio and retrospective ratios are
introduced based on [13] as follows:
\[
\rho_k^{\text{NR}} = \frac{R_k - f(x_k + d_k)}{e_k f(x_k) + m_k(0) - m_k(d_k)}
\]
(12)
\[
\rho_{k+1}^{\text{NR}} = \frac{R_k - f(x_k) + d_k}{e_k f(x_{k+1}) + m_{k+1}(0) - m_{k+1}(d_k)}
\]
(13)

where \(e_k \in [0, \eta_k]\). As can be seen from the motivation of this nonmonotone term \(R_k\), the better convergence result can be obtained by freely selecting the parameter \(\eta_k\) and \(e_k\). As a result of above discussion, a new nonmonotone ratio is introduced to improve computational efficiency through the convex combination of \(\rho_k^{\text{NR}}\) and \(\rho_{k+1}^{\text{NR}}\), i.e.
\[
\rho_{k+1}^{\text{NC}} = \gamma \rho_k^{\text{NR}} + (1 - \gamma) \rho_{k+1}^{\text{NR}}
\]
(14)

where \(\gamma \in [\gamma_{\min}, \gamma_{\max}] \subset [0, 1]\). More exactly, \(\rho_k^{\text{NR}}\) is used to determine whether the trial step is acceptable, while \(\rho_{k+1}^{\text{NC}}\) is employed for updating trust region radius.

In the criteria presented earlier, the Hessian matrix is time-consuming to calculate and the convergence rate of the adaptive trust region method will be less efficient. In order to reduce the workload and computational time, some simpler information of known iteration points can be used to reconstruct the regulation formula of trust region radius.

In this way, the trust region radius adjustment formula takes into account the gradient information of the function and the solution of the trust region subproblem, which ensures the calculation accuracy of the algorithm is not reduced. The improved adaptive trust region radius as follows,
\[
\Delta_{k+1} := \tau_{k+1} \frac{\|g_k\|}{\|g_{k+1} - g_k\|}
\]
(15)

Note that \(\tau_k\) is computed by the following formula,
\[
\tau_{k+1} = \begin{cases}
\min\{\beta_2 \tau_k, \tau_{\max}\} & \text{If } \rho_{k+1}^{\text{NC}} \geq \mu_2 \\
\tau_k & \text{If } \mu_1 \leq \rho_{k+1}^{\text{NC}} < \mu_2 \\
\beta_1 \tau_k & \text{If } \rho_{k+1}^{\text{NC}} < \mu_1
\end{cases}
\]
(16)

More formally, the new Algorithm is described as follows.

**Algorithm 2.1** (Nonmonotone Adaptive Filter Retrospective Trust Region Method)

**Step 0.** Give \(x_0 \in \mathbb{R}^n\), \(\Delta_{\max} > 0\), \(B_0 \in \mathbb{R}^{n \times n}\), \(\varepsilon > 0\), \(\gamma_k \in (0, \frac{1}{b})\), \(\epsilon > 0\), \(0 < \mu_1 < \mu_2 < 1\), \(\tau_0 > 0\), \(0 < \beta_1 < 1\), \(\beta_2 > 0\), \(k = 0\).

**Step 1.** If \(k = 0\), set \(\Delta_k = \min\{\tau_0 \|g_k\|, \Delta_{\max}\}\), then go to Step 2.

If \(\rho_{k+1}^{\text{NR}} < \mu_1\), then set \(\tau_k = \beta_1 \tau_{k-1}\) and \(\Delta_k = \beta_1 \tau_{k-1} \frac{\|g_k\|}{\|g_{k+1} - g_k\|}\), go to Step 2.

Else, compute \(\rho_k^{\text{NR}}\) and \(\rho_{k+1}^{\text{NC}}\) by (13) and (14), respectively. \(\tau_k\) is updated by (16) and \(\Delta_k = \min\{\tau_k \frac{\|g_{k+1} - g_k\|}{\|g_k\|}, \Delta_{\max}\}\).

**Step 2.** Compute \(\|g_k\|\). If \(\|g_k\| \leq \epsilon\), then stop.

**Step 3.** Solve the subproblem (2) to find the trial step \(d_k\), set \(x_k^* = x_k + d_k\).

**Step 4.** Compute \(R_k\) and \(\rho_k^{\text{NR}}\), respectively.

**Step 5.** If \(\rho_k^{\text{NR}} \geq \mu_1\), set \(x_{k+1} = x_k^*\). Otherwise, compute \(g_k^* = \nabla f(x_k^*)\), at the same time, if \(x_k^*\) is accepted by the filter \(\mathcal{F}\), then \(x_{k+1} = x_k^*\), add \(g_k^*\) into the \(\mathcal{F}\), otherwise, find the step size \(\alpha_k\) satisfying (8), set \(x_{k+1} = x_k + \alpha_k d_k\).
Step 6. Update the symmetric matrix $B_k$ by (30). Set $k = k+1$, and go to Step 1.

In this way, it is not necessary to precisely solve the subproblem in Algorithm 2.1, as a result, an approximation of $d_k$ satisfies

$$m_k(0) - m_k(d_k) \geq \vartheta \| g_k \| \min \left\{ \Delta_k, \frac{\| g_k \|}{\| B_k \|} \right\}$$

(17)

$$g_k^T d_k \leq -\vartheta \| g_k \| \min \left\{ \Delta_k, \frac{\| g_k \|}{\| B_k \|} \right\}$$

(18)

where $\vartheta \in (0, 1)$.

Assumption 2.1.

1. The level set $L(x_0) = \{ x \in \mathbb{R}^n | f(x) \leq f(x_0) \subseteq \Omega \}$ is closed and bounded. $f(x)$ is a twice continuously differentiable on the level set $L(x_0)$, our claim that in a neighborhood $N$ of $\Omega$, $\nabla f(x)$ is Lipschitz continuous, there exists a positive constant $L$ such that

$$\| \nabla f(x) - \nabla f(y) \| \leq L \| x - y \| \quad \forall x, y \in \Omega$$

2. The matrix $B_k$ is uniformly bounded, i.e., there exists a constant $M_1 > 0$ such that $\| B_k \| \leq M_1$.

3. Convergence analysis

In order to ease of operation, the following index sets are defined as follows:

$D = \{ k | \rho_k^{nn} \geq \mu_k \}$, $A = \{ k | g(x_k^+) \text { or } x_k^+ \text { is added to the filter } \mathcal{F} \}$, $S = \{ k | x_{k+1} = x_k + d_k \}$. Then, $S = \{ k | \rho_k^{nn} \geq \mu_k \text { or } x_k^+ \text { is added to the filter } \mathcal{F} \}$. When $k \notin S$, we have $x_{k+1} = x_k + \alpha_k d_k$.

Lemma 3.1. For all $k$, it makes sense for me that

$$| f_k - f(x_k + d_k) - (m_k(0) - m_k(d_k)) | \leq (O(\| d_k \|)^2)$$

(19)

Proof. Motivated by the Taylor’s expansion and Assumption 2.1, we have

$$f_k - f(x_k + d_k) - (m_k(0) - m_k(d_k))$$

$$= f_k - \left( f_k + g_k^T d_k + \int_0^1 d_k^T [g(x_k + \xi d_k) - g_k] d\xi \right) + g_k^T d_k + \frac{1}{2} d_k^T B_k d_k$$

$$\leq \frac{1}{2} d_k^T B_k d_k + \int_0^1 \| d_k \| \| g(x_k + \xi d_k) - g_k \| d\xi$$

$$\leq \frac{1}{2} \| d_k \|^2 M_1 + L \int \| d_k \| d\xi$$

$$= \frac{1}{2} \| d_k \|^2 M_1 + \frac{1}{2} L \| d_k \|^2$$

$$= O(\| d_k \|^2)$$

This completes the proof of the Lemma 3.1.

Lemma 3.2. Suppose that Assumption 2.1 holds, the sequence $\{ x_k \}$ is generated by Algorithm 2.1. Moreover, assume that there exists a constant $0 < \epsilon < 1$, so that $\| g_k \| > \epsilon$. Then, for any $k$, there exists a nonnegative integer $p$ such that $x_{k+p+1}$ is a successful iteration point, i.e.,

$\rho_{k+p}^{nn} \geq \mu_i$. 


**Proof.** On the contrary, that is, assume that there is an iteration $k$ at which $x_{k+p+1}$ is unsuccessful, for an arbitrary nonnegative integer $p$, we have

$$
\rho_{k+p}^{NB} < \mu_1, \quad p = 0, 1, 2, \ldots
$$

(20)

This clearly implies that

$$
\Delta_{k+p+1} = \beta_1^{p+1} \tau_k \left\| g_{k+p} \right\| \left\| d_{k+p} \right\| \leq \beta_1^{p+1} \tau_{\max} \frac{\left\| g_{k+p} \right\| \left\| d_{k+p} \right\|}{\left\| g_{k+p} - g_{k+p} \right\|}
$$

(21)

Thus, according to $0 < \beta_1 < 1$ and Eq (21), we have

$$
\lim_{p \to \infty} \Delta_{k+p+1} = 0
$$

(22)

Now, according to Lemma 3.1, and Eq (17), we get

$$
\frac{f(x_{k+p}) - f(x_{k+p} + d_{k+p})}{\varepsilon_{k+p} + m_{k+p}(0) - m_{k+p}(d_{k+p})} - 1 \leq \frac{f(x_{k+p}) - f(x_{k+p} + d_{k+p})}{m_{k+p}(0) - m_{k+p}(d_{k+p})} - 1
$$

$$
= \frac{f(x_{k+p}) - f(x_{k+p} + d_{k+p}) - (m_{k+p}(0) - m_{k+p}(d_{k+p}))}{m_{k+p}(0) - m_{k+p}(d_{k+p})}
$$

$$
\leq O(\|d_{k+p}\|^2)
$$

$$
\beta \left\| g_{k+p} \right\| \Delta_{k+p} + \frac{\left\| g_{k+p} \right\|}{\left\| B_{k+p} \right\|}
$$

$$
\leq \frac{O(\|d_{k+p}\|^2)}{\delta \min \left\{ \Delta_{k+p}, \frac{e}{M_1} \right\}}
$$

$$
\leq \frac{O(\Delta_{k+p}^2)}{O(\Delta_{k+p})} \to 0 (p \to \infty)
$$

Following the definition of $R_k$, we get $R_k \geq \eta f_k + (1 - \eta) f_k = f_k$. Thus, for sufficiently large $p$, we have

$$
\frac{R_{k+p} - f(x_{k+p} + d_{k+p})}{\varepsilon_{k+p} + m_{k+p}(0) - m_{k+p}(d_{k+p})} \geq \frac{f(x_{k+p}) - f(x_{k+p} + d_{k+p})}{\varepsilon_{k+p} + m_{k+p}(0) - m_{k+p}(d_{k+p})} \geq \mu_1
$$

(23)

which contradicts (20). This completes the proof of the Lemma 3.3.

**Lemma 3.3.** Suppose that the infinite sequence $\{x_k\}$ is generated by Algorithm 2.1. The number of successful iterations is infinite, that is, $|S| = +\infty$. Then, we have $\{x_k\} \subset L(x_0)$, meanwhile, the sequence $\{f_{k(h)}\}$ is not monotonically increasing and convergent.

**Proof.** The proof follows the process of Lemma 3 and Lemma 4 in [15].

**Lemma 3.4.** Suppose that Assumption 2.1 holds, and there exists a constant $\varepsilon$ such that $\|g_k\| \geq \varepsilon$ for all $k$. Therefore, there is a constant $\nu$ such that

$$
\Delta_k > \nu.
$$

**Proof.** The proof is similar to the proof of Theorem 6.4.3 in [18].
Theorem 3.1. Suppose that Assumption 2.1 holds, and Algorithm 2.1 generates an infinite sequence \( \{x_k\} \) which satisfies
\[
\lim_{k \to \infty} \inf \|g_k\| = 0 \tag{24}
\]

Proof. The standard of proof can be classified into the following both cases.

Case 1. There is no limit to the number of successful iterations, that is \( |S| = +\infty \). Meanwhile, there are many infinitely filter iterations, i.e., \( |A| = +\infty \).

Prove by contradiction, assuming that Eq (24) is not true, then there exists a positive constant \( \varepsilon \) such that \( \|g_k\| > \varepsilon \). On account of Assumption 2.1, \( \{\|g_k\|\} \) is bounded. Set the index in set \( A \) is the sequence \( \{k_i\} \). Therefore, there exists a subsequence \( \{k_t\} \subseteq \{k_i\} \) which satisfies
\[
\lim_{t \to \infty} g_{k_t} = g_\infty, \quad g_\infty \geq \varepsilon \tag{25}
\]

This fact, along with the definition of \( k_n \), leads to \( g(x_{k_t}) \) is accepted by the filter. Then there exists \( j \in \{1,2, \ldots, n\} \), for \( \forall t>1 \), that is to say
\[
|g_{a_{k_t}}| - \max_{1 \leq j \leq n} |g_{a_{k_{t-1}}}| \leq -\gamma_k \Phi_{k_{t-1}} \tag{26}
\]

By the fact that \( 0 < |g_{a_{k_t}}| - \max_{1 \leq j \leq n} |g_{a_{k_{t-1}}}| \leq -\gamma_k \varepsilon < 0 \), as \( t \) is sufficiently large, which is a contradiction.

Case 2. There is no limit to the number of successful iterations, that is \( |S| = +\infty \). Meanwhile, there are many finite filter iterations, i.e., \( |A| = +\infty \).

Contrarily, suppose that the Eq (24) is not true. Then there exists a positive constant \( 0 < \varepsilon < 1 \) such that \( \|g_k\| > \varepsilon \), for all \( k \).

As a consequence of \( |A| = +\infty \), for sufficiently large \( k \in S \), we have \( \mu_k \geq \mu_1 \). Set
\[
\xi_k = |\{p, p + 1, \ldots, k\} \cap S| \]

Following the definition of \( R_k \), Lemma 3.4 and Eq (17), without loss of generality, we have
\[
\sum_{j=p, j \in S}^{k} (f_{ij} - f_{j+1}) \geq \sum_{j=p, j \in S}^{k} (R_j - f_{j+1}) \geq \xi_k \delta \mu_k \varepsilon \min \left\{ v, \frac{\varepsilon}{M_1} \right\} \to \infty \tag{27}
\]

Obviously, when \( p \) is fixed and \( k \to +\infty \), \( \xi_k \delta \mu_k \varepsilon \min \left\{ v, \frac{\varepsilon}{M_1} \right\} \to \infty \) is obtained by \( \xi_k \to +\infty \). It’s not surprising that the left end of Eq (27) has no lower bound. Now, we have
\[
\sum_{j=p, j \neq k}^{k} (f_{ij} - f_{j+1}) \geq \sum_{j=p}^{k} (f_{ij} - f_{j+1})
= f_{ij} - f_{j(k+1)} \tag{28}
\]

It indicates that there is a lower limit on the left end of Eq (28), which is a contradiction with Eq (27).

Theorem 3.2. Suppose that Assumption 2.1 holds, and \( \{x_k\} \) is the sequence generated by Algorithm 2.1 converges to \( x^* \). On the other hand, \( d_k = -B_k^{-1} g_k \) is a solution of the
subproblem, and \( \nabla^2 f(x^*) \) is a positive definite matrix. \( B_k \) satisfies the following condition

\[
\lim_{k \to \infty} \frac{\| (B_k - \nabla^2 f(x^*) d_k) \|}{\| d_k \|} = 0
\] (29)

Then the sequence \( \{x_k\} \) converges to \( x^* \) superlinearly.

**Proof.** The proof follows the same path as given in the proof of Theorem 4.1 in [19].

4. Preliminary numerical experiments

In this section, our purpose is to investigate the computational results of algorithm 2.1 on some middle-large size test problems from Andrei [20]. All algorithms are implemented in MATLAB (R2018a) on a PC Intel(R) Core(TM) i7-4558u CPU @2.80GHz 2.80 GHz 4.00 GB RAM memory and double precision format, the following notations represent the different algorithms.

- ANTRL: ANTRL method is denoted by Ahookhosh et al. in [6];
- NFTR: NFTR method is denoted by [17];
- WA FTR: WAFTR method is expressed by Qu et al. in [15];
- NAFTRT-1: Algorithm 2.1 with \( \Delta_0 = 1 \);
- NAFTRT-2: Algorithm 2.1 with \( \Delta_0 = 10 \);
- NAFTRT-3: Algorithm 2.1 with \( \Delta_0 = 100 \);

As is known to all, BFGS correction is one of the most important methods in quasi-Newton method. Several improved BFGS methods are given in [21, 22], and the convergence theory has been well established. More specifically,

\[
B_{k+1} = B_k - \frac{B_k d_k y_k^m}{d_k^T B_k d_k} + \frac{y_k^m y_k^{mT}}{d_k^T y_k^m}
\] (30)

where \( y_k^m = y_k + \frac{\rho_k}{\| y_k \|} d_k \) and \( \rho_k = 2(f(x_k) - f(x_{k+1}))(g(x_{k+1}) + g(x_k))^T d_k \). It is easy to know that the formula has not only gradient value information, but also function value information.

The parameters of these algorithms are exploited identically as follows: \( \mu_1 = 0.25, \mu_2 = 0.75, N = 5, \beta_1 = 0.25, \beta_2 = 1.5, \Delta_{\text{max}} = 100 \). It is worth mentioning that the stopping criterion is either \( \| g_k \| \leq 10^{-6} \), or the number of iterations exceeds 10,000. As menthoded in [6], set \( \eta_0 = 0.25 \), the updating principle of \( \eta_k \) follows the following formula,

\[
\eta_k = \begin{cases} 
\eta_0/2, & \text{if } k = 1 \\
(\eta_{k-1} + \eta_{k-2})/2, & \text{if } k \geq 2 
\end{cases}
\] (31)

For the sake of simplicity, we draw efficiency comparisons involved in the number of function evaluations \( (n_f) \), the number of gradient function evaluations \( (n_i) \), and the running time (CPU) by using the Dolan–More ´ performance profile, in particular, the particulars are detailed in [24]. To account for this, we can choose a performance index as a comparison metric between the above algorithms. For every \( \tau \geq 1 \), the proportion \( \rho(\tau) \) of the test problems is given by the performance profile. The performance of each considered algorithmic variant was the best within a range of \( \tau \) of the best as well.

In this way, it has to be noted that the selection of initial trust region radius has a great influence in the efficiency of the algorithm. Tables 1 and 2 and Figs 1–3 imply that NAFTRT-2 solves more than 95% of the problems with the minimum number of failures compared with the other two initial radius. At a glance, we choose the initial trust region radius \( \Delta_0 = 10 \), as a given parameter for the following algorithm.
Moreover, as shown in Figs 4–6 and Table 2, the NAFRTR-2 is the best solver, in the field of CPU, \( n_f \) and \( n_i \), about 98% problems respectively. As it is clear that NAFRTR-2 is effective and obtains better performance profiles by comparing with ANTRL, NFTR and WAFTR. Based on the above main observations, the modified trust region method comes out to be fairly effective for unconstrained optimization.
Moreover, as shown in Figs 4–6 and Table 3, the RNATR-A is the best solver, in the field of CPU, \( n_f \) and \( n_i \), about 98% problems respectively. As it is clear that NAFRTR-2 is effective and obtains better performance profiles by comparing with ANTRL, NFTR and WAFTR. Based on the above main observations, the modified trust region method comes out to be fairly effective for unconstrained optimization.

Table 2. The numerical results of algorithm with various initial radius.

| No. | CPU \( f(x) \) | CPU \( f(x) \) | CPU \( f(x) \) |
|-----|----------------|----------------|----------------|
| 1   | 46.637771      | 4.583815e-15  | 36.260390      |
| 2   | 32.602418      | 3.892592e-14  | 21.731327      |
| 3   | 0.312776       | 9.151778e-15  | 0.417474       |
| 4   | 0.138253       | 2.782988e-15  | 0.237082       |
| 5   | 0.316953       | 1.577771e-04  | 0.29693        |
| 6   | 14.254856      | 2.010000e+00  | 15.063743      |
| 7   | 0.254316       | 2.000000e+00  | 0.247084       |
| 8   | 5.763769       | 1.985454e+00  | 74.892615      |
| 9   | 5.451341       | 3.966591e+00  | 0.41747        |
| 10  | 8.688504       | 2.466836e+00  | 4.95965        |
| 11  | 5.865066       | 6.398167e+02  | 2.679197       |
| 12  | 3.361856       | 1.839389e-15  | 5.634865       |
| 13  | 36.22765       | 1.689856e-18  | 13.036732      |
| 14  | 32.856925      | 9.547051e-10  | 47.699712      |
| 15  | 15.867802      | -2.043783e+02 | 9.285441       |
| 16  | 143.85266      | 3.173473e-14  | 169.873944     |
| 17  | -              | -              | 99.263988      |
| 18  | 2083.100452    | -1.000000e-03 | 813.043527     |
| 19  | -              | -              | 5.321657       |
| 20  | 63.320012      | 8.218692e-12  | 120.302857     |
| 21  | 23.49461       | 1.123477e-15  | 25.847274      |
| 22  | -              | -              | 1.708625       |
| 23  | 257.32042      | 2.468230e-03  | 80.194234      |
| 24  | 22.803333      | 1.123477e-15  | 33.48053       |
| 25  | 128.64854      | 1.265847e-15  | 120.333564     |
| 26  | 45.951071      | 1.904278e-17  | 20.745042      |
| 27  | -              | -              | 4.49182        |
| 28  | 16.206452      | -5.000000e+00 | 14.705810      |
| 29  | 106.474900     | 7.885800e+04  | 63.472192      |
| 30  | 195.604987     | 1.377615e-07  | -              |
| 31  | 72.153478      | 1.387487e-06  | 59.246356      |
| 32  | 627.723800     | 1.154419e-08  | 481.009080     |
| 33  | 815.975027     | 1.159799e+00  | 412.175439     |
| 34  | -              | -              | 123.673018     |
| 35  | 109.413710     | 1.168798e+03  | 59.507554      |
| 36  | 278.038612     | -8.168486e-01 | 206.770321     |
| 37  | 163.553000     | -4.804530e-01 | 5.985675       |
| 38  | 2767.400749    | 1.347062e-20  | 763.733536     |
| 39  | 327.567654     | 1.593518e-08  | 151.893298     |
| 40  | 3077.334970    | 2.394372e-14  | 575.187327     |

https://doi.org/10.1371/journal.pone.0253016.t002
Fig 1. Performance profiles on the basis of running time (in seconds).
https://doi.org/10.1371/journal.pone.0253016.g001

Fig 2. Performance profiles on the basis of the number of function evaluations.
https://doi.org/10.1371/journal.pone.0253016.g002
Fig 3. Performance profiles on the basis of the number of gradient evaluations.
https://doi.org/10.1371/journal.pone.0253016.g003

Fig 4. Performance profiles of these algorithms on the basis of running time (in seconds).
https://doi.org/10.1371/journal.pone.0253016.g004
Fig 5. Performance profiles of these algorithms on the basis of the number of function evaluations.
https://doi.org/10.1371/journal.pone.0253016.g005

Fig 6. Performance profiles of these algorithms on the basis of the number of gradient evaluations.
https://doi.org/10.1371/journal.pone.0253016.g006
Table 3. Numerical comparisons on a subset of test problems.

| No. | n   | ANTRL33 | NFTR   | WA FTR57 | NAFRTR-2 84 |
|-----|-----|---------|--------|----------|-------------|
| 1   | 200 | -       | 1190/626 | 129/70  | 1260/631  |
| 2   | 100 | 1984/1000 | 3203/1609 | 1069/543 | 875/438   |
| 3   | 200 | 32/17  | 29/25  | 48/44   | 37/19   |
| 4   | 36  | -      | 116/81  | 107/78  | 87/44   |
| 5   | 1000 | 85/43 | 2057/1039 | 121/66  | 113/57  |
| 6   | 200 | 1054/578 | 507/254  | 501/251  | 539/270  |
| 7   | 200 | 478/252 | 17/9   | 15/8   | 15/7   |
| 8   | 500 | 12/7   | 260/132 | 256/131 | 393/197  |
| 9   | 500 | 461/238 | 153/77  | 55/28   | 53/27   |
| 10  | 500 | 51/27  | 41/24  | 36/22  | 35/18  |
| 11  | 500 | 35/19  | 15/8   | 15/8   | 15/8   |
| 12  | 500 | 17/9   | 6/5    | 10/9   | 11/6   |
| 13  | 1000 | 28/16 | 69/55  | 70/56   | 29/15   |
| 14  | 200 | -      | 1138/588 | 978/540 | 806/420  |
| 15  | 1000 | 14/8  | 9/8    | 9/8    | 9/8    |
| 16  | 1000 | 27/15 | 27/17  | 49/28  | 183/92  |
| 17  | 500 | 2127/1072 | 207/140 | 275/172 | 261/131  |
| 18  | 500 | 1356/694 | -      | 2567/1481 | 1190/708  |
| 19  | 500 | 46/24  | 34/25  | 169/96  | 43/22   |
| 20  | 1000 | 67/34 | 107/86  | 258/139  | 89/45   |
| 21  | 1000 | 40/22 | 53/44   | 27/25  | 35/18   |
| 22  | 1000 | 51/27 | 74/63  | 34/34   | 34/34   |
| 23  | 1000 | 539/273 | 498/310 | 427/289 | 384/202  |
| 24  | 192  | 3389/1706 | 430/228 | 634/333 | 717/360  |
| 25  | 500 | 621/329 | 1048/553 | 704/373  | 559/280  |
| 26  | 500 | 30/17  | 110/73  | 112/75  | 35/18   |
| 27  | 1000 | 79/43 | 85/57   | 68/38  | 61/31   |
| 28  | 1000 | 16/9  | 8/8    | 8/8    | 8/8    |
| 29  | 1000 | 24/13 | 27/14  | 46/29  | 23/12   |
| 30  | 1000 | -     | 3179/1649 | 5040/2632 | 3403/1702  |
| 31  | 1000 | 47/25 | 26/25  | 26/25  | 47/24   |
| 32  | 3000 | 47/25 | 48/29  | 26/24  | 47/24   |
| 33  | 3000 | 42/23 | 29/20  | 28/18  | 35/18   |
| 34  | 200  | 343/172 | 750/716  | 1033/782 | 686/641  |
| 35  | 3000 | 9/5   | 11/6   | 11/6   | 9/5    |
| 36  | 3000 | 17/9  | 14/9   | 15/10  | 16/10   |
| 37  | 1000 | 13/7  | 10/7   | 10/7   | 7/6    |
| 38  | 1000 | 598/368 | 327/254 | 439/261 | 571/286  |
| 39  | 3000 | 107/68 | 95/48   | 44/23  | 43/22   |
| 40  | 3000 | 386/249 | 359/295 | -      | 294/148  |

https://doi.org/10.1371/journal.pone.0253016.t003

5. Conclusions

In this paper, making proper use of the filter technique, a new trust region method has been proposed that the trust region radius takes into account the gradient information of the function and the solution of the trust region subproblem. To some extent, it is more reasonable to adopt the convex combination of nonmonotone trust region ratio and retrospective ratio. In addition, the approximation of the Hessian matrix is updated by an improved quasi-Newton
formula. From the theoretical view, the new algorithm keeps global convergent and super-linear convergent. Numerical experiments show the availability of the new proposed algorithm in accordance with the Dolan-Moré performance profile.

Acknowledgments
At the point of finishing this paper, I’d like to express my sincere thanks to all those who have lent me hands in the course of my writing this paper. Xinyi Wang proposes the new trust region algorithm for unconstrained optimization. Quan Qu proves the global convergence and superlinear convergence of the algorithm. Xianfeng Ding conceives of the study and participated in its design and coordination. All authors read and approved the final manuscript.

Author Contributions
Conceptualization: Xinyi Wang.
Data curation: Xianfeng Ding.
Resources: Xinyi Wang.
Software: Quan Qu.
Supervision: Xinyi Wang.
Writing – original draft: Quan Qu.
Writing – review & editing: Xianfeng Ding, Xinyi Wang.

References
1. Nocedal J, Yuan YX. Combining trust region and line search techniques. In: Yuan Y.X. (ed.) Advances in Nonlinear Programming. 1998;153–175.
2. Cui Z, Wu B. A new modified nonmonotone adaptive trust region method for unconstrained optimization. Comput. Optim. Appl.2012; 53(3);795–806.
3. Zhang, Wang Y. A new trust region method for nonlinear equations. Math. Meth. Oper. Res.2003; 58;283–298.
4. Wachter A, Biegler LT. Line search filter methods for nonlinear programming and global convergence. SIAM J. Optim.2005; 16;1–31. https://doi.org/10.1137/S1052623403426556
5. Chamberlain RM, Powell MJD, Lemarechal C, Pedersen HC. The watchdog technique for forcing convergence in algorithm for constrained optimization. Math. Program.Stud.1982; 16;1–17.
6. Ahookhosh M, Amini K, Peyghami M. A nonmonotone trust region line search method for large scale unconstrained optimization. Appl. Math. Model.2012; 36(1);478–487. https://doi.org/10.1016/j.apm.2011.07.021
7. Zhang HC, Hager WW. A nonmonotone line search technique and its application to unconstrained optimization. SIAM Journal on Optimization.2004; 14(4);1043–1056.
8. Gu NZ, Mo JT. Incorporating nonmonotone strategies into the trust region for unconstrained optimization. Computers and Mathematics with Applications. Computers and Mathematics with Applications.2008; 55(9);2158–2172.
9. Deng NY, Xiao Y, Zhou FJ. Nonmonotonic trust region algorithm. J. Optim. Theory Appl.1993; 76;259–285.
10. Grippo L, Lamparillo F, Lucidi S. A nonmonotone line search technique for Newton’s method. Siam J. Numer. Anal.1986; 23;707–716.
11. Zhou S, Yuan GL, Cui ZR. A new adaptive trust region algorithm for optimization problems. Acta Mathematica Scientia.2018; 38B(2);479–496.
12. Kimiae M. A new class of nonmonotone adaptive trust-region methods for nonlinear equations with box constraints. Calcolo.2017; 54;769–812.
13. Amini K, Shiker Mushtak AK, Kimiae M. A line search trust-region algorithm with nonmonotone adaptive radius for a system of nonlinear equations. Q. J. Oper. Res.2016; 4;132–152.
14. Zhang XS, Zhang JL, Liao LZ. An adaptive trust region method and its Convergence. Sci. China. 2002; 45; 620–631.
15. Qu Q, Ding XF, Wang XY. A Filter and Nonmonotone Adaptive Trust Region Line Search Method for Unconstrained Optimization. Symmetry-Basel. 2020; 12; 656.
16. Fletcher R, Leyffer S. Nonlinear programming without a penalty function. Math. Program. 2002; 91; 239–269.
17. Gould NI, Sainvitu C, Toint PL. A filter-trust-region method for unconstrained optimization. SIAM J. Optim. 2005; 16; 341–357.
18. Conn AR, Gould NIM, Toint PL. Trust-Region Methods. MPS/SIAM Series on Optimization, SIAM, Philadelphia, 2000.
19. Zhou Sheng, Gonglin Yuan. A NEW ADAPTIVE TRUST REGION ALGORITHM FOR OPTIMIZATION PROBLEMS. Acta Mathematica Scientia. 2018; 38(2); 479–496.
20. Andrei N. An unconstrained optimization test functions collection. Environ. Sci. Technol. 2008; 28; 6552–6558.
21. D Li M Fukushima. A modified BFGS method and its global convergence in nonconvex minimization. J Comp. Appl. Math. 2001; 129; 15–35.
22. Wei Z, Li G, Qi L. New quasi-Newton methods for unconstrained optimization problems. Appl. Math. Comput. 2006; 175(2); 1156–1188.
23. Yuan G, Wei Z. Convergence analysis of a modified BFGS method on convex minimizations. Comput. Optim. Appl. 2010; 47(2); 237–255.
24. Dolan E.D, Moré J.J. Benchmarking optimization software with performance profiles. Math. Program. 2002; 91; 201–213.