Comput Optim Appl (2015) 61:321–341
DOI 10.1007/s10589-015-9726-8

A relaxed nonmonotone adaptive trust region method for solving unconstrained optimization problems

M. Reza Peyghami · D. Ataee Tarzanagh

Received: 11 March 2013 / Published online: 30 January 2015

Abstract In this paper, we present a new relaxed nonmonotone trust region method with adaptive radius for solving unconstrained optimization problems. The proposed method combines a relaxed nonmonotone technique with a modified version of the adaptive trust region strategy proposed by Shi and Guo (J Comput Appl Math 213:509–520, 2008). Under some suitable and standard assumptions, we establish the global convergence property as well as the superlinear convergence rate for the new method. Numerical results on some test problems show the efficiency and effectiveness of the new proposed method in practice.

Keywords Trust region methods · Nonmonotone techniques · Adaptive trust region methods

1 Introduction

Consider the following unconstrained optimization problem:

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

Electronic supplementary material The online version of this article (doi:10.1007/s10589-015-9726-8) contains supplementary material, which is available to authorized users.

M. Reza Peyghami (✉)
Faculty of Mathematics, K.N. Toosi University of Technology, P.O. Box 16315-1618 Tehran, Iran
e-mail: peyghami@kntu.ac.ir

M. Reza Peyghami · D. Ataee Tarzanagh
Scientific Computations in Optimization and Systems Engineering (SCOPE), K.N. Toosi University of Technology, Tehran, Iran
e-mail: Tarzanagh@gmail.com

in which $f : R^n \to R$ is a twice continuously differentiable function. Trust region and line search methods are two popular approaches in the literature for solving these kinds of optimization problems. Line search methods refer to a procedure that computes a steplength α_k in the specific direction d_k at current point x_k and generates a new point as $x_{k+1} = x_k + \alpha_k d_k$. It is well known that in these methods, one may require the Hessian approximation matrix to be positive definite for ensuring that the search direction is a descent direction, see e.g. [17,18].

Trust region (TR) methods are another class of iterative methods for solving unconstrained optimization problems [5]. These methods generate a sequence of points that converges to a point in which the first and second-order necessary conditions hold under some mild assumptions. Moreover, they can be applied to ill-conditioned problems, have strong global convergence properties and do not require the Hessian approximation to be positive definite. Due to their strong convergence property and robustness, they have been intensively studied in the literature [5,17]. In the classical TR methods, at the current point x_k, a trial step d_k is computed by approximately solving the following TR subproblem:

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

where $\|.\|$ can be an arbitrary vector norm, usually the Euclidean norm, $g_k = \nabla f(x_k)$, B_k is the exact Hessian, i.e. $\nabla^2 f(x_k)$, or its symmetric approximation and Δ_k is the TR radius. Then, the agreement between the actual and the predicted reductions is computed by the so called TR ratio r_k, which is defined by:

$$r_k = \frac{Ared_k}{Pred_k},$$

where the actual reduction $Ared_k$ and the predicted reduction $Pred_k$ are given by:

$$Ared_k := f(x_k) - f(x_k + d_k), \tag{3}$$

$$Pred_k := m_k(0) - m_k(d_k). \tag{4}$$

Based on the magnitude of r_k, the classical TR methods decide whether the trial step is accepted or rejected. More precisely, for a given $\mu \in (0, 1)$, if $r_k \ge \mu$, then the trial step is accepted and the new point is introduced by $x_{k+1} = x_k + d_k$. In this case, the TR radius is updated appropriately. On the other hand, if the actual reduction is poor compared with the predicted reduction, i.e. $r_k < \mu$, then the trial step is rejected and the current point remains unchanged for the next iteration. In this case, the TR radius is shrunk in an appropriate manner. This process is repeated until the convergence criteria hold, see e.g. [5,12,18,20].

Appropriately choosing the initial radius and the way of updating TR radii are crucial points in the performance of standard TR methods, see e.g. [1,15,22–24,30,31], and may cause a meaningful decrease in the number of subproblem solving. Sartenaer [22] proposed a strategy for automatically determining an initial radius by letting it to be $\| g_0 \|$. Later, in practical point of view, Lin and Moré [15] introduced a better choice for the initial radius by letting $\Delta_0 = \kappa \parallel g_0 \parallel$, where κ is a positive constant. It is worth mentioning that when the sequence $\{x_k\}$, generated by the classical TR algorithm, converges to a minimizer x^*, the ratio $\{r_k\}$ may converge to 1. Thus, for sufficiently large k, the TR radius might be larger than a positive constant. In this case, the trust region constraint doesn't play any role at the end. Due to this fact, Fan and Yuan [10] proposed a TR method with the radius Δ_k converging to zero. In their approach, the radius is introduced by $\Delta_k = \upsilon_k \parallel g_k \parallel$, where υ_k is updated according to the magnitude of the ratio r_k. Zhang et al. [30] suggested another scheme for adaptively determining the radius. They employed the adaptive formula $\Delta_k = c^p \parallel g_k \parallel \parallel \hat{B}_k^{-1} \parallel$ in their scheme, where $c \in (0, 1)$ is a constant, p is a nonnegative integer and $\hat{B}_k = B_k + iI$ is a positive definite matrix, for some $i \in N$. Recently, some variants of adaptive trust region methods based on the following updating formula have been proposed in [6,21]:

$$\Delta_k = \upsilon_k \parallel g_k \parallel \parallel \hat{B}_k^{-1} \parallel, \tag{5}$$

where υ_k is updated according to the magnitude of r_k.

Despite using the current gradient and Hessian information in Zhang's method, computing Δ_k in (5) requires an estimation of $\parallel \hat{B}_k^{-1} \parallel$ in each iteration, which causes some extra computational costs. In order to overcome this drawback, a simple adaptive trust region method was proposed by Shi and Wang [24], in which the radius is updated by $\Delta_k = c^p \parallel g_k \parallel^3 / g_k^T \hat{B}_k g_k$, where $c \in (0, 1)$ is a constant, $\hat{B}_k$ is a positive definite matrix and p is a nonnegative integer. A practically efficient and globally convergent adaptive TR method has been suggested by Shi and Guo in [23]. In their method, the radius is updated by $\Delta_k = \alpha_k \parallel q_k \parallel$, where q_k is chosen so that, for given $\tau \in (0, 1]$,

$$-\frac{g_k^T q_k}{\parallel g_k \parallel . \parallel q_k \parallel} \geq \tau. \tag{6}$$

Moreover, for given $\rho \in (0, 1)$, α_k is the largest possible $\alpha \in \{s_k, \rho s_k, \rho^2 s_k, \ldots\}$, so that $r_k \geq \mu$, where s_k is determined by

$$s_k = -\frac{g_k^T q_k}{q_k^T \hat{B}_k q_k}, \tag{7}$$

and $\hat{B}_k$ is defined by

$$\hat{B}_k = B_k + iI, \tag{8}$$

in which I is the identity matrix and i is the smallest nonnegative integer so that $\hat{B}_k$ is a positive definite matrix.

In the monotone trust region methods, the value of objective function is required to be decreased in each iteration. It has been shown that this procedure may result in slow convergence rate in some problems [13,14]. One way to overcome this difficulty is to use nonmonotone techniques in the TR frameworks. Apparently, the first nonmonotone line search technique is the so called watchdog technique, which was proposed by

Chamberlain et al. [4]. Later, Grippo et al. [13] provided a nonmonotone line search technique for Newton's method and extended it for solving unconstrained optimization problems [14]. In their approach, for given $\gamma \in (0,\ 1)$, the steplength α_k is chosen so that the following condition holds:

$$f\ (x_k\ +\ \alpha_k d_k) \leq f_{l(k)} + \gamma \alpha_k \nabla f(x_k)^{\mathrm{T}} d_k,$$

where the *nonmonotone term* $f_{l(k)}$ is defined by

$$f_{l(k)} = \max_{0 \leq j \leq m(k)} \left\{ f\ \left(x_{k-j}\right) \right\}, \tag{9}$$

in which, for given nonnegative integer M, $m\ (0) = 0$ and $0 \leq m\ (k) \leq \min\ \{m(k-1) + 1,\ M\}$, for all $k \geq 1$.

Nowadays, due to the behavior of nonmonotone techniques in practice, several authors have been fascinated to employ nonmonotone strategies in the optimization methods, especially in TR methods. The first variant of this kind goes back to the work of Deng et al. [8] in which the ratio r_k is changed according to the nonmonotone strategy as proposed in [13]. Later on, several works have been developed based on various nonmonotone techniques. Among them are the works proposed by Zhou and Xiao [28,32], Xiao and Chu [27], Toint [25,26], Dai [7] and Panier and Tits [19]. However, the Grippo's nonmonotone technique has some disadvantages, see e.g. [2,29]. In order to overcome these drawbacks, Zhang and Hager [29] suggested another nonmonotone line search technique in which the maximum over the function values in (9) is replaced by an average of the function values. More precisely, in their approach, for given $\gamma \in (0,\ 1)$, the steplength α_k is chosen so that

$$f\ (x_k\ +\ \alpha_k d_k) \leq C_k + \gamma \alpha_k \nabla f(x_k)^{\mathrm{T}} d_k,$$

where the *nonmonotone term* C_k is defined by

$$C_k = \begin{cases} f(x_k) & \text{if} \quad k = 0, \\ \frac{(\vartheta_{k-1} Q_{k-1} C_{k-1} + f(x_k))}{Q_k} & \text{if} \quad k \geq 0, \end{cases} \tag{10}$$

and

$$Q_k = \begin{cases} 1 & \text{if} \quad k = 0, \\ \vartheta_{k-1} Q_{k-1} + 1 & \text{if} \quad k \geq 0, \end{cases}$$

where $\vartheta_{k-1} \in [\vartheta_{\min}, \vartheta_{\max}]$ and $0 \leq \vartheta_{\min} < \vartheta_{\max} \leq 1$.

Recently, the nonmonotone techniques and adaptive strategies are simultaneously employed in the framework of classical TR methods in order to propose an efficient algorithm in terms of convergence property and practical performance. The first work in this area goes back to 2003 in which Zhang et al. [31] combined the Grippo's nonmonotone technique with adaptive trust region method. Fu and Sun [11] used Zhang's adaptive method with a new structured nonmonotone technique in which the predicted reduction is computed in a different way than the classical TR methods. In a recent work, Cui and Wu [6] provided a nonmonotone adaptive approach based on a combination of the nonmonotone term (10) with the adaptive strategy as provided in [30]. A combination of a variant of Shi and Guo's adaptive scheme with Grippo's nonmonotone technique has been done by Ahookhosh and Amini [1]. They showed that their method is practically efficient while it has global convergence property under some standard assumptions. As the Grippo's nonmonotone technique suffers from some drawbacks [2,29], Ahookhosh and Amini in [2] proposed a new nonmonotone technique as below

$$R_k = \varepsilon_k f_{l(k)} + (1 - \varepsilon_k)\, f(x_k), \tag{11}$$

where $\varepsilon_k \in [\varepsilon_{\min}, \varepsilon_{\max}]$, $\varepsilon_{\min} \in [0, 1)$, $\varepsilon_{\max} \in (\varepsilon_{\min}, 1]$ and $f_{l(k)}$ is defined by (9). This nonmonotone term is motivated from the fact that the best convergence results are obtained by stronger nonmonotone strategy whenever the iterates are far from the optimum and by weaker nonmonotone strategy whenever the iterates are close enough to the optimum, see e.g. [29]. By this definition, a stronger nonmonotone strategy is obtained whenever ε_k is close to 1 and a weaker strategy is followed whenever ε_k is close to 0. Although, the proposed algorithm in [2] has some appealing properties, especially in practical performance, it roughly uses $R_k = f(x_k)$ in the first iterations until $f_{l(k)}$ could be able to play an active role in (11). Therefore, in the first iterations, this fact may limit the performance of the proposed algorithm in [2] as the bottom of steep and curved valleys in some problems may happen in the first iterations. In order to overcome this difficulty, in this paper, we propose a new relaxed nonmonotone technique by using (11).

In this paper, we propose a new adaptive TR algorithm which incorporates the Shi and Guo's adaptive trust region method with a variant of nonmonotone technique as provided in [2]. Our approach aims to relax the acceptance of the trial step d_k by allowing the new nonmonotone term to be larger than R_k, especially in the first iterations, while keeping the properties of R_k to be held. We establish the global convergence property as well as the local superlinear convergence rate of the new algorithm under some suitable and standard assumptions. The proposed method is implemented in MATLAB environment and applied on some test problems. Numerical results confirm that the proposed algorithm is practically efficient, too.

The rest of the paper is organized as follows: In Sect. 2, we describe the structure of our new proposed nonmonotone adaptive trust region algorithm. Section 3 is devoted to establish the local and global convergence properties of the algorithm under some standard assumptions. The superlinear convergence rate is provided in Sect. 4. Numerical results of performing the new algorithm on some test problems, taken from the literature, are given in Sect. 5. We end up the paper by some concluding remarks in Sect. 6.

2 The new algorithm

In this section, we propose a new adaptive nonmonotone trust region algorithm based on a combination of the nonmonotone technique proposed in [2] and the Shi and

Guo's adaptive scheme provided in [23]. In our algorithm, the new nonmonotone term is defined by $(1+\varphi_k)\,R_k$, where R_k is given by (11) and φ_k is determined by

$$\varphi_k = \begin{cases} \eta_k & \text{if } R_k > 0, \\ 0 & \text{if } R_k \le 0, \end{cases} \tag{12}$$

in which $\{\eta_k\}$ is a positive sequence satisfying the following condition:

$$\sum_{k=0}^{\infty} \eta_k \le \eta < \infty. \tag{13}$$

It is worth mentioning that, as $k \to \infty$, we have $\eta_k \to 0$, and therefore the proposed nonmonotone term converges to that proposed in [2]. Despite the nonmonotone term (11), the properties of strong nonmonotone strategy are taken into account in the proposed relaxed term, especially in the first iterations.

The trust region radius is updated by $\Delta_k = \min\{v_k s_k \parallel q_k \parallel, \Delta_{\max}\}$, where q_k and s_k are the parameters of the Shi and Guo's adaptive scheme, defined by (6) and (7), respectively, and v_k is appropriately adjusted in each iteration according to the magnitude of the following nonmonotone ratio:

$$r_k = \frac{(1+\varphi_k)\,R_k - f\,(x_k + d_k)}{Pred_k}. \tag{14}$$

The whole procedure of the new nonmonotone adaptive TR algorithm for solving (1) is outlined in Algorithm 2.1.

Algorithm 2.1: A New Nonmonotone Adaptive Trust Region Algorithm

Step 0. Given $x_0 \in R^n$, $0 < \mu_2 < \mu_1 < 1$, $0 < \sigma_0 < 1 < \sigma_1$, $\epsilon > 0$, $0 < \varepsilon_{\min} < \varepsilon_{\max} < 1$ $v_0 > 0$, $\Delta_{\max}$, an initial symmetric matrix $B_0 \in R^{n\times n}$, a positive sequence $\{\eta_k\}$ satisfying (13) and a positive integer M, set $v_0 = 1$ and $k = 0$.

Step 1. If $\parallel g_k \parallel < \epsilon$, then Stop.

Step 2. Choose q_k so that (6) holds and compute s_k from (7). Solve

$$\begin{aligned} &\min g_k^T d + \tfrac{1}{2} d^T B_k d \\ &s.t. \;\parallel d \parallel \le \Delta_k = \min\{v_k s_k \parallel q_k \parallel, \Delta_{\max}\}, \end{aligned} \tag{15}$$

and determine a trial step d_k.

Step 3. Set

$$x_{k+1} = \begin{cases} x_k + d_k & r_k \ge \mu_2, \\ x_k & r_k < \mu_2. \end{cases}$$

Step 4. Update v_k as below:

$$v_{k+1} := \begin{cases} \min\{\sigma_1 v_k, v_{\max}\} & r_k \ge \mu_1, \\ v_k & \mu_2 \le r_k < \mu_1, \\ \sigma_0\, v_k & \text{Otherwise.} \end{cases} \tag{16}$$

Step 5. Update B_k by a symmetric quasi-Newton formula. Set $k := k + 1$ and go to Step 1.

Throughout the paper, we use the following two index sets in our analysis:

$$I = \{k : r_k \geq \mu_2\}, J = \{k : r_k < \mu_2\}.$$

We refer to the k-th iteration as a successful iteration whenever $x_{k+1} = x_k + d_k$, i.e. $k \in I$, and as an unsuccessful iteration whenever $x_{k+1} = x_k$, i.e. $k \in J$.

Remark 2.1 Let the index set I be denoted by $\{k_0, k_1, k_2, \ldots\}$. For a given nonnegative integer M, set $m(0) = 0$ and $0 \leq m(k_i) \leq \min\{m(k_{i-1}) + 1, M\}$. At the kth iteration, let $k_i \in I$ be the largest index so that $k_i \leq k$. In our setting, we define $f_{l(k)} = \max_{0 \leq j \leq m(k_i)} f_{k_{i-j}}$. Indeed, $f_{l(k)}$ is considered as the maximum value of $f(x)$ over the last $m(k_i) + 1$ successful iteration points.

3 Convergence analysis

In this section, our aim is to establish convergence properties of Algorithm 2.1 under some suitable assumptions. To do so, the following assumptions are considered on the problem:

- A1 The level set $\Omega = \{x \in R^n | f(x) \leq e^{\eta} |f(x_0)|\}$ is a closed and bounded set, where η is defined by (13) and f is a twice continuously differentiable function over Ω.
- A2 Matrices B_k are uniformly bounded, i.e., there exists a positive constant m_1 so that, for all $k \in N \cup \{0\}$, we have $\| B_k \| \leq m_1$.

Remark 3.1 Assumption A1 implies that there exists a positive constant m_2, so that $\| \nabla^2 f(x_k) \| \leq m_2$, for all $x_k \in \Omega$.

Remark 3.2 Assumption A2 implies that the matrices $\hat{B}_k$ are also uniformly bounded. This can be proved by considering the procedure of generating $\hat{B}_k$ in Shi and Guo's approach. Indeed, Assumption A2 implies that $\| \hat{B}_k \| = \| B_k + iI \| \leq 2m_1 + 1$, where the inequality is obtained from the fact that (8) holds for $m_1 < i \leq m_1 + 1$.

Remark 3.3 In order to analyze the convergence property of Algorithm 2.1, we assume that the trial step d_k is approximately computed by Algorithm 2.6 in [18]. Therefore, as it has been shown in [18], there exists a constant $\theta \in (0, 1)$ so that d_k satisfies the following inequality:

$$Pred_k \geq \theta \| g_k \| \min\left\{\Delta_k, \frac{\| g_k \|}{\| B_k \|}\right\}. \tag{17}$$

This inequality is known as a sufficient reduction condition in the literature and implies that $d_k \neq 0$ whenever $g_k \neq 0$.

Note that, throughout this section, we use the definition of $f_{l(k)}$ as given in Remark 2.1.

Lemma 3.1 *For all k, we have*

$$|f(x_k) - f(x_k + d_k) - Pred_k| \leq O\left(\| d_k \|^2\right), \tag{18}$$

where $Pred_k$ is defined by (4).

Proof The proof is easily obtained by using Taylor's expansion, Assumptions A2 and Remark 3.1. □

Lemma 3.2 *Let Assumptions A1 and A2 hold and the sequence* $\{x_k\}$ *be generated by Algorithm 2.1. Moreover, suppose that there exists a constant* $\delta \in (0, 1)$*, so that* $\| g_k \| > \delta$*, for all k. Then, for each k, there exists a nonnegative integer p, so that* x_{k+p+1} *is a successful iteration point.*

Proof By contrary, suppose that there exists an iteration k so that, for all nonnegative integer p, x_{k+p+1} is an unsuccessful iteration point, i.e.

$$r_{k+p} < \mu_2, \quad p = 0, 1, 2, \ldots \tag{19}$$

Using (8), we have $q_{k+p}^T \hat{B}_{k+p} q_{k+p} > 0$. Therefore, there exists a sufficiently small positive constant ϱ so that

$$0 < \varrho \| q_{k+p} \|^2 \leq q_{k+p}^T \hat{B}_{k+p} q_{k+p}.$$

Thus, from Step 4 of Algorithm 2.1 and (7), we have

$$\begin{aligned} \Delta_{k+p+1} &\leq \sigma_0^{p+1} \upsilon_k \, s_{k+p} \| q_{k+p} \| \\ &= -\sigma_0^{p+1} \upsilon_k \frac{g_{k+p}^T q_{k+p}}{q_{k+p}^T \hat{B}_{k+p} q_{k+p}} \| q_{k+p} \| \\ &\leq \sigma_0^{p+1} \upsilon_k \frac{\| g_{k+p} \| \| q_{k+p} \|}{\varrho \| q_{k+p} \|^2} \| q_{k+p} \| \\ &\leq \frac{\upsilon_{\max} \| g_k \|}{\varrho} \sigma_0^{p+1}, \end{aligned}$$

where the second inequality is followed by applying the Cauchy–Schwartz inequality and the last inequality is obtained from (16) and the fact that in the unsuccessful iterations, we have $x_{k+p} = x_k$. Now, since $\sigma_0 \in (0, 1)$, the latter inequality implies that

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

Therefore, from Lemma 3.1 and (17), we have

$$\begin{aligned} \left| \frac{f(x_{k+p}) - f(x_{k+p} + d_{k+p})}{Pred_{k+p}} - 1 \right| &= \left| \frac{f(x_{k+p}) - f(x_{k+p} + d_{k+p}) - Pred_{k+p}}{Pred_{k+p}} \right| \\ &\leq \frac{O(\| d_{k+p} \|^2)}{\theta \| g_{k+p} \| \min\left\{ \Delta_{k+p}, \frac{\| g_{k+p} \|}{\| B_{k+p} \|} \right\}} \\ &\leq \frac{O\left(\| \Delta_{k+p} \|^2\right)}{\theta \delta \min\left\{ \Delta_{k+p}, \frac{\delta}{m_1} \right\}}. \end{aligned}$$

This implies that $\left|\frac{f(x_{k+p})-f(x_{k+p}+d_{k+p})}{Pred_{k+p}}-1\right| \to 0$, as $p \to \infty$. Thus, for sufficiently large p, we have

$$r_{k+p} = \frac{\left(1+\varphi_{k+p}\right) R_{k+p} - f\left(x_{k+p}+d_{k+p}\right)}{Pred_{k+p}} \geq \frac{f\left(x_{k+p}\right) - f\left(x_{k+p}+d_{k+p}\right)}{Pred_{k+p}} \to 1,$$

which contradicts (19). This completes the proof of the lemma. □

Remark 3.4 As a consequence of Lemma 3.2, one can realize that whenever Algorithm 2.1 does not stop in the finite number of iterations, i.e. the sequence $\{x_k\}$ is an infinite sequence, then the index set I is infinite.

Lemma 3.3 *Suppose that $\{x_k\}$ is generated by Algorithm 2.1. Then, we have*

$$f_{k+1} \leq |f_0| \prod_{i=0}^{k} (1+\varphi_i) - \omega_k, \quad k \in I, \tag{20}$$

where $\omega_k := \theta\mu_2 \min\left\{\| g_k \| \, \Delta_k, \frac{\| g_k\|^2}{\|B_k\|}\right\} \geq 0$, $\mu_2 \in (0, 1)$ and $\theta \in (0, 1)$ is the same constant as in Remark 3.3.

Proof Let $k \in I$, then $x_k + d_k$ is a successful iteration point. From (14) and (17), for all $k \in I$, we have

$$(1+\varphi_k) R_k - f(x_{k+1}) \geq \mu_2 Pred_k \geq \theta\mu_2 \min\left\{\| g_k \| \, \Delta_k, \frac{\| g_k \|^2}{\| B_k \|}\right\} = \omega_k \geq 0. \tag{21}$$

We proceed the proof by induction. To establish the first step of the induction, the following two cases are considered:

Case 1 Let $k = 0$ be a successful iteration. In this case, using (21), we obtain

$$f_1 \leq (1+\varphi_0) R_0 - \omega_0 = (1+\varphi_0) f_0 - \omega_0 \leq (1+\varphi_0) |f_0| - \omega_0,$$

where the equality is followed from (11) and the fact that $\varphi_0 \geq 0$, by (12). Assume that (20) holds for $k \geq 1$, i.e.,

$$f_{k+1} \leq |f_0| \prod_{i=0}^{k} (1+\varphi_i) - \omega_k \leq |f_0| \prod_{i=0}^{k} (1+\varphi_i). \tag{22}$$

Due to Lemma 3.2, there exists the smallest positive integer p, so that $k+p$ is a successful iteration. We show that (20) holds for $k + p \in I$, i.e.,

$$f_{k+p+1} \leq |f_0| \prod_{i=0}^{k+p} (1+\varphi_i) - \omega_{k+p}.$$

For this purpose, from (12) and (21), we have

$$\begin{aligned} f_{k+p+1} &\leq \left(1+\varphi_{k+p}\right) R_{k+p} - \omega_{k+p} \\ &= \left(1+\varphi_{k+p}\right) R_{k+1} - \omega_{k+p} \\ &= \left(1+\varphi_{k+p}\right) \left[\varepsilon_k f_{l(k+1)} + (1-\varepsilon_k)\, f_{k+1}\right] - \omega_{k+p} \\ &\leq \left(1+\varphi_{k+p}\right) \left[\varepsilon_k f_{l(k+1)} + (1-\varepsilon_k)\, f_{l(k+1)}\right] - \omega_{k+p} \\ &\leq \left(1+\varphi_{k+p}\right) f_{l(k+1)} - \omega_{k+p} \\ &\leq \left(1+\varphi_{k+p}\right) |f_0| \prod_{i=0}^{l(k+1)-1} (1+\varphi_i) - \omega_{k+p}, \end{aligned}$$

where the first equality is obtained from the fact that iterations $k+1; \ldots ; k+p-1$ are unsuccessful iterations, and therefore $R_{k+1} = \cdots = R_{k+p}$, and the last inequality is followed from Remark 2.1 and the induction's hypothesis by considering the fact that $k+1$ is an unsuccessful iteration and $l\,(k+1) \leq k+1$. Now, as $l\,(k+1) \leq k+1 \leq k+p$, we conclude that

$$f_{k+p+1} \leq \left(1+\varphi_{k+p}\right) |f_0| \prod_{i=0}^{k+p-1} (1+\varphi_i) - \omega_{k+p} \leq |f_0| \prod_{i=0}^{k+p} (1+\varphi_i) - \omega_{k+p}.$$

Case 2 Let $k = 0 \in J$. Due to Lemma 3.2, there exists the smallest positive integer p, so that p is a successful iteration. In this case, we start the first step of the induction by $k = p$ (strong induction). The rest of the proof is similar to *Case 1*. □

Corollary 3.1 *For all k, one has:*

$$f_{k+1} \leq |f_0| \prod_{i=0}^{k} (1+\varphi_i)\,.$$

Proof From (20), it is easily seen that the statement is true for all $k \in I$. Now, let $k \in J$. Consider two following cases:

Case 1 Assume that there exists at least a successful iteration before iteration k. Then, from Lemma 3.3, there exists the smallest positive integer p so that $k-p \in I$. This implies that:

$$f_{k-p+1} \leq |f_0| \prod\nolimits_{i=0}^{k-p} (1+\varphi_i).$$

Moreover, we have $f_{k-p+1} = \cdots = f_k = f_{k+1}$. Therefore, using (12), we obtain:

$$f_{k+1} \leq |f_0| \prod_{i=0}^{k} (1+\varphi_i).$$

Case 2 Assume that there is no successful iteration before iteration k. In this case, we have: $f_0 = f_1 = \cdots = f_{k+1}$. Therefore, using (12), we obtain:

$$f_{k+1} = f_0 \leq |f_0| \prod_{i=0}^{k} (1 + \varphi_i) .$$

This completes the proof of the corollary. □

Lemma 3.4 *For the sequence $\{x_k\}$, generated by Algorithm 2.1, one has $\{x_k\} \subseteq \Omega$.*

Proof We proceed by induction on k. For $k = 0$, the result is trivial from Assumption **A1**. Assume that $x_k \in \Omega$ (induction hypothesis). We show that $x_{k+1} \in \Omega$. Using Corollary 3.1 and the inequality between the geometric and arithmetic means, we have

$$\begin{aligned} f_{k+1} &\leq |f_0| \prod_{i=0}^{k} (1 + \varphi_i) \\ &\leq |f_0| \left(\frac{1}{k+1} \sum_{i=0}^{k} (1 + \varphi_i) \right)^{k+1} \\ &= |f_0| \left(1 + \frac{1}{k+1} \sum_{i=0}^{k} \varphi_i \right)^{k+1} \\ &\leq |f_0| \left(1 + \frac{\eta}{k+1} \right)^{k+1} \\ &\leq e^{\eta} \, |f_0| , \end{aligned} \tag{23}$$

where the last two inequalities are followed from (12), (13) and the fact that the sequence $\left\{ \left(1 + \frac{\eta}{k+1}\right)^{k+1} \right\}$ is an increasing sequence converging to e^{η}. Therefore, $x_{k+1} \in \Omega$, and the proof is completed. □

Lemma 3.5 *Let q_k satisfy (6) and*

$$\lim_{k\to\infty} \frac{-g_k^T q_k}{\| q_k \|} = 0.$$

Then, we have

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

Proof Using (6), we have

$$0 \leq \lim_{k\to\infty} \tau \| g_k \| \leq \lim_{k\to\infty} \frac{-g_k^T q_k}{\| q_k \| \| g_k \|} \| g_k \| = \lim_{k\to\infty} \frac{-g_k^T q_k}{\| q_k \|} = 0.$$

This completes the proof of the lemma. □

Lemma 3.6 *Assume that the index set I is infinite and is denoted by $\{k_0, k_1, k_2, \ldots\}$. Then, for each successful iteration k_j, there exist a nonnegative integer L and $0 \le r \le M-1$, so that $k_j = k_{LM+r}$ and*

$$f_{k_{LM+r}} \le |f_0| \prod_{i=0}^{k_j-1} (1+\varphi_i) - \sum_{i=0}^{L} \tilde{\omega}_i, \quad L = 0, 1, 2, \ldots \tag{24}$$

where $\tilde{\omega}_i = \min_{0 \le r \le M-1} \omega_{k_{iM+r}}$.

Proof See Appendix. □

The following theorem states the global convergence property of Algorithm 2.1.

Theorem 3.1 *Suppose that assumptions A1 and A2 hold. Then, Algorithm 2.1 either stops at a stationary point of the problem (1) or generates an infinite sequence $\{x_k\}$ so that*

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

Proof Suppose that Algorithm 2.1 does not stop at a stationary point. We show that

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

By contrary, assume that there exists a positive constant δ so that, for all k,

$$\| g_k \| > \delta. \tag{25}$$

Using Lemma 3.6 and (23), we have

$$\begin{aligned}\sum_{i=0}^{L} \tilde{\omega}_i &\le |f_0| \prod_{i=0}^{k_{LM+r}-1} (1+\varphi_i) - f_{k_{LM+r}} \\ &\le e^{\eta} |f_0| - f_{k_{LM+r}}.\end{aligned}$$

Thus, as $L \to \infty$, Assumption A1 implies that

$$\lim_{i\to\infty} \tilde{\omega}_i = 0. \tag{26}$$

Due to the definition of $\tilde{\omega}_i$, there exists $0 \le \bar{r} \le M-1$, so that $\tilde{\omega}_i = \omega_{k_{iM+\bar{r}}}$. Letting $\beta_i = k_{iM+\bar{r}}$, from Lemma 3.3, we have

$$\tilde{\omega}_i = \omega_{\beta_i} = \theta \mu_2 \min \left\{ \| g_{\beta_i} \| \Delta_{\beta_i}, \frac{\| g_{\beta_i} \|^2}{\| B_{\beta_i} \|} \right\} \to 0, \text{ as } i \to \infty.$$

Using (25), (26) and Assumption A2, we conclude that

$$\Delta_{\beta_i} \to 0, \text{ as } i \to \infty. \tag{27}$$

Therefore, we have $\Delta_{\beta_i} = \min\{v_{\beta_i} s_{\beta_i} \parallel q_{\beta_i} \parallel, \Delta_{\max}\} = v_{\beta_i} s_{\beta_i} \parallel q_{\beta_i} \parallel$. On the other hand, using Cauchy–Schwartz inequality, we have

$$\begin{aligned} 0 &\le \frac{v_{\beta_i}}{2m_1+1}\left(\frac{g_{\beta_i}^T q_{\beta_i}}{\parallel q_{\beta_i} \parallel}\right)^2 \le v_{\beta_i} \frac{\left(g_{\beta_i}^T q_{\beta_i}\right)^2}{q_{\beta_i}^T \hat{B}_{\beta_i} q_{\beta_i}} = v_{\beta_i} s_{\beta_i}\left(-g_{\beta_i}^T q_{\beta_i}\right) \\ &\le v_{\beta_i} s_{\beta_i} \parallel q_{\beta_i} \parallel\parallel g_{\beta_i} \parallel = \Delta_{\beta_i} \parallel g_{\beta_i} \parallel, \end{aligned}$$

where the second inequality is obtained from Remark 3.2. Now, Lemma 3.5 and (25) imply that

$$\lim_{i\to\infty} \frac{-g_{\beta_i}^T q_{\beta_i}}{\parallel q_{\beta_i} \parallel} \ne 0.$$

Therefore, the inequality

$$0 \le \frac{v_{\beta_i}}{2m_1+1}\left(\frac{g_{\beta_i}^T q_{\beta_i}}{\parallel q_{\beta_i} \parallel}\right)^2 \le \parallel g_{\beta_i} \parallel \Delta_{\beta_i},$$

together with (25) and the fact that $\nabla f(x)$ is continuous over the compact set Ω, by Assumption A1, imply that

$$\lim_{i\to\infty} v_{\beta_i} = 0. \tag{28}$$

Now, we have:

$$\begin{aligned} \left|\frac{f\left(x_{\beta_i}\right) - f\left(x_{\beta_i} + d_{\beta_i}\right)}{Pred_{\beta_i}} - 1\right| &= \left|\frac{f\left(x_{\beta_i}\right) - f\left(x_{\beta_i} + d_{\beta_i}\right) - Pred_{\beta_i}}{Pred_{\beta_i}}\right| \\ &\le \frac{O\left(\parallel d_{\beta_i} \parallel^2\right)}{\theta\mu_2 \parallel g_{\beta_i} \parallel \min\left\{\Delta_{\beta_i}, \frac{\parallel g_{\beta_i} \parallel}{\parallel B_{\beta_i} \parallel}\right\}} \\ &= \frac{O\left(\parallel d_{\beta_i} \parallel^2\right)}{O(\Delta_{\beta_i})} \le \frac{O\left(\parallel d_{\beta_i} \parallel^2\right)}{O\left(\parallel d_{\beta_i} \parallel^2\right)} \overset{i\to\infty}{\to} 0, \end{aligned}$$

where the first inequality is obtained from Lemma 3.1 and Remark 3.3. Therefore,

$$r_{\beta_i} = \frac{\left(1+\varphi_{\beta_i}\right) R_{\beta_i} - f\left(x_{\beta_i} + d_{\beta_i}\right)}{Pred_{\beta_i}} \ge \frac{f\left(x_{\beta_i}\right) - f\left(x_{\beta_i} + d_{\beta_i}\right)}{Pred_{\beta_i}} \to 1.$$

Thus, there exists a positive constant υ^*, so that, for sufficiently large $\beta_i \in I$, we have $\upsilon_{\beta_i} \geq \upsilon^*$, which is a contradiction with (28). This completes the proof of the theorem. □

4 Convergence rate analysis

In this section, we provide the superlinear convergence rate of Algorithm 2.1. The following theorem sates the requirements for holding the superlinear convergence rate of the sequence generated by Algorithm 2.1.

Theorem 4.1 *Let Assumptions A1 and A2 hold and $d_k = -B_k^{-1} g_k$ be a solution of the subproblem (2). Moreover, suppose that $\{x_k\}$ is the sequence generated by Algorithm 2.1, which converges to a stationary point x^*. Also, assume that $\nabla^2 f\,(x^*)$ is a positive definite matrix and $\nabla^2 f\,(x)$ is a Lipschitz continuous in a neighborhood of x^*. If B_k satisfies the following condition:*

$$\lim_{k\rightarrow\infty} \frac{\| \left(B_k - \nabla^2 f(x^*)\right) d_k \|}{\| \, d_k \, \|} = 0, \tag{29}$$

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

Proof The proof is similar to the proof of Theorem 4.1 in [6]. □

5 Numerical results

In this section, we present and compare the computational results of applying two versions of Algorithm 2.1, based on two nonmonotone terms given by (10) and (11), and some other existing algorithms. For ease of reference, we call Algorithm 2.1 with the nonmonotone terms (10) and (11) as RNATR-Z and RNATR-A, respectively. All of test problems are taken from Andrei [3] and Moré et al. [16]. In the considered TR methods, q_k has a wide scope for choosing which only needs to satisfy (4). Two popular choices of q_k are $q_k = -g_k$, which is a natural choice, and $q_k = -B_k^{-1} g_k$, which has some interesting properties in theory and in practice, see e.g. [1,23]. In order to compare the efficiency of the new proposed adaptive radius, we use both above mentioned q_k's, but in numerical results in terms of number of iterations and function evaluations, we just focus on the case $q_k = -g_k$ in order to save the computational costs in large scale problems.

We have implemented algorithms RNATR-Z and RNATR-A along with the following algorithms in MATLAB 7.10.0 (R2010a) environment and run the problems on a PC with CPU 2.0 GHz and 2GB RAM memory and double precision format:

NATSG:	NMATR method proposed in [1] with $q_k = -g_k$.
NATSH:	NMATR method proposed in [1] with $q_k = -B_k^{-1} g_k$.
NATFG:	Algorithm 2.1 with $q_k = -g_k$ and the same nonmonotone term of NMATR.
NATFH:	Algorithm 2.1 with $q_k = -B_k^{-1} g_k$ and the same nonmonotone term of NMATR.

NBTR: Classical TR algorithm with the same nonmonotone term of NMATR;

NATR-Z: Algorithm 2.1 with the nonmonotone term as proposed in [29] with $\vartheta_0 = 0.85$, $\varphi_k = 0$, and $q_k = -g_k$;

NATR-A: Algorithm 2.1 with the same nonmonotone term of NMTR-N1, as proposed in [2], $\varphi_k = 0$, and $q_k = -g_k$;

The following parameters are set in the related algorithms:

$$\mu_1 = 0.9, \quad \mu_2 = 0.1, \; \sigma_0 = \frac{1}{8}, \; \sigma_1 = 6, \; \rho = 0.5, \; \upsilon_0 = 0.1, \; \upsilon_{\max} = 10^5,$$
$$\Delta_{\max} = 100.$$

For the problem of size n, we set $M = 10$ whenever $n \geq 5$, otherwise we set $M = 2n$. Moreover, we set $\eta_0 = 10^4$, and

$$\eta_k = \frac{10^3}{k^2}, \quad k = 1, 2, \ldots$$

As proposed in [15], for the NBTR algorithm, we set the initial radius to be $\Delta_0 = 0.1 \parallel g_k \parallel$ and the consequence radii are updated according to the following formula:

$$\Delta_{k+1} = \begin{cases} \min\{c_1 \parallel d_k \parallel, \; \Delta_k\} & r_k \geq \mu_1, \\ \Delta_k & \mu_2 \leq r_k < \mu_1, \\ c_0 \parallel d_k \parallel & \text{Otherwise,} \end{cases}$$

where $c_0 = 0.25$ and $c_1 = 2.5$. The matrix B_k is being updated by the BFGS formula [17] as below:

$$B_{k+1} = B_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k},$$

where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. Note that, the matrix B_k is being updated as long as the curvature condition holds, i.e. $s_k^T y_k > 0$; otherwise we set $B_{k+1} = B_k$. For proper comparison, we provide all codes in the same subroutine and solve the trust region subproblems by Steihaug–Toint procedure, see e.g. page 205 in [5].

Numerical results are given in Tables 1 and S1 (see on-line supplementary material). In these tables, n, n_i and n_f represent the problem size, the number of iterations and the number of function evaluations, respectively. For the results of Table 1, all the considered algorithms are being stopped when $\parallel g_k \parallel \leq 10^{-8}$. Moreover, for the results of Table S1 (see on-line supplementary material), all the considered algorithms are being stopped whenever $\parallel g_k \parallel \leq 10^{-6} \parallel g_0 \parallel$. We declare that an algorithm is failed whenever the number of iteration and the number of function evaluations exceed 10,000 and 20,000, respectively. It is worth mentioning that the number of iterations and gradient evaluations are the same in the considered algorithms. Moreover, we have only kept the problems in the tables for which all the considered algorithms converge to the same local solution. We have also utilized the advantages of the performance profile of Dolan and Moré [9] to compare the algorithms. Figures 1 and 2 provide the

Table 1 The effect of the new proposed adaptive radius

Problem	Dimension n	NBTR n_i/n_f	NATSH n_i/n_f	NATSG n_i/n_f	NATFH n_i/n_f	NATFG n_i/n_f
Powell badly scaled	2	293/347	258/311	713/834	229/302	241/324
Brown badly scaled	2	34/35	Failed	Failed	18/90	18/90
Beale	2	17/19	19/19	16/16	16/16	16/18
Rosenbrock	2	61/67	47/66	126/145	47/57	43/53
Freudenstein and Roth	2	29/37	13/24	19/24	16/20	18/22
Jennrich and Sampson	2	23/42	19/45	225/246	22/33	18/36
Bard	2	25/28	26/37	30/47	21/24	21/24
Helical Valley	3	49/65	46/71	351/377	51/61	36/44
Gaussian	3	8/8	6/16	6/16	8/9	8/9
Gulf Res. and Development	3	68/77	54/64	1429/1435	50/69	47/62
Box 3-dimensional	3	40/49	43/43	254/254	33/38	33/38
MEYER	3	Failed	3697/4452	4011/4596	2473/2964	3033/4060
Wood	4	91/120	107/123	1867/1882	69/101	79/115
Brown and Dennis	4	50/64	68/70	Failed	34/52	37/52
Powell singular	4	59/67	47/64	115/213	44/53	59/62
Kowalik and Osborne	4	41/49	38/42	989/1101	30/42	31/38
Penalty I	4	68/79	60/71	1326/1397	67/93	63/89
Penalty II	4	189/231	395/527	956/1054	299/549	291/511
Osborne 1	5	Failed	109/119	279/330	94/111	103/131
Brown almost linear	5	673/846	442/1292	888/1501	287/256	287/256
Broyden banded	5	43/57	38/47	36/48	30/44	30/45
Biggs EXP6	6	169/173	259/271	270/301	166/207	213/304
Extended Rosenbrock	8	86/115	70/105	501/516	65/106	69/109
Watson	9	Failed	2984/2993	3167/4103	1089/1998	1769/2024
Chebyquad	10	40/56	42/70	39/47	38/54	41/64
Trigonometric	10	32/32	28/28	28/28	28/28	28/28
Discrete boundary value	10	35/48	32/47	29/43	31/39	32/38
Discrete integral equation	10	18/18	17/19	12/13	16/17	17/18
Broyden tridiagonal	10	57/111	54/101	91/97	54/81	48/68
Linear—Full rank	10	4/4	2/3	2/3	3/4	3/4
Linear—Rank 1	10	3/10	3/21	3/21	10/16	10/16
Linear—Rank 1 with zero	10	3/10	3/19	3/19	10/16	10/16
Osborne 2	11	66/70	64/73	297/354	60/75	58/76
Variably dimensioned	12	22/37	20/39	20/39	23/27	23/27
Extended powell singular	16	206/218	127/145	154/176	125/166	78/101

$\log_2$ scale performance profile of the considered algorithms based on n_i and n_f. The left and right figures are drawn in terms of n_i and n_f, respectively. Note that, for every $\tau \geq 1$, the performance profile gives the proportion $P(\tau)$ of test problems in which each considered algorithms has a performance within a factor τ of the best; see [9] for more complete discussion.

The main focus in the results of Table 1 is to show the efficiency of the new proposed adaptive radius. It can be easily seen from Fig. 1 and Table 1 that the new proposed adaptive radius works very well, especially in the sense that it solves all the considered test problems without any failure, while the other algorithms fail in some problems. In Table S1 (see on-line supplementary material), our aim is to show the efficiency of Algorithm 2.1, especially on large scale problems. As it is clear

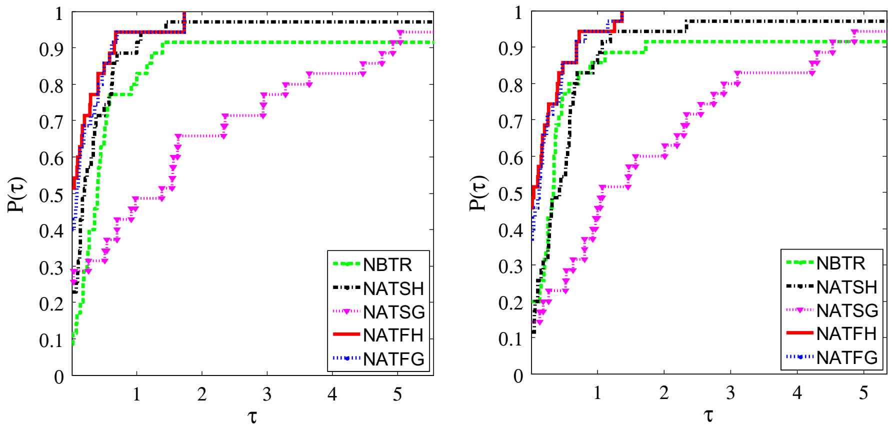

Fig. 1 Performance profile based on n_i (*left*) and n_f (*right*) for the results of Table 1

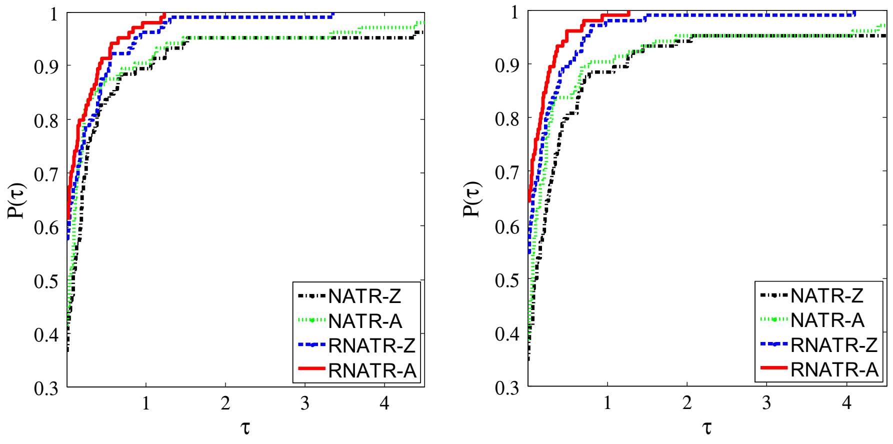

Fig. 2 Performance profile based on n_i (*left*) and n_f (*right*) for the results of Table S1 (see on-line supplementary material)

from Fig. 2, one can easily see that: Firstly, the RNATR-A is the best performed algorithm among the considered algorithms, as it solves about 65 % of test problems in minimum number of n_i and n_f. Secondly, the performance index of RNATR-A grows up rapidly in comparison with the other considered algorithms. It means that in the cases that RNATR-A is not the best algorithm; its performance index is close to the performance index of the best algorithm. Moreover, by comparing the performance of NATR-Z and RNATR-Z, one can see that the relaxation technique results in a great increase in the number of test problems that are solved in the minimum number of n_i and n_f. In addition, in contrast to NATR-Z, the RNATR-Z algorithm solves all the considered test problems without any failure. Analogous behaviors can be seen in the performance of NATR-A and RNATR-A. Therefore, the new relaxation technique has a great influence in the performance of the new proposed algorithm. As an illustration

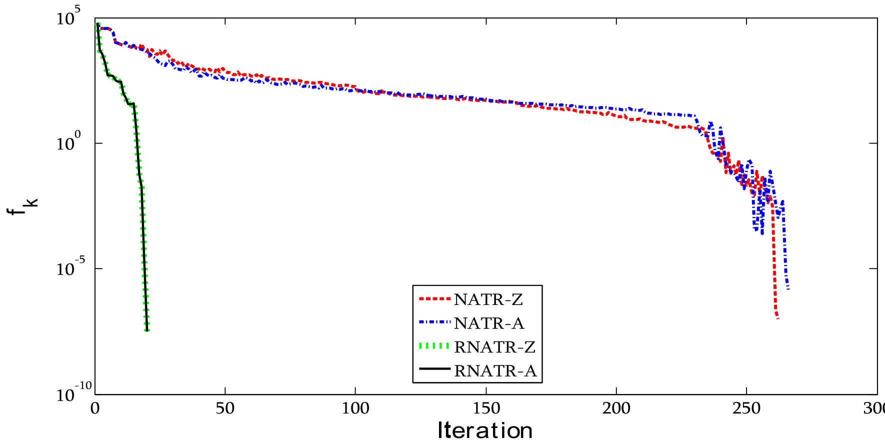

Fig. 3 The value of *Extended Rosenbrock* function on successive iterations for nonmonotone adaptive trust region methods (NATR-Z and NATR-A), and relaxed nonmonotone adaptive trust region methods (RNATR-Z and RNATR-A) with dimension $n = 5{,}000$

of the effect of our proposed relaxation technique, a typical example is shown in Fig. 3, where the sequences $\{f_k\}$ are plotted against k for both nonmonotone and the relaxed nonmonotone techniques.

6 Conclusions

In this paper, a new relaxed nonmonotone adaptive trust region method for solving unconstrained optimization problems is presented. The new proposed algorithm incorporates the Shi and Guo's adaptive trust region method, proposed in [23], with a new nonmonotone term inspired by that proposed in [2]. Under some standard assumptions, theoretical results show that the new algorithm inherits global convergence property of standard TR methods. We have also established the superlinear convergence rate of the new proposed algorithm. Numerical results confirm that the new nonmonotone adaptive strategy provides a powerful tool for the algorithm to perform efficiently in practice, too.

Acknowledgments The authors would like to thank the Research Councils of K.N. Toosi University of Technology and the SCOPE research center for supporting this research. The authors would also like to appreciate the anonymous referees for generously providing us insightful comments which significantly improved the quality of the paper.

Appendix

Proof of Lemma 3.6 We proceed by induction on L. For $L = 0$, using Lemma 3.3, we have

$$f_{k_r} \leq |f_0| \prod_{i=0}^{k_r-1} (1+\varphi_i) - \omega_{k_r-1} \leq |f_0| \prod_{i=0}^{k_r-1} (1+\varphi_i) - \tilde{\omega}_0, \quad r = 0, 1, \ldots, M-1,$$

where $\tilde{\omega}_0 = \min_{0\le r\le M-1} \omega_{k_r}$. Suppose that (24) holds for $L = t$ (the induction hypothesis). For $L = t+1$, we show that

$$f_{k_{(t+1)M+r}} \le |f_0| \prod_{i=0}^{k_{(t+1)M+r}-1} (1+\varphi_i) - \sum_{i=0}^{t+1} \tilde{\omega}_i,\ r = 0, 1, \ldots, M-1. \tag{30}$$

For this purpose, we proceed by induction on r. From (19), for $r = 0$, we have:

$$\begin{aligned} f_{k_{(t+1)M}} &\le \left(1+\varphi_{k_{(t+1)M}-1}\right) R_{k_{(t+1)M}-1} - \omega_{k_{(t+1)M}-1} \\ &\le \left(1+\varphi_{k_{(t+1)M}-1}\right) f_{l(k_{(t+1)M}-1)} - \omega_{k_{(t+1)M}-1}. \end{aligned}$$

Due to the definition of $f_{l(j)}$ in Remark 2.1, we have

$$f_{l(k_{(t+1)M}-1)} = f_{l(k_{(t+1)M-1})} = f_{l(k_{tM+M-1})}. \tag{31}$$

Thus, from the latter inequality, we have:

$$f_{k_{(t+1)M}} \le \left(1+\varphi_{k_{(t+1)M}-1}\right) f_{l(k_{tM+M-1})} - \omega_{k_{(t+1)M}-1}. \tag{32}$$

Now, using (31), (32), the induction hypothesis (over L) and the fact that $l(k_{(t+1)M}-1) \le k_{(t+1)M} - 1$, we have

$$\begin{aligned} f_{k_{(t+1)M}} &\le \left(1+\varphi_{k_{(t+1)M}-1}\right) \left\{ |f_0| \prod_{i=0}^{l(k_{tM+M-1})-1} (1+\varphi_i) - \sum_{i=0}^{t} \tilde{\omega}_i \right\} - \omega_{k_{(t+1)M}-1} \\ &\le \left(1+\varphi_{k_{(t+1)M}-1}\right) \left\{ |f_0| \prod_{i=0}^{l(k_{(t+1)M}-1)-1} (1+\varphi_i) - \sum_{i=0}^{t} \tilde{\omega}_i \right\} - \omega_{k_{(t+1)M}-1} \\ &\le \left(1+\varphi_{k_{(t+1)M}-1}\right) \left\{ |f_0| \prod_{i=0}^{k_{(t+1)M}-2} (1+\varphi_i) - \sum_{i=0}^{t} \tilde{\omega}_i \right\} - \omega_{k_{(t+1)M}-1} \\ &= |f_0| \prod_{i=0}^{k_{(t+1)M}-1} (1+\varphi_i) - \left(1+\varphi_{k_{(t+1)M}-1}\right) \sum_{i=0}^{t} \tilde{\omega}_i - \omega_{k_{(t+1)M}-1} \\ &\le |f_0| \prod_{i=0}^{k_{(t+1)M}-1} (1+\varphi_i) - \sum_{i=0}^{t} \tilde{\omega}_i - \omega_{k_{(t+1)M}-1} \\ &\le |f_0| \prod_{i=0}^{k_{(t+1)M}-1} (1+\varphi_i) - \sum_{i=0}^{t} \tilde{\omega}_i - \tilde{\omega}_{t+1}, \end{aligned}$$

where the second inequality is obtained from the fact that $\varphi_i \geq 0$. Assume that (30) holds for $r = j \leq M-2$ (induction hypothesis). For $r = j+1$, from (21), we have:

$$\begin{aligned} f_{k_{(t+1)M+j+1}} &\leq \left(1+\varphi_{k_{(t+1)M+j+1}-1}\right) R_{k_{(t+1)M+j+1}-1} - \omega_{k_{(t+1)M+j+1}-1} \\ &\leq \left(1+\varphi_{k_{(t+1)M+j+1}-1}\right) f_{l\left(k_{(t+1)M+j+1}-1\right)} - \omega_{k_{(t+1)M+j+1}-1}. \end{aligned}$$

From Remark 2.1, we have

$$f_{l\left(k_{(t+1)M+j}\right)} = f_{l\left(k_{(t+1)M+j+1}-1\right)}.$$

Moreover, as $l\left(k_{(t+1)M+j}\right) \leq l\left(k_{(t+1)M+j+1}-1\right) \leq k_{(t+1)M+j+1}-1$, then from induction hypothesis (over r), we obtain:

$$\begin{aligned} f_{k_{(t+1)M+j+1}} &\leq \left(1+\varphi_{k_{(t+1)M+j+1}-1}\right)\left\{|f_0| \prod_{i=0}^{l\left(k_{(t+1)M+j}\right)-1}(1+\varphi_i) - \sum_{i=0}^{t}\tilde{\omega}_i\right\} - \omega_{k_{(t+1)M+j+1}-1} \\ &\leq \left(1+\varphi_{k_{(t+1)M+j+1}-1}\right)\left\{|f_0| \prod_{i=0}^{l\left(k_{(t+1)M+j+1}-1\right)-1}(1+\varphi_i) - \sum_{i=0}^{t}\tilde{\omega}_i\right\} - \omega_{k_{(t+1)M+j+1}-1} \\ &\leq \left(1+\varphi_{k_{(t+1)M+j+1}-1}\right)\left\{|f_0| \prod_{i=0}^{k_{(t+1)M+j+1}-2}(1+\varphi_i) - \sum_{i=0}^{t}\tilde{\omega}_i\right\} - \omega_{k_{(t+1)M+j+1}-1} \\ &\leq |f_0| \prod_{i=0}^{k_{(t+1)M+j+1}-1}(1+\varphi_i) - \left(1+\varphi_{k_{(t+1)M+j+1}-1}\right)\sum_{i=0}^{t}\tilde{\omega}_i - \omega_{k_{(t+1)M+j+1}-1} \\ &\leq |f_0| \prod_{i=0}^{k_{(t+1)M+j+1}-1}(1+\varphi_i) - \sum_{i=0}^{t}\tilde{\omega}_i - \tilde{\omega}_{t+1} \\ &\leq |f_0| \prod_{i=0}^{k_{(t+1)M+j+1}-1}(1+\varphi_i) - \sum_{i=0}^{t+1}\tilde{\omega}_i. \end{aligned}$$

This completes the proof of the lemma. □

References

1. Ahookhosh, M., Amini, K.: A nonmonotone trust region method with adaptive radius for unconstrained optimization. Comput. Math. Appl. **60**, 411–422 (2010)
2. Ahookhosh, M., Amini, K.: An efficient nonmonotone trust-region method for unconstrained optimization. Numer. Algorithms **59**, 523–540 (2012)
3. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**(1), 147–161 (2008)
4. Chamberlain, R.M., Powell, M.J.D., Lemarechal, C., Pedersen, H.C.: The watchdog technique for forcing convergence in algorithm for constrained optimization. Math. Program. Stud. **16**, 1–17 (1982)
5. Conn, A., Gould, N., Toint, PhL: Trust Region Method. SIAM, Philadelphia (2000)
6. Cui, Z., Wu, B.: A new modified nonmonotone adaptive trust region method for unconstrained optimization. Comput. Optim. Appl. **53**, 795–806 (2012)
7. Dai, Y.H.: On the nonmonotone line search. J. Optim. Theory Appl. **112**(2), 315–330 (2002)

8. Deng, N.Y., Xiao, Y., Zhou, F.J.: Nonmonotonic trust region algorithm. J. Optim. Theory Appl. **76**, 259–285 (1993)
9. Dolan, E., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
10. Fan, J.Y., Yuan, Y.X.: A new trust region algorithm with trust region radius converging to zero. In: Proceedings of the 5th International Conference on Optimization: Techniques and Applications [C], Hong Kong (2001)
11. Fu, J.H., Sun, W.Y.: Nonmonotone adaptive trust-region method for unconstrained optimization problems. Appl. Math. Comput. **63**, 489–504 (2005)
12. Gertz, E.M.: A quasi-Newton trust-region method. Math. Program. **100**, 447–470 (2004)
13. Grippo, L., Lamparillo, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
14. Grippo, L., Lamparillo, F., Lucidi, S.: A truncated Newton method with nonmonotone line search for unconstrained optimization. J. Optim. Theory Appl. **60**(3), 401–419 (1989)
15. Lin, C.J., Moré, J.J.: Newton's method for large bound-constrained optimization problems. SIAM J. Optim. **9**(4), 1100–1127 (1999)
16. Moré, J.J., Grabow, B.S., Hillstrom, K.E.: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**, 17–41 (1981)
17. Nocedal, J., Wright, S.J.: Numerical Optimization. Springer, NewYork (2006)
18. Nocedal, J., Yuan, Y.: Combining trust region and line search techniques. In: Yuan, Y. (ed.) Advances in Nonlinear Programming, pp. 153–175. Kluwer Academic Publishers, Dordrecht (1996)
19. Panier, E.R., Tits, A.L.: Avoiding the Maratos effect by means of a nonmonotone linesearch. SIAM J. Numer. Anal. **28**, 1183–1195 (1991)
20. Powell, M.J.D.: On the global convergence of trust region algorithms for unconstrained optimization. Math. Program. **29**, 297–303 (1984)
21. Sang, Z., Sun, Q.: Aself-adaptive trust region method with linesearch based on a simple subproblem model. J. Comput. Appl. Math. **232**, 514–522 (2009)
22. Sartenaer, A.: Automatic determination of an initial trust region in nonlinear programming. SIAM J. Sci. Comput. **18**(6), 1788–1803 (1997)
23. Shi, Z.J., Guo, J.H.: A new trust region method for unconstrained optimization. J. Comput. Appl. Math. **213**, 509–520 (2008)
24. Shi, Z.J., Wang, H.Q.: A new self-adaptive trust region method for unconstrained optimization. Technical Report, College of Operations Research and Management, Qufu Normal University (2004)
25. Toint, PhL: An assessment of nonmonotone linesearch technique for unconstrained opti-mization. SIAM J. Sci. Comput. **17**, 725–739 (1996)
26. Toint, PhL: Non-monotone trust-region algorithm for nonlinear optimization subject to convex constraints. Math. Program. **77**, 69–94 (1997)
27. Xiao, Y., Chu, E.K.W.: Nonmonotone trust region methods. Technical Report 95/17, Monash University, Clayton, Australia (1995)
28. Xiao, Y., Zhou, F.J.: Nonmonotone trust region methods with curvilinear path in uncon-strained optimization. Computing **48**, 303–317 (1992)
29. Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. **14**(4), 1043–1056 (2004)
30. Zhang, X.S., Zhang, J.L., Liao, L.Z.: An adaptive trust region method and its convergence. Sci. China **45**, 620–631 (2002)
31. Zhang, X.S., Zhang, J.L., Liao, L.Z.: A nonmonotone adaptive trust region method and its convergence. Comput. Math. Appl. **45**, 1469–1477 (2003)
32. Zhou, F., Xiao, Y.: A class of nonmonotone stabilization trust region methods. Computing **53**(2), 119–136 (1994)