Journal of Mathematical Modeling
Vol. 11, No. 3, 2023, pp. 507–526. Research Article

JMM

On determining radius in nonmonotone trust-region approaches

Keyvan Amini[†]*, **Mehri Rashidi**[‡]

[†]*Department of Mathematics, Faculty of Science, Razi University,Kermanshah, Iran*
[‡]*Faculty of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran*
Email(s): kamini@razi.ac.ir, mrashidi@aut.ac.ir

Abstract. This paper proposes two effective nonmonotone trust-region frameworks for solving nonlinear unconstrained optimization problems while provide a new effective policy to update the trust-region radius. Conventional nonmonotone trust-region algorithms apply a specific nonmonotone ratio to accept new trial step and update the trust-region radius. This paper recommends using the nonmonotone ratio only as an acceptance criterion for a new trial step. In contrast, the monotone ratio or a hybrid of monotone and nonmonotone ratios is proposed as a criterion for updating the trust-region radius. We investigate the global convergence to first- and second-order stationary points for the proposed approaches under certain classical assumptions. Initial numerical results indicate that the proposed methods significantly enhance the performance of nonmonotone trust-region methods.

Keywords: Unconstrained optimization, trust-region framework, trust-region radius, nonmonotone technique.
AMS Subject Classification 2010: 90C30 . 65K05

1 Introduction

Trust-region techniques are a common category of iterative procedures for solving optimization problems. Powell first described them in 1970 in [21] to solve unconstrained optimization problems. In a sense, the technique of trust-region approaches is analogous to the conventional Levenberg-Marquardt strategy used to solve nonlinear least-squares and nonlinear equations problems. This procedure was first introduced by Levenberg in [15] and later proposed by Marquardt in [16]. Trust-region approaches have had a well-behaved and superior reputation for the past four decades due to their outstanding numerical reliability and strong theoretical convergence properties. Numerous researchers examined and employed

*Corresponding author
Received: 11 April 2023 / Revised: 6 June 2023 / Accepted: 8 June 2023
DOI: 10.22124/JMM.2023.24275.2174

the trust-region method to address multimodal problems, optimal control, and optimization branches (see, for instance, [8, 20]).

Trust-region methods minimize the model within the trust region or the region surrounding the current point at each iteration by approximating the objective function with a quadratic or conic model. Trust-region approaches produce new trial steps by solving a quadratic subproblem subject to the trust region. If the trial step reduces the model sufficiently within the trust region, the algorithm expands the trust region. If the model and objective function disagrees, the algorithm reduces the trust region. We solve the subproblem to find an acceptable step.

The traditional trust-region structure provides strong convergence theories and reliable computational results, but it also has some significant drawbacks. It generates sequential iterations that yield a monotone objective function value, resulting in slow convergence in highly nonlinear problems and a narrow, curved valley; see [1–3, 8, 17]. The nonmonotone technique is one of the most promising iterative optimization methods to overcome this obstacle. Inspired by the watchdog strategy proposed for the standard line search condition in [6] to overcome the Marotos effect, Grippo et al. in [12] propose an innovative nonmonotone line search technique for the Newton method. Based on their results, the new approach might improve the performance of Armijo-type line search frameworks. Nonmonotone approaches can also accelerate algorithm convergence and increase the probability of finding the global optimum (see [8, 20]). Numerous researchers are interested in employing nonmonotone methods in various disciplines of optimization procedures due to their remarkable efficiency; see [1–3, 12–14, 17, 22, 23].

This study investigates the impact of nonmonotone techniques on updating the trust region and accepting new points. In this research, we believed that the best results might be achieved whenever a method employs a nonmonotone ratio to take a new trial step. To develop a strategy for updating the trust-region radius, it uses the traditional monotone trust-region ratio or a hybrid with a nonmonotonic ratio. According to the trust-region methodology, a larger radius leads to a larger number of subproblems being solved, increasing computation cost. This observation leads us to conclude that most nonmonotone trust-region-based methods extend the trust-region radius beyond what is necessary to attain the best convergence for real-world problems. Using some nonmonotone ratios as the acceptance criterion and the standard monotone ratio or its hybrid with a nonmonotone ratio to update the trust-region radius, we will introduce new trust-region-based algorithms.

According to our analysis, the modified nonmonotone trust-region framework deals with problems with a narrow curved valley in somewhat the same manner as the traditional trust-region method; thus, the radius is not sufficiently expanded compared to the traditional nonmonotone trust-region framework. Thus, the proposed method attempts to identify narrow, curved valleys based on how frequently a very successful iteration occurs. If the number of very successful iterations exceeds an integer constant, the algorithm may encounter a narrow, curved valley. In this situation, nonmonotone ratios can be used to determine the trust region's radius. In this way, we can state that the proposed method does necessarily not lead to a smaller trust-region radius compared to the traditional trust-region framework. According to the analysis, the innovative strategies possess both the reliability of trust-region frameworks and the effectiveness of nonmonotone approaches. In addition, the global convergence of the proposed techniques, we demonstrate the efficacy and reliability of the proposed approaches in practice by employing them in some numerical experiments on an extensive collection of standard, unconstrained test problems.

The remainder of the paper is structured as follows. Section 2 describes two novel trust-region-based algorithms. The global convergence of the proposed algorithms to first- and second-order stationary points are the subject of Section 3, which follows a discussion of the algorithms' features. We provide initial numerical results for the approaches described in Section 4. Section 5 represents the conclusion of the paper.

2 Algorithmic framework

In this section, we begin by discussing how conventional trust-region methods work. Then, taking into account two well-known nonmonotone trust-region algorithms, we propose two new variant nonmonotone trust-region algorithms with novel approaches to determining radius. Let $f:\mathbb{R}^n \to \mathbb{R}$ be a twice continuously differentiable function. Traditionally this has been considered as the following unconstrained optimization problem

$$\text{Minimize } f(x), \quad \text{subject to } x \in \mathbb{R}^n. \tag{1}$$

As discussed in the preceding section, traditional trust-region algorithms seek a neighborhood near the current step x_k in which a quadratic model should agree with the objective function. We solve the following quadratic subproblem to produce a trial step d_k.

$$\text{Minimize } m_k(d) = f_k + g_k^T d + \frac{1}{2} d^T B_k d, \quad \text{subject to } d \in \mathbb{R}^n \text{ and } \|d\| \le \Delta_k, \tag{2}$$

where $\|.\|$ is the Euclidean norm, $f_k = f(x_k)$, $g_k = \nabla f(x_k)$, B_k is the exact Hessian $G_k = \nabla^2 f(x_k)$ or its symmetric approximation, and Δ_k is a trust-region radius. The method should decide whether or not it can be accepted after obtaining d_k. The conventional monotone trust-region framework uses the following ratio to achieve this goal:

$$\rho_k = \frac{f_k - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)}, \tag{3}$$

where the numerator represents the actual reduction, and the denominator represents the predicted reduction. The algorithm decides whether to accept or reject a trial step and whether to update the existing trust-region radius Δ_k based on the value of this ratio. Consider the situation where ρ_k is close to 1 in the current region. It indicates that the model and the objective function are in accordance. If so, the algorithm accepts the trial step. In this case, increasing the trust-region radius Δ_k for the subsequent iteration is safe. The approach rejects the trial step and reduces the trust region Δ_k if ρ_k is a tiny positive or negative number, indicating that the model and the objective are not in satisfactory agreement.

Grippo et al. in [12] developed the earliest nonmonotone strategy for the Newton method, inspired by the watchdog technique proposed by Chamberlain et al. in [6] to avoid the Maratos effect. Their proposal suggests that the steplength α_k can be accepted whenever

$$f(x_k + \alpha_k d_k) \le f_{l(k)} + \beta \alpha_k \nabla f(x_k)^T d_k, \tag{4}$$

in which $\beta \in (0,1)$ and

$$f_{l(k)} = \max_{0 \le j \le m(k)} \{f_{k-j}\}, \quad \text{for all } k \in \mathbb{N} \cup \{0\}, \tag{5}$$

where $m(0) = 0$, $0 \le m(k) \le \min\{m(k-1)+1, N\}$ and $N \ge 0$. In highly nonlinear problems and the presence of a narrow, curved valley, their conclusions are particularly relevant to the nonmonotone approach.

Based on the exciting behavior of this nonmonotone technique, numerous researchers have effectively developed the nonmonotone strategy for other optimization branches. Despite its many benefits, the classic nonmonotone (5), has several drawbacks; see, for instance, [1, 22]. For instance, Zhang and Hager in [23] proposed a novel nonmonotone technique together within a line search framework to circumvent some disadvantages. The approach relaxes the Armijo-type condition (4) by replacing $f_{l(k)}$ with C_k, which is defined as

$$C_k = \begin{cases} f_0, & k = 0, \\ (\eta_{k-1} Q_{k-1} C_{k-1} + f_k)/Q_k, & k \geq 1, \end{cases} \tag{6}$$

where Q_k is specified with

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

and $0 \leq \eta_{min} \leq \eta_{k-1} \leq \eta_{max} \leq 1$. This method, in our opinion, is thriving and promising when addressing unconstrained optimization problems. Nonmonotonic techniques for line search procedures have yielded significant results, motivating researchers to examine their influence on trust-region frameworks. see [1–3, 17]. Therefore, other trust-region ratios have been proposed, with the following ratio being the most popular nonmonotone trust-region ratio.

$$\tilde{\rho}_k = \frac{f_{l(k)} - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)}, \tag{7}$$

where $f_{l(k)}$ is defined by (5) and the numerator is called the nonmonotone reduction. As another example, Mo, Liu, and Yan in [17] take advantage of the nonmonotone strategy (6) of Zhang and Hager in the trust-region framework to propose the following ratio

$$\hat{\rho}_k = \frac{C_k - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)}. \tag{8}$$

The theoretical and computational results of this method are noticeably fascinating for unconstrained optimization problems.

The traditional nonmonotone trust-region framework employs this ratio as a measure of acceptance for the trial step and as a criterion for updating the radius after calculating the ratio of the nonmonotone reduction to the predicted one. For example, if d_k is a subproblem solution and $\hat{\rho}_k$ is the applied ratio, then the trust-region radius is usually updated by

$$\Delta_{k+1} = \begin{cases} \max\left[\Delta_k, \gamma_2 \|d_k\|\right], & \hat{\rho}_k \geq \mu_2, \\ \Delta_k, & \mu_1 \leq \hat{\rho}_k < \mu_2, \\ \gamma_1 \|d_k\|, & \hat{\rho}_k < \mu_1, \end{cases}$$

where $0 < \mu_1 < \mu_2 \leq 1$ and $0 < \gamma_1 < 1 < \gamma_2$, and the new point is accepted if $\hat{\rho}_k \geq \mu_1$. As a consequence of the fact that $C_k \geq f_k$, it is clear that $\hat{\rho}_k \geq \rho_k$, i.e., it seems that if the problem is not highly nonlinear, then the number of iterates that for which $\hat{\rho}_k \geq \mu_2$ is grown, so the procedure increases the trust-region radius more than needed. To overcome this disadvantage, we believe it is logical to exploit a nonmonotone ratio for the measure of acceptance and employ the monotone ratio (3) to update the trust-region radius.

Suppose A_k is a nonmonotone term in kth iteration. Based on the considered discussion, we summarize the new nonmonotone trust-region procedure.

Algorithm 1: Modified nonmonotone trust-region framework 1

Input: An initial point $x_0 \in \mathbb{R}^n$, a symmetric positive definite matrix $B_0 \in \mathbb{R}^{n\times n}$, an integer k_{max} and parameters $0 < \eta_0 < 1$, $0 \leq \mu_1 < \mu_2 < 1$, $0 < \gamma_1 < 1 < \gamma_2$ and $\Delta_0 > 0$ and $\varepsilon > 0$. **Begin**

$f_0 \leftarrow f(x_0)$; $A_0 \leftarrow f_0$; $g_0 \leftarrow g(x_0)$; $k \leftarrow 0$;

While ($\|g_k\| \geq \varepsilon$ **and** $k \leq k_{max}$)

Step 1:

Determine a trial point d_k by solving the subproblem (2);

$\hat{f}_{k+1} \leftarrow f(x_k + d_k)$;

Step 2:

$\widehat{Ared} \leftarrow A_k - \hat{f}_{k+1}$;

$Pred \leftarrow m(x_k) - m(\hat{x}_{k+1})$;

$\hat{\rho}_k \leftarrow \widehat{Ared}/Pred$;

Step 3: {**Trust-region update**}

$Ared \leftarrow f_k - \hat{f}_{k+1}$;

$\rho_k \leftarrow Ared/Pred$;

If $\rho_k \geq \mu_2$ **Then** Enlarge the trust-region radius by choosing

$\Delta_{k+1} = \max\,[\Delta_k, \gamma_2\|d_k\|]$;

else if $\mu_1 \leq \rho_k < \mu_2$ **Then**

$\Delta_{k+1} = \Delta_k$;

else Reduce the trust-region radius by selecting

$\Delta_k = \gamma_1\|d_k\|$; Go to Step 1;

end if;

Step 4: {**Trust-region acceptance**}

If $\hat{\rho}_k > \mu_1$ **Then**

$x_{k+1} = x_k + d_k$;

$f_{k+1} = \hat{f}_{k+1}$;

End if ;

Step 5: {**Parameters update**}

Update B_k by a quasi-Newton updating formula and compute B_{k+1};

Compute $g_{k+1} = g(x_{k+1})$;

Generate A_{k+1}.

$k \leftarrow k+1$;

End While

End

Here, the `While` loop containing Steps 1-5 is called outer cycle and the loop including Steps 1 to `Go to Step 1` in Step 3 is named inner cycle.

Remark 1 *There are some choices for A_k. Here, we use two standard choices. The first choice is $A_k = C_k$ defined by (6) and the second choice is $A_k = f_{l(k)}$ defined by (5). We call Algorithm 1 by NTRM-1 and NTRG-1 when we use $A_k = C_k$ and $A_k = f_{l(k)}$ respectively.*

Definition 1. We define any iterate of Algorithm 1 satisfying $\hat{\rho}_k \geq \mu_1$, causing $x_{k+1} = x_k + d_k$, as a successful iterate. Furthermore, we identify it as a very successful iterate if this iterate satisfies $\rho_k \geq \mu_2$.
Example 1. To study the updating process of the nonmonotone trust region framework in depth, we select two famous test functions from [18] and apply NTRG-1 and NTRM-1 and their traditional versions, NTRG and NTRM, were proposed by Grippo et al. in [13] and Mo et al. in [17] on these two problems, to show the various performances of these algorithms. We summarized the corresponding results in the following table.

Table 1. Results for (Iterates/Function evaluations)

Problem name	Dim	NTRG	NTRG-1	NTRM	NTRM-1
Penalty function II	100	491/606	140/149	298/368	139/147
Brown badly scale function	2	17/19	78/78	38/39	78/78

The results for the first problem indicate that the proposed nonmonotone trust-region method is more effective than the traditional nonmonotone framework. In contrast, the results for the second problem show the reverse. To comprehend the cause behind these observations, we must closely examine the functions' features. Both functions are highly nonlinear, but only the second function has a narrow, curved valley. According to our analysis, the modified nonmonotone trust-region framework handles this problem similar to the traditional trust-region method; therefore, the radius is not sufficiently enlarged in contrast to the conventional nonmonotone framework.

Based on the outcomes given in Example 1, we must adjust the proposed nonmonotone trust-region structure to have better behavior when confronted with highly nonlinear problems and when a narrow valley meets. Obviously, in narrow, curved valleys, the actual reduction is significantly more than predicted, it is apparent that $\rho_k > \mu_2$ in most cases. Hence, we suggest a procedure to identify narrow curved valleys based on how often the condition $\rho_k > \mu_2$ is consecutively satisfied. As a result, if the number of observations in this condition is more than an integer constant like $\mathscr{I}$, then one can conclude that the algorithm encounters a narrow curved valley. So some nonmonotone ratios can be used to determine the trust-region radius. According to this discussion, by setting `Flag` as the counter of the condition $\rho_k > \mu_2$, we introduce the following framework for the nonmonotone trust-region procedures:

Algorithm 2: Modified nonmonotone trust-region framework 2

Keep all the steps of Algorithm 1 , let a constant $\mathscr{I} \in \mathbb{N}$ is given, set $\texttt{Flag} = 0$. only modify Step 3 by the following step:
Step 3: $\{$**Trust-region update**$\}$
if $\rho_k \geq \mu_2$ **Then** Enlarge the trust-region radius by choosing $\Delta_{k+1} = \max[\Delta_k, \gamma_2 \|d_k\|]$; $\texttt{Flag} \leftarrow \texttt{Flag} + 1$;
else if $\texttt{Flag} \geq \mathscr{I}$ **and** $\hat{\rho}_k \geq \mu_2$ **Then** Enlarge the trust-region radius by choosing $\Delta_{k+1} = \max[\Delta_k, \gamma_2 \|d_k\|]$;
else if $\mu_1 \leq \rho_k < \mu_2$ **Then** $\Delta_{k+1} = \Delta_k$;
else Reduce the trust-region radius by selecting $\Delta_k = \gamma_1 \|d_k\|$; $\texttt{Flag} \leftarrow 0$; Go to Step 1;
end if

In this point, we call Algorithm 2 by NTRM-2 when $A_k = C_k$ and NTRG-2 if $A_k = f_{l(k)}$. Based on what discussed in Step 3 of Algorithm 2, we require to define a new meaning for the term "very successful iterates." For this purpose, we introduce the following definition:

Definition 2. In Algorithm 2, when (`Flag` $\geq \mathscr{I}$ and $\hat{\rho}_k \geq \mu_2$) or $\rho_k \geq \mu_2$, the trust-region radius is enlarged and iteration is named a very successful iterate.

3 Global convergence properties

This section is devoted to the theoretical analysis including the global convergence to first- and second-order stationary points of the new algorithms for solving unconstrained nonlinear optimization problems. First of all, we prove that any limit point x_* of the sequence $\{x_k\}$ generated by Algorithm 2 satisfies $g(x_*) = 0$, paying no attention to the choice of the arbitrary starting point x_0 and the initial trust-region radius Δ_0. Then the aim of ensuring convergence to second-order stationary points is followed based on employing the second order information of the objective function. In the rest of this section, Algorithm 2 means NTRM-2.

Throughout the paper, the following classical assumptions are considered to analyze the strong global convergence results of the proposed algorithms:

(H1) The objective function $f(x)$ is twice continuously differentiable and has a lower bound on the level set $L(x_0) = \{x \in \mathbb{R}^n | f(x) \leq f(x_0)\}$; that is, there exists a constant κ_{lbf} such that $f(x) \geq \kappa_{\text{lbf}}$, for all $x \in L(x_0)$.

(H2) The approximate Hessian matrixs, B_k , are uniformly bounded, i.e. there exists $\kappa_{\text{umh}} > 0$ such that

$$\|B_k\| \leq \kappa_{\text{umh}}, \text{ for all } k \in \mathbb{N} \cup \{0\}.$$

(H3) The decrease on the model m_k is at least as much as a fraction of that obtained by the Cauchy point, i.e. there exists a constant $\kappa_{\text{mdc}} \in (0,1)$ such that

$$m_k(x_k) - m_k(x_k + d_k) \geq \kappa_{\text{mdc}} \|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right], \tag{9}$$

for all $k \in \mathbb{N} \cup \{0\}$.

As known, we call the condition (9) the sufficient reduction condition throughout trust-region literature. By solving the trust region subproblem based on specific procedures, this condition is easily specified. See for example, [8, 19, 20]. This condition implies $d_k \neq 0$ in the case that $g_k \neq 0$.

Pay attention to the fact that the objective function $f(x)$ is twice continuously differentiable and the level set $L(x_0)$ is bounded along with the assumption (H1) imply that $\|G_k\|$ is uniformly continuous and bounded on the open bounded convex set Ω, containing $L(x_0)$ and $\mathscr{B}_k$. Therefore, there exists a constant $\kappa_{\text{ufh}} > 0$ such that

$$\|G(\xi_k)\| \leq \kappa_{\text{ufh}}, \quad \text{for all } \xi_k \in \mathscr{B}_k, \tag{10}$$

where $\mathscr{B}_k = \{x \in \mathbb{R}^n | \|x - x_k\| \leq \Delta_k\}$. This fact together with the mean value theorem directly conclude that

$$\|g(x) - g(y)\| \leq \kappa_{\text{ufh}} \|x - y\|, \quad \text{for all } x, y \in \Omega$$

which means that $f(x)$ is a Lipschitz function.

Providing an error bound for the distance between the objective function and its current quadratic model (2) at the new iterate $x_k + d_k$ is the first step of our analysis.

Lemma 1. *Suppose that (H1) and (H2) hold and the sequence $\{x_k\}$ be generated by Algorithm 2. Then we have*

$$|f(x_k+d_k)-m_k(x_k+d_k)| \le \kappa_{ubh}\,\Delta_k^2,$$

where $\kappa_{ubh} = \max\left[\kappa_{ufh}, \kappa_{umh}\right]$.

Proof. Based on the Taylor expansion and the mean value theorem, we obtain that

$$f(x_k+d_k) = f_k + g_k^T d_k + \frac{1}{2} d_k^T G(\zeta_k) d_k,$$

for some ζ_k in the line segment $[x_k, x_k+d_k]$.

This fact, the definition of $m_k(d)$, (H2) and (10) imply that

$$\begin{aligned}
|f(x_k+d_k)-m_k(x_k+d_k)| &= |f_k + g_k^T d_k + \frac{1}{2} d_k^T G(\zeta_k) d_k - f_k - g_k^T d_k - \frac{1}{2} d_k^T B_k d_k| \\
&\le \frac{1}{2}\left[|d_k^T G(\zeta_k) d_k| + |d_k^T B_k d_k|\right] \\
&\le \frac{1}{2}(\kappa_{\mathrm{ufh}} + \kappa_{\mathrm{umh}})\|d_k\|^2 \\
&\le \kappa_{\mathrm{ubh}}\Delta_k^2.
\end{aligned}$$

So the proof is completed. □

We first prove the following lemma. This lemma is necessary to establish that Algorithm 2 will finitely terminate and to yield evidence about the convergence of the sequence $\{C_k\}$.

Lemma 2. *Suppose that (H3) holds and the sequence $\{x_k\}$ be generated by Algorithm 2. Then we have*

$$f_{k+1} \le C_{k+1} \le C_k, \tag{11}$$

for all $k \in \mathbb{N} \cup \{0\}$.

Proof. Let iterate k be a successive iterate so that, from $\hat{\rho}_k \ge \mu_1$ and (9), we easily get

$$f_{k+1} \le C_k - \mu_1 \kappa_{\mathrm{mdc}}\,\|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right]. \tag{12}$$

This fact together with (6) imply that

$$\begin{aligned}
C_{k+1} &= \frac{\eta_k Q_k C_k + f_{k+1}}{Q_{k+1}} \\
&\le \frac{\eta_k Q_k C_k + C_k - \mu_1 \kappa_{\mathrm{mdc}}\,\|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right]}{Q_{k+1}} \\
&= C_k - \frac{\mu_1 \kappa_{\mathrm{mdc}}\,\|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right]}{Q_{k+1}},
\end{aligned}$$

which clearly suggests

$$C_{k+1} \le C_k. \tag{13}$$

On the other hand, if $\eta_k \neq 0$, from (6), we obtain

$$C_{k+1} - C_k = \frac{f_{k+1} - C_{k+1}}{\eta_k Q_k}. \tag{14}$$

Thus, by (13) and (14), we get $f_{k+1} \leq C_{k+1} \leq C_k$, for all $k \in \mathbb{N} \cup \{0\}$. If $\eta_k = 0$, then we have $C_{k+1} = f_{k+1}$. Therefore, in both cases (11) holds, and so the proof is completed. □

As a result of Lemma 2, it can be deduced that the sequence $\{C_k\}$ is non-increasing. This fact along with (H1) suggest that there exists a real-valued constant κ_{lbf} such that

$$\kappa_{\text{lbf}} \leq f_{k+n} \leq C_{k+n} \leq \cdots \leq C_{k+1} \leq C_k,$$

for all $n \in \mathbb{N} \cup \{0\}$, i.e., the non-increasing sequence $\{C_k\}$ has a lower bound. This obviously leads to the fact that the sequence $\{C_k\}$ is convergent. This result can be summarized in the following corollary.

Corollary 1. *Suppose that (H1)–(H3) hold and the sequence $\{x_k\}$ be generated by Algorithm 2. Then the sequence $\{C_k\}$ is convergent.*

Lemma 1 states that the objective function will be adequately decreased by minimizing the model inside a small trust-region radius. The following theorem illustrates that if the current iteration is not a first-order stationary point and the trust-region radius Δ_k is small enough, the next iteration must be very successful. Considering this result alongside Lemma 2 demonstrates that the inner cycle of the algorithms terminates finitely.

Lemma 3. *Suppose that (H1)–(H3) hold and the sequence $\{x_k\}$ be generated by Algorithm 2. Furthermore, suppose that $g_k \neq 0$ and*

$$\Delta_k \leq \frac{\kappa_{mdc} \|g_k\| (1 - \mu_2)}{\kappa_{ubh}}. \tag{15}$$

Then the iterate k is very successful, i.e. $\Delta_{k+1} \geq \Delta_k$ and the inner cycle of Algorithm 2 is finitely terminated.

Proof. Based on Definition 2, we divide the proof into two following cases:
Case 1. ($\texttt{Flag} < \mathscr{I}$) Using the fact that $\mu_2, \kappa_{\text{mdc}} \in (0, 1)$, it is clear that $\kappa_{\text{mdc}}(1 - \mu_2) < 1$. As a consequence of (15), $\kappa_{\text{umh}} \leq \kappa_{\text{ubh}}$ and (H2) we have

$$\Delta_k < \frac{\|g_k\|}{\kappa_{\text{umh}}} \leq \frac{\|g_k\|}{\|B_k\|}.$$

Using this inequality and (9), we directly obtain

$$m_k(x_k) - m_k(x_k + d_k) \geq \kappa_{\text{mdc}} \, \|g_k\| \, \min\left[\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right] = \kappa_{\text{mdc}} \, \|g_k\| \, \Delta_k. \tag{16}$$

By applying Lemma 1, (15) and (16), we have

$$|\rho_k - 1| = \left| \frac{f_k - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)} - 1 \right| = \left| \frac{f(x_k + d_k) - m_k(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)} \right| \leq \frac{\kappa_{\text{ubh}}}{\kappa_{\text{mdc}} \, \|g_k\|} \, \Delta_k \leq 1 - \mu_2,$$

which means $\rho_k \geq \mu_2$. Thus, the definition 2 implies that the current iterate is very successful.
Case 2. (`Flag` $\geq \mathscr{I}$) Similar to what has been stated in Case 1, one can conclude that $\rho_k \geq \mu_2$. As a result, the relation (11) implies that

$$\hat{\rho}_k = \frac{C_k - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)} \geq \frac{f_k - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)} \geq \mu_2.$$

Therefore, $\hat{\rho}_k \geq \mu_2$ and the current iterate is very successful.

The results of Case 1 and 2 are clearly indicated that $\Delta_{k+1} \geq \Delta_k$. Meanwhile, like Case 1, the fact that $\rho_k \geq \mu_2$ joint with $\mu_2 \geq \mu_1$ and $C_k \geq f_k$ suggest that the condition $\hat{\rho}_k \geq \mu_1$ finally holds so that the inner cycle will be exited in finite steps. Therefore, the proof is completed. □

As a result of Lemma 4, the trust-region radius can not become so small until a first-order stationary point is achieved. It demonstrates how small the trust-region radius Δ_k should depend on $\|g_k\|$ to guarantee the iterate's success. The following lemma ensures this. The detailed proof of this lemma can be seen in Theorem 6.4.3 of Conn et al. in [8].

Lemma 4. *Suppose that (H1)–(H3) hold and the sequence $\{x_k\}$ be generated by Algorithm 2. Also suppose that there exists a constant $\varepsilon > 0$ such that $\|g_k\| \geq \varepsilon$. Then there exists a constant κ_{lbd} such that $\Delta_k \geq \kappa_{lbd}$, for all $k \in \mathbb{N} \cup \{0\}$.*

We know that the best convergence results of nonmonotone methods are obtained by a stronger nonmonotone strategy when iterates are far from the optimum and by a weaker one when iterates are close to the optimum (see [1, 22]). It is clear that Algorithm 2, to some extent, starts with the strong nonmonotone ratio $\hat{\rho}_k$. The following lemma shows that $C_k \approx f_k$, for sufficiently large k, which means that the nonmonotone ratio $\hat{\rho}_k$ is approximately the same as the monotone ratio ρ_k, in the sequel, so that a weaker nonmonotone strategy is employed whenever iterates become close to the local optimum.

Lemma 5. *Suppose that (H1)–(H3) hold and the sequence $\{x_k\}$ be generated by Algorithm 2. Then we have*

$$\lim_{k\to\infty} C_k = \lim_{k\to\infty} f(x_k). \tag{17}$$

Proof. It should be first noticed that $\eta_{max} \in [0,1)$ and $\eta_k \in [\eta_{min}, \eta_{max}]$, for all $k \in \mathbb{N} \cup \{0\}$. Thus, the definition of Q_k straightforwardly gives

$$Q_k = 1 + \sum_{i=0}^{k-1} \prod_{m=0}^{i} \eta_{k-m} \leq 1 + \sum_{i=0}^{k-1} \eta_{max}^{i+1} \leq \sum_{i=0}^{k} \eta_{max}^{i} \leq \sum_{i=0}^{\infty} \eta_{max}^{i} = \frac{1}{1-\eta_{max}}.$$

This leads to

$$\eta_k Q_k \leq \frac{\eta_k}{1-\eta_{max}} \leq \frac{\eta_{max}}{1-\eta_{max}}. \tag{18}$$

Now, if $\eta_k \neq 0$, by (6), we get

$$C_{k+1} - C_k = \frac{f_{k+1} - C_{k+1}}{\eta_k Q_k}. \tag{19}$$

Using (19) and Corollary 3, as $k \to \infty$, we finally obtain

$$\lim_{k\to\infty} \frac{f_{k+1} - C_{k+1}}{\eta_k Q_k} = \lim_{k\to\infty} C_{k+1} - C_k = 0. \tag{20}$$

Therefore, (18) and (20) give our desired result. □

We are now ready to propose the global convergence property to first-order stationary points of Algorithm 2.

Theorem 1. *Suppose that (H1)–(H3) hold and the sequence* $\{x_k\}$ *be generated by Algorithm 2. Then we have*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{21}$$

Proof. To get a result, we divide the proof into two following cases:
Case 1. Assume that the algorithm has finitely successful iterates. Then there exists a constant $k_0 \in \mathbb{N}$ which is the index of the last successful iterate. If $\|g_{k_0+1}\| > 0$, then it follows from Lemma 4 that there must be a very successful iterate of index larger than k_0, which is impossible. This means that $\|g_{k_0+1}\| = 0$ and x_* is a first-order critical point.

Case 2. To drive a contradiction, assume that there exists a constant $\varepsilon > 0$ such that $\|g_k\| \geq \varepsilon$, for all k. We consider a successful iterate with index k. Now, Lemma 5 along with (H1)–(H3) suggest that

$$\begin{aligned} C_k - f_{k+1} &\geq \mu_1 [m_k(x_k) - m_k(x_k + d_k)] \\ &\geq \mu_1 \kappa_{\mathrm{mdc}} \, \|g_k\| \, \min \left[\Delta_k, \frac{\|g_k\|}{\|B_k\|} \right] \\ &\geq \mu_1 \varepsilon \kappa_{\mathrm{mdc}} \, \min \left[\kappa_{\mathrm{lbd}}, \frac{\varepsilon}{\kappa_{\mathrm{umh}}} \right] > 0. \end{aligned}$$

As a consequence, Lemma 6, when $k \to \infty$, immediately gives that

$$0 \geq \mu_1 \varepsilon \kappa_{\mathrm{mdc}} \, \min \left[\kappa_{\mathrm{lbd}}, \frac{\varepsilon}{\kappa_{\mathrm{umh}}} \right],$$

which is an obvious contradiction so that (21) can be held.

Therefore, in both cases, we established that x_* is a first-order critical point of the sequence $\{x_k\}$ and the proof is completed. □

Theorem 7 clearly implies that if the sequence $\{x_k\}$ has some limit points, then at least one of them will satisfy the first-order necessary condition, stating that if this point is x_*, then $g(x_*) = 0$. Now, we prove the stronger result establishing that any limit point of the sequence $\{x_k\}$ is a first-order stationary point.

Theorem 2. *Suppose that (H1)–(H3) hold and the sequence* $\{x_k\}$ *be generated by Algorithm 2. Then we have that*

$$\lim_{k\to\infty} \|g_k\| = 0. \tag{22}$$

Proof. To establish a contradiction, it is assumed that $\lim_{k\to\infty} \|g_k\| \neq 0$. Hence, there exists $\varepsilon > 0$ and an infinite subsequence of successful iterates $\{x_k\}$, indexed by $\{t_i\} \subseteq \mathbf{S} = \{\, k \geq 0 \mid \hat{\rho}_k \geq \mu_1 \}$, such that

$$\|g_{t_i}\| \geq 2\varepsilon > 0 \tag{23}$$

for some $\varepsilon > 0$ and all $i \in \mathbb{N}$. Theorem 7 guarantees that, for each i, there exists the first successful iterate $l(t_i) > t_i$ such that $\|g_{l(t_i)}\| < \varepsilon$. Defining $l_i = l(t_i)$, there exists another subsequence in $\mathbf{S}$, indexed by $\{l_i\}$, such that

$$\|g_k\| \geq \varepsilon \;\text{ for }\; t_i \leq k < l_i \;\text{ and }\; \|g_{l_i}\| < \varepsilon. \tag{24}$$

The full attention is focused on the sequence of successful iterates whose indices are in the set

$$\mathscr{K} = \{k \in \mathbf{S} \mid t_i \leq k < l_i, \;\text{ for all } i \in \mathbb{N}\}.$$

From (H3) and (24), for an arbitrary $k \in \mathscr{K}$, we can write

$$m_k(x_k) - m_k(x_k + d_k) \geq \kappa_{\text{mdc}} \, \|g_k\| \, \min\left[\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right] \geq \varepsilon \kappa_{\text{mdc}} \min\left[\Delta_k, \frac{\varepsilon}{\kappa_{\text{umh}}}\right]. \tag{25}$$

Because $k \in \mathscr{K}$, it is clear that

$$C_k - f_{k+1} \geq \mu_1 [m_k(x_k) - m_k(x_k + d_k)] \geq \varepsilon \kappa_{\text{mdc}} \min\left[\Delta_k, \frac{\varepsilon}{\kappa_{\text{umh}}}\right].$$

This fact along with Lemma 6 imply that

$$\lim_{k \in \mathscr{K}, k \to \infty} \Delta_k = 0. \tag{26}$$

Therefore, from (25) and (26), it can conclude

$$m_k(x_k) - m_k(x_k + d_k) \geq \kappa_{\text{mdc}} \, \varepsilon \, \Delta_k.$$

Now, this inequality and Lemma 1 lead to

$$\left| \frac{f_k - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)} - 1 \right| \leq \frac{\kappa_{\text{ubh}}}{\kappa_{\text{mdc}} \, \varepsilon} \, \Delta_k \to 0, \quad \text{as } \; k \to \infty \;\text{ and }\; k \in \mathscr{K}.$$

So, for sufficiently large $k \in \mathscr{K}$, we have

$$f_k - f_{k+1} \geq \mu_1 [m_k(x_k) - m_k(x_k + d_k)].$$

This along with (25), (H2) and (H3) imply

$$f_k - f_{k+1} \geq \mu_1 \kappa_{\text{mdc}} \, \|g_k\| \, \min\left[\Delta_k, \frac{\|g_k\|}{\|B_k\|}\right] \geq \mu_1 \varepsilon \kappa_{\text{mdc}} \min\left[\Delta_k, \frac{\varepsilon}{\kappa_{\text{umh}}}\right] \geq \mu_1 \varepsilon \kappa_{\text{mdc}} \, \Delta_k. \tag{27}$$

As a result of (27), for sufficiently large $k \in \mathscr{K}$, it can be easily concluded that

$$\Delta_k \leq \frac{1}{\mu_1 \varepsilon \kappa_{\text{mdc}}} (f_k - f_{k+1}).$$

Based on this bound and Lemma 2, for sufficiently large i, it follows that

$$\|x_{t_i} - x_{l_i}\| \leq \sum_{j \in \kappa, j = t_i}^{l_i - 1} \|x_j - x_{j+1}\| \leq \sum_{j \in \kappa, j = t_i}^{l_i - 1} \Delta_j \leq \frac{1}{\mu_1 \varepsilon \kappa_{\text{mdc}}} (f_{t_i} - f_{l_i}) \tag{28}$$

Now, Lemma 6 and (28) suggest that the right hand side of this inequality must tend to zero. Therefore, it is obtained that $\|x_{t_i} - x_{l_i}\|$ converges to zero, as i tends to infinity. From the continuity of the gradient, it can be deduced

$$\lim_{i\to\infty} \|g_{t_i} - g_{l_i}\| = 0. \tag{29}$$

On the basis of the definitions of $\{t_i\}$ and $\{l_i\}$, this is impossible implying that $\|g_{t_i} - g_{l_i}\| \geq \varepsilon$. Hence, there is no subsequence satisfying (23) and the proof is completed. □

Theorem 3. *Suppose that all assumptions of Theorem 8 hold, then there is no limit point of the sequence $\{x_k\}$ being a local maximum of $f(x)$.*

Proof. The proof is similar to that of the theorem in [12]. The details are omitted. □

In what follows, there is an intention to explore conditions guaranteeing the convergence to second-order stationary points, that is, points x_* at which the second order necessary optimality conditions

$$g(x_*) = 0 \;\text{ and }\; G(x_*) \;\text{ is positive semidefinite}$$

are satisfied. In order to investigate the global convergence to second-order stationary points, similar to [4, 8], the following additional assumptions is needed:

(H4) The model is asymptotically second-order coherent with the objective function close to first-order critical points, i.e.

$$\lim_{k\to\infty} \|G_k - B_k\| = 0, \;\text{ whenever }\; \lim_{k\to\infty} \|g_k\| = 0.$$

(H5) If the smallest eigenvalue τ_k of B_k at x_k is negative, it may be determined a direction d_k provided that

$$m_k(x_k) - m_k(x_k + d_k) \geq \kappa_{\text{sod}}|\tau_k| \min[\tau_k^2, \Delta_k^2],$$

for some constant $\kappa_{\text{sod}} \in (0, \frac{1}{2})$.

Considering some assumptions, the following lemma implies that if the steps tend to zero, the trust-region radius Δ_k is bounded away from zero, for k sufficiently large .

Lemma 6. *Suppose that (H1), (H2) and (H4) hold and $f(x)$ be a twice continuously differentiable function. Suppose furthermore that there exists a subsequence $\{x_{k_i}\}$ and a constant κ_{qmd} such that*

$$m_k(x_k) - m_k(x_k + d_{k_i}) \geq \kappa_{qmd}\|d_{k_i}\| > 0,$$

for all i sufficiently large. Finally, suppose that

$$\lim_{i\to\infty} \|d_{k_i}\| = 0.$$

Then we have that the iterate x_{k_i} is very successful and so $\Delta_{k_i+1} \geq \Delta_{k_i}$, for i sufficiently large.

Proof. For Algorithm 2, similar to Lemma 6.5.3 of Conn et al. in [8], we can prove that $\rho_{k_i} > \mu_2$. Now, by using this fact, dividing the proof into the two cases like Lemma 4 and employing Lemma 2, one can conclude that the current iterate is very successful, for i sufficiently large. Therefore, Step 3 of the algorithm indicates the results. □

Theorem 4. *Suppose that (H1)–(H4) hold and that $\{x_{k_i}\}$ is a subsequence of the iterates generated by Algorithm 2 converging to a first-order critical point x_* where the Hessian of the objective function $G(x_*)$ is positive definite. Consider also that $s_k \neq 0$ for all k sufficiently large. Then the complete sequence of iterates converges to x_*, all iterates are eventually very successful, and the trust-region radius Δ_k is bounded away from zero.*

Proof. In the case of Algorithm 2, the proof is held by following Theorem 6.5.5 of Conn et al. in [8] and regarding very successful iterates in Lemma 10. □

Theorem 5. *Suppose that (H1)–(H5) hold and all iterates lie within a close, bounded domain Ψ. Then there exists at least one limit point x_* of the sequence $\{x_k\}$ produced by Algorithm 2, which is a second-order stationary point.*

Proof. See Theorem 6.6.5 of Conn et al. in [8]. The details are omitted. □

Theorem 6. *Suppose that (H1)–(H5) hold and let x_* be any limit point of the sequence of iterates $\{x_k\}$. Then x_* is a second-order stationary point.*

Proof. The conclusion can be obtained similar to Theorem 6.6.8 of Conn et al. in [8]. □

Here, similar to the theorem from Grippo et al. in [12], one can conclude that there is no limit point of the sequence $\{x_k\}$ being a local maximum of $f(x)$. Furthermore, superlinear and quadratic convergence rates of the proposed algorithms can be established like what is described in [2].

Remark 1. *As we mentioned, the results of this section have established for NTRM-2, but this results can be satisfied for NTRM-1 similar to that cases concern with ($\texttt{Flag} < \mathscr{I}$) and also for NTRG-1 and NTRG-2 by just attending to this fact that $\{f_{l(k)}\}$ is decreasing sequence and substituting lemma 6 with Lemma 7 from Ahookhosh and Amini in [1].*

4 Numerical experiments

In this section, we evaluate the efficacy and robustness of the paper's fundamental idea by reporting the computational outcomes of the proposed approach. In particular, we provide numerical results for the traditional trust-region, TTR, and the traditional nonmonotone trust-region method using the nonmonotone term (5), NTRG, as well as its variations resulting from Algorithms 1 and 2, NTRG-1 and NTRG-2. Similarly, we exploit the nonmonotone term [17], NTRM, and our modified variants of this method, NTRM-1, and NTRM-2, for the nonmonotone trust-region algorithm of Mo et al. Almost 90 standard test functions from the CUTEst libraries [5], and [10] are utilized to evaluate the effectiveness of these techniques. The dimensions of examination tasks range from 2 to 5625. All problems are unconstrained, with some of them being highly nonlinear. These problems have proved to be reasonably hard in the past. Table 2 displays their names and sizes.

All presented codes are written in MATLAB 8.01 programming environment in double precision arithmetic format on an ASUS laptop (2.20 GHz CPU ,8GB RAM) with a Linux operation system . We employ the Steihaug-Toint procedure, see [8], to solve trust-region subproblems. As denoted by Bastin et al. in [4], the Steihaug-Toint algorithm terminates at $x_k + d$ whenever

$$\|\nabla f(x_k+d)\| \leq \min\left[\frac{1}{10}, \|g_k\|^{\frac{1}{2}}\right]\|g_k\| \quad \text{or} \quad \|d\| = \Delta_k,$$

as well holds. Furthermore, we stop the implementations whenever the total number of iterates exceeds 20000, or the condition $\|g_k\| \leq 10^{-5}$ satisfies. We exploit $N = 10$ for NTRG, NTRG-1, NTRG-2, and $\eta_0 = 0.85$ for NTRM, NTRM-1, and NTRM-2 as considered in [17, 23]. Meanwhile, in all algorithms, $\mu_1 = 0.05$ and $\mu_2 = 0.9$ and the initial trust-region radius, the same as the literature [7], set to $\Delta_0 = \frac{1}{10}\|g_0\|$. We also update the trust-region radius using

$$\Delta_{k+1} = \begin{cases} \max\left[\Delta_k, \gamma_2\|d_k\|\right], & r_k \geq \mu_2, \\ \Delta_k & \mu_1 \leq r_k < \mu_2, \\ \gamma_1\|d_k\|, & r_k < \mu_1. \end{cases}$$

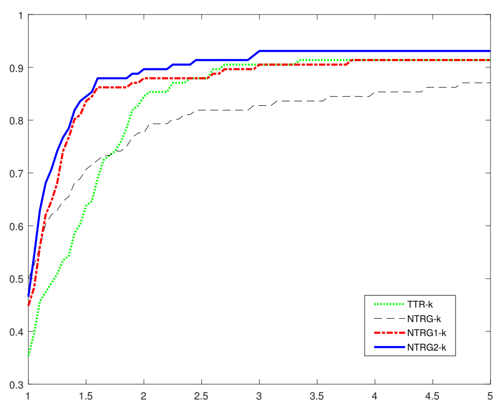
Regarded to the applied algorithm, we set r_k equal to ρ_k or $\hat{\rho}_k$. We select the parameters $\gamma_1 = 0.25$ and $\gamma_2 = 3$ similar to the literature [11]. As known, based on the results provided in [11], three categories of parameters have been analyzed: "worst," "standard," and "best" cases. The general conclusions reveal that the worst-case value for γ_2 is 5, which causes increasing the CPU time and the number of iterations in most of the test problems. The worst-case value of γ_2 provides a suitable measure of how large the trust region should be in each iteration to avoid the algorithm's poor performance in terms of CPU time and number of iterations. Algorithm 2 controls this issue, and the only difference between it and Algorithm 1 in Step 3 is whether the radius remains unchanged or expands without going over the trust region's maximum radius. Besides, the presented algorithms employ the well-known BFGS formula to update the matrix B_k as an approximation of the exact Hessian matrix by the following formula

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

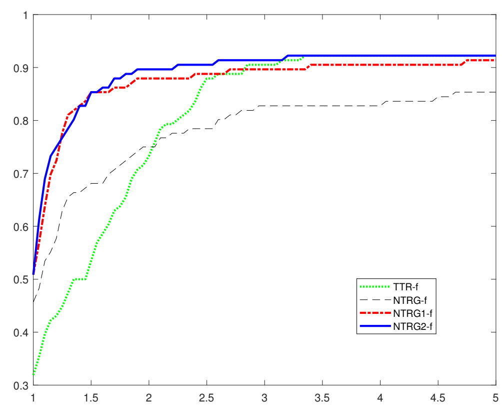
where $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$. It has been mentioned that the considered algorithms do not update B_k whenever the curvature condition $s_k^T y_k > 0$ does not hold. Additionally, while the algorithms are running, we ensure that all the codes reach the same optimal points. As a result, we present those results in which all algorithms reach the same places. The results are summarized in Table 2. In 2002, Dolan and Moré in [9] introduced an exciting technique that used a statistical process to compare iterative algorithms by illustrating the results in performance profiles. This procedure selects a performance index to compare the algorithms under consideration, and it graphically depicts the results using performance profiles to make a more precise comparison. Here, we use this technique to compare the presented algorithms in terms of the number of iterations and the number of function evaluations separately. performance profiles in Figure 1 show the results of these comparisons. From Figure 1(a), first, we notice that when the performance measure is the total number of iterates, NTRG-1 and NTRG-2 are so competitive in the sense of the most wins. The most wins are related to NTRG and NTRG-2, with about 50% of all test functions. Meanwhile, NTRG-2 solves approximately 93% of all test problems. The diagram of NTRG-1 and NTRG-2 grow up faster than TTR and NTRG, which certify the effectiveness of these two algorithms. We observe that the results of NTRG-1 and NTRG-2 are remarkably better than TTR and NTRG. On the other hand, From Figure 1(b), the total number of function evaluations for NTRG-1 and NTRG-2 are considerably less than for TTR, NTRG, and the most wins are approximately 51% of all tests functions. While from Figure 1(c), NTRM, NTRM-1, and NTRM-2 compete in the number of iterations, and 1(d) show that NTRM-1 and NTRM-2 outperform NTRM regarding the number of function evaluations. Based on our detailed examination, we conclude that the proposed frameworks for addressing unconstrained optimization problems are more efficient and robust than traditional ones.

5 Conclusion

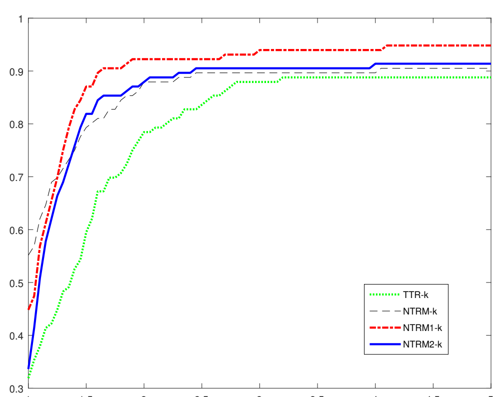
Nonmonotonic trust-region algorithms provide a promising iterative method for solving optimization problems. The theoretical convergence properties of trust-region methods are robust and numerically reliable. Nonmonotone strategies, on the other hand, can increase the efficacy and robustness of methods, particularly when dealing with highly nonlinear problems and narrow valleys . Combining these strategies provides strong and efficient optimization approaches. The standard nonmonotone trust-region framework accepts new points and updates the trust-region radius using a nonmonotone ratio. Although this method is effective for highly nonlinear problems, it may significantly enlarge the trust region. This procedure increases the total number of subproblems and function evaluations. The authors claim that employing a nonmonotone ratio for accepting new points, as well as the classic monotone ratio or its hybrid with nonmonotone ratios, can reduce the total cost of computing for solving optimization problems.



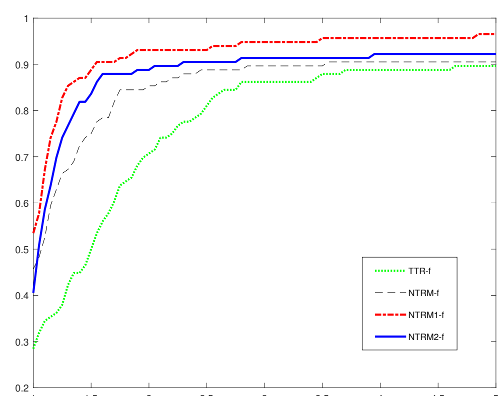
(a) Iterates performance profile



(b) Function evaluations performance profile



(c) Iterates performance profile



(d) Function evaluations performance profile

Figure 1: Iterates and Function evaluations performance profiles for the considered algorithms

Table 2. Numerical results

Problem name	Dim	TTR		NTRG		NTRG-1		NTRG-2		NTRM		NTRM-1		NTRM-2	
		N_i	N_f	N_i	N_f	N_i	N_f	N_i	N_f	N_i	N_f	N_i	N_f	N_i	N_f
BEALE	2	18	23	15	18	16	19	16	19	18	24	19	23	19	23
BRKMCC	2	7	10	7	10	7	10	7	10	7	10	7	10	7	10
BROWNBS	2	72	81	30	34	78	79	32	33	29	31	78	79	32	33
CLIFF	2	50	58	50	58	50	58	50	58	50	58	50	58	50	58
CUBE	2	63	95	62	75	41	47	41	47	63	74	41	47	33	43
DENSCHNA	2	8	9	8	9	8	9	8	9	8	9	8	9	8	9
DENSCHNB	2	8	9	8	9	8	9	8	9	8	9	8	9	8	9
DENSCHNF	2	11	18	10	17	14	19	14	19	10	17	14	19	14	19
DJTL	2	faild		260	486	251	444	288	520	257	479	251	444	286	516
EXPFIT	2	21	32	14	21	17	22	17	22	19	26	22	29	22	29
HAIRY	2	89	125	124	160	56	62	56	62	64	63	46	53	43	50
HILBERTA	2	6	7	6	7	6	7	6	7	6	7	6	7	6	7
HIMMELBB	2	30	40	30	40	30	40	30	40	30	40	30	40	30	40
HIMMELBG	2	16	23	21	25	12	14	12	14	16	24	17	20	17	20
HIMMELBH	2	9	12	8	9	10	11	10	11	8	9	10	11	10	11
HUMPS	2	374	520	215	258	1201	1223	645	662	3141	3204	293	305	9686	9696
LOGHAIRY	2	2064	2738	221	281	1835	1880	163	173	105	141	619	649	120	135
MARATOSB	2	4922	6810	2522	2937	2191	2390	3037	3522	1845	1968	2641	2978	2665	3051
ROSENBR	2	65	103	47	103	35	40	39	46	47	61	35	40	39	46
SINEVAL	2	161	227	96	123	89	101	129	168	95	117	119	146	136	181
SISSER	2	21	22	21	22	21	22	21	22	21	22	21	22	21	22
SNAIL	2	167	239	156	202	112	120	175	213	135	174	113	120	137	161
ZANGWIL2	2	3	4	3	4	3	4	3	4	3	4	3	4	3	4
BARD	3	29	38	23	26	23	26	23	26	23	26	23	26	23	26
BOX2	3	17	18	17	18	17	18	17	18	17	18	17	18	17	18
BOX3	3	17	18	17	18	17	18	17	18	17	18	17	18	17	18
DENSCHNE	3	29	36	32	37	31	36	31	36	32	37	31	36	31	36
ENGVAL2	3	36	60	29	38	26	32	26	32	23	32	25	34	30	40
GULF	3	46	54	39	43	42	46	43	47	39	43	42	46	43	47
HATFLDD	3	73	96	37	40	48	53	40	44	40	43	37	38	46	49
HATFLDE	3	36	44	53	63	26	29	26	29	26	29	26	29	26	29
HATFLDFL	3	6	9	6	9	6	9	6	9	6	9	6	9	6	9
HELIX	3	35	49	22	27	24	29	24	29	43	50	24	29	24	29
YFITU	3	178	218	109	120	143	154	146	159	138	156	127	138	147	162
ALLINITU	4	22	37	21	28	24	28	24	28	14	18	19	22	19	22
BROWNDEN	4	faild		43	65	40	52	41	53	37	57	39	50	42	58
HIMMELBF	4	42	50	43	53	39	45	39	45	38	48	42	50	42	50
KOWOSB	4	32	35	32	35	31	32	31	32	31	34	31	32	31	32
OSBORNEA	5	227	306	105	121	89	98	124	145	81	93	89	98	71	83
BIGGS6	6	41	42	211	256	41	42	41	42	40	41	41	42	41	42
HEART6LS	6	faild		1684	1798	907	961	2020	2115	1134	1275	986	1067	1263	1393
PALMER5C	6	36	58	23	28	28	32	28	32	23	31	28	33	28	33
PALMER1D	7	faild		34	47	48	61	43	59	34	47	48	61	35	48
AIRCRFTB	8	31	52	38	52	36	45	34	43	38	50	30	40	30	40
PALMER1C	8	57	80	39	54	54	69	39	54	39	54	54	69	39	54
PALMER2C	8	98	142	44	57	53	66	44	57	44	57	53	66	44	57

Table 2. Numerical results (*continued*)

PALMER3C	8	71	102	38	49	48	59	46	59	38	49	48	59	40	51
PALMER4C	8	59	83	38	49	50	61	45	56	38	49	50	61	44	55
PALMER6C	8	57	72	45	54	50	59	45	54	45	54	50	59	45	54
PALMER7C	8	68	87	40	51	52	63	42	53	40	51	52	63	42	53
PALMER8C	8	49	66	46	57	43	54	46	57	46	57	43	54	46	57
HILBERTB	10	13	22	13	14	15	16	15	16	13	14	15	16	15	16
OSCIPATH	10	31	60	69	101	104	115	104	115	77	118	53	70	98	118
OSBORNEB	11	117	172	73	82	71	76	71	76	71	85	70	75	71	76
WATSON	12	68	101	49	60	56	61	58	63	46	53	47	54	47	54
DIXMAANK	15	1160	1171	1547	1547	350	354	350	354	221	252	330	335	290	295
ERRINROS	50	811	1081	434	588	440	456	453	478	434	601	469	492	482	526
TOINTGOR	50	221	315	115	156	127	130	127	130	116	160	137	142	137	142
TOINTPSP	50	126	187	144	193	94	102	92	100	109	144	103	111	93	101
TOINTQOR	50	65	105	89	107	97	100	97	100	86	109	97	100	97	100
VAREIGVL	50	55	66	133	175	108	109	108	109	91	115	89	92	89	92
SENSORS	100	faild		462	596	250	254	250	254	337	480	233	242	233	242
MANCINO	100	34	83	609	868	630	771	551	677	512	750	178	230	178	230
ARGLINA	200	3	4	3	4	3	4	3	4	3	4	3	4	3	4
BOX	200	57	87	28	42	39	50	30	41	26	43	34	48	37	54
BROWNAL	200	18	33	7	16	9	18	7	16	7	16	9	18	7	16
VARDIM	200	41	67	41	67	41	67	41	67	41	67	41	67	41	67
EG2	1000	5	10	5	10	5	10	5	10	5	10	5	10	5	10
PENALTY1	1000	67	96	103	140	71	95	97	129	faild		92	110	100	118
MSQRTBLS	1024	10080	12628	5357	7590	6807	6861	6869	6966	6367	9375	7027	7146	6853	7040
EDENSCH	2000	faild		770	1126	216	221	218	225	322	477	216	221	218	225
EIGENALS	2550	3380	3890	5531	5532	4948	4979	4324	4395	2533	2963	3290	3341	3467	3585
DIXMAANA	3000	11	14	10	11	11	12	11	12	10	11	11	12	11	12
DIXMAANB	3000	16	18	30	37	43	48	47	57	16	18	16	18	16	18
DIXMAANC	3000	27	39	85	109	37	47	42	52	34	42	36	43	36	43
DIXMAAND	3000	18	23	116	142	28	34	28	34	30	38	28	34	28	34
DIXMAANE	3000	244	245	244	245	244	245	244	245	244	245	244	245	244	245
DIXMAANF	3000	318	383	296	302	235	236	235	236	358	367	235	236	235	236
DIXMAANG	3000	232	248	244	263	213	217	213	217	278	289	365	373	367	375
DIXMAANH	3000	190	196	380	404	189	193	189	193	202	214	189	193	189	193
DIXMAANJ	3000	345	348	227	228	1868	1869	1868	1869	227	228	1868	1869	1868	1869
DIXMAANL	3000	557	566	1307	1322	1056	1052	1056	1052	1344	1361	1052	1056	1052	1056
BROYDN7D	5000	faild		12365	12756	13800	13816	13800	13816	338	511	250	266	204	228
BRYBND	5000	84	145	834	1191	250	266	204	228	338	511	250	266	204	228
DQDRTIC	5000	54	79	20	28	26	31	27	33	23	31	27	34	25	32
ENGVAL1	5000	faild		941	1362	143	147	194	200	285	415	143	147	143	147
NONCVXU2	5000	faild		12246	13715	18650	18653	18650	18653	13205	13759	17809	17839	18686	18716
NONDQUAR	5000	1081	1100	1165	1176	1541	1549	1057	1070	1165	1176	36	44	1057	1070
SINQUAD	5000	faild		39	52	41	45	26	30	30	41	41	45	26	30
TQUARTIC	5000	51	62	29	34	110	115	29	34	29	34	53	58	29	34
FMINSRF2	5625	495	515	529	548	495	496	485	487	489	486	213	214	489	492
FMINSURF	5625	392	429	372	373	379	380	372	373	376	372	276	277	372	373
NLMSURF	5625	868	953	474	475	560	561	474	475	474	475	344	346	474	475

References

[1] M. Ahookhosh, K. Amini, *An efficient nonmonotone trust-region method for unconstrained optimization*, Numer. Algorithms. **59** (2012) 523–540.

[2] M. Ahookhosh, K. Amini, *A nonmonotone trust region method with adaptive radius for unconstrained optimization*, Comput. Math. Appl. **60**(2010) 411–422.

[3] M. Ahookhosh, K. Amini, M.R. Peyghami, *A nonmonotone trust-region line search method for large-scale unconstrained optimization*, Appl. Math. Model. **36** (2012) 478–487.

[4] F. Bastin, V. Malmedy, M. Mouffe, Ph.L. Toint , D. Tomanos, *A retrospective trust-region method for unconstrained optimization*, Math. Program. **123** (2010) 395–418.

[5] I. Bongartz, A.R. Conn, N.I.M. Gould, Ph.L. Toint, *CUTE: constrained and unconstrained testing environments*, ACM Trans. Math. Software **21** (1995) 123–160.

[6] R.M. Chamberlain, M.J.D. Powell, C. Lemarechal, H.C. Pedersen, *The watchdog technique for forcing convergence in algorithms for constrained optimization*, Math. Program. **16** (1982) 1–17.

[7] A.R. Conn, N.I.M. Gould, Ph. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, (Springer Verlag, Series on Computational Mathematics **(17)**, Berlin, 1992.

[8] A.R. Conn, N.I.M. Gould, Ph.L. Toint, *Trust-Region Methods*, SIAM, Philadelphia PA, 2000.

[9] E. Dolan, J.J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program. **91** (2002) 201–213.

[10] N.I.M. Gould, D. Orban, Ph.L. Toint, *CUTEr: A constrained and unconstrained testing environment, revisited*, ACM Trans. Math. Softw. **29** (2003) 373–394.

[11] N.I.M Gould, D. Orban, A. Sartenaer, Ph.L. Toint, *Sentesivity of trust-region algorithms to their parameters*, 4OR **3** (2005) 227–241.

[12] L. Grippo, F. Lampariello, S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal. **23** (1986) 707–716.

[13] L. Grippo, F. Lampariello, S. Lucidi, *A truncated Newton method with nonmonotone line search for unconstrained optimization*, J. Optim. Theory. Appl. **60** (1989) 401–419.

[14] L. Grippo, F. Lampariello, S. Lucidi, *A class of nonmonotone stabilization methods in unconstrained optimization*, Numer. Math. **59** (1991) 779–805.

[15] K. Levenberg, *A method for the solution of certain non-linear problems in least squares*, Q. Appl. Math. **2** (1944) 164–166.

[16] D.W. Marquardt, *An algorithm for least-squares estimation of nonlinear parameters*, SIAM J. Appl. Math. **11** (1963) 431–441.

[17] J. Mo, C. Liu, S. Yan, *A nonmonotone trust region method based on nonincreasing technique of weighted average of the succesive function value*, J. Comput. Appl. Math. **209** (2007) 97–108.

[18] J.J. Moré, B.S. Garbow, K.E. Hillstrom, *Testing unconstrained optimization software*, ACM Trans. Math. Softw. **7** (1981) 17–41.

[19] J. Nocedal, Y. Yuan, *Combining trust region and line search techniques*, in: Y. Yuan(Ed.), Advanced in Nonliear Programming (Kluwer), (1998) 153–175.

[20] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer, New York, 2006.

[21] M.J.D. Powell, *A new algorithm for unconstrained optimization*, In: J.B. Rosen, O.L. Mangassarian and K. Ritter (eds.), Nonlinear Programming, Academic Press, New York (1970) 31–66.

[22] Ph.L. Toint, *An assessment of nonmonotone linesearch technique for unconstrained optimization*, SIAM J. Sci. Comput. **17** (1996) 725–739.

[23] H.C. Zhang, W.W. Hager, *A nonmonotone line search technique for unconstrained optimization*, SIAM J. Optim. **14** (2004) 1043–1056.