Mathematics Interdisciplinary Research **7** (2022) 45 – 60

Original Scientific Paper

An Improved Trust Region Method Equipped by Filter Technique

Zeinab Saeidian * *and Farshid Arzani*

Abstract

Using a filter technique, a new efficient nonmonotone trust region method is proposed. The proposed scheme is based on updating the approximation of the Hessian matrix with the scaled memoryless BFGS update formula. To update the trust region radius, an appropriate adaptive scheme is used. Moreover, a proper nonmonotone procedure is applied. Assuming some suitable assumptions, the global convergence is obtained. Numerical results are reported to show the efficiency of the offered approach.

Keywords: nonmonotone adaptive trust region methods, scaled memoryless BFGS update, filter technique, global convergence.

2020 Mathematics Subject Classification: 65K05, 90C30, 90C06.

How to cite this article
Z. Saeidian and F. Arzani, An improved trust region method equipped by filter technique, *Math. Interdisc. Res.* **7** (2022) 45 – 60.

1. Introduction

In this paper, we deal with the following optimization problem

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice continuously differentiable function. Two efficient classes methods for solving such unconstrained optimization problems are trust region and line search approaches. There has been a surge of articles on trust

*Corresponding author (E-mail: saeidian@kashanu.ac.ir)
Academic Editor: Abbas Saadatmandi
Received 5 January 2022, Accepted 27 February 2022
DOI: 10.22052/mir.2022.243421.1328

region methods over the past four decades. The reader can refer to [7, 22] for a review on trust region methods. At the kth iteration of standard trust region method, the trial step d_k, is the solution of the following subproblem:

$$\min q_k(d) = f_k + g_k^T d + \frac{1}{2} d^T B_k d$$
$$s.t.\|d\| \leq \Delta_k, \tag{2}$$

where $f_k = f(x_k)$, $g_k = \nabla f(x_k)$, $B_k = B(x_k)$ is a $n \times n$ symmetric matrix which is the exact Hessian, i.e. $\nabla^2 f(x_k)$, or its approximation, Δ_k is a positive parameter that is called trust region radius, and $\|.\|$ denotes the Euclidean norm.

Given a trial step d_k, we can evaluate the consistency of the model and the original function by the trust region ratio:

$$\rho_k = \frac{f(x_k) - f(x_k + d_k)}{q_k(0) - q_k(d_k)}.$$

Then, we can decide that at the kth iteration, the trial step is accepted or rejected. If $\rho_k \geq \mu$, the new point $x_{k+1} = x_k + d_k$ is introduced and the trust region radius is expanded. Conversely, if $\rho_k < \mu$, the kth iteration is called unsuccessful and the trial step is rejected.

Two crucial factors in the efficieny of the algorithm are selecting the initial radius and the procedure of updating the trust region radius in each iteration [27]. The large amount of trust region radius may causes increas in the number of subproblems that should be solved. On the other hand, small trust region radius increases the number of iterations. So, it is important to manage the updating strategies of the radii properly.

So, many researchers studied these issues [9, 27, 28, 29, 32]. Sartnear [27] proposed a procedure to determine an initial radius. Gould et al. [13] suggested the practical scheme by choosing appropriate parameters for updating the trust region radius. The adaptive trust region methods have interesting results, such as Zhang et al. in [32] that proposed $\Delta_k = c^p \|g_k\| \|\hat{B}_k^{-1}\|$, where $c \in (0,1)$, $0 < p \in \mathbf{Z}$ and $\hat{B}_k = B_k + iI$ is a positive definite matrix for some $i \in \mathbb{N}$. Unfortunatly, becuase of calculating the inverse matrix $\hat{B}_k^{-1}$ at each iteration, this method cannot be applied on large scaled optimization problems. Therefore, Shi and Gou in [28] defined:

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

where q_k satisfies $-\frac{g_k^T q_k}{\|g_k\|\|q_k\|} \geq \tau$, $\tau \in (0,1)$, and c, p is defined from above. and $\tilde{B}_k = B_k + iI$ where i is the smallest nonnegative integer such that $q_k^T \tilde{B}_k q_k > 0$. Recently, Kamandi et al. [17] introduced another adaptive rule. Their proposed scheme used the information of the model based on the current and previous iterations. So, the trust region radius is automatically adjusted such as follows:

$$q_k = \begin{cases} -g_k, & \text{if } k=0 \quad or \quad \frac{-(g_k^T d_{k-1})}{\|g_k\|\|d_{k-1}\|} \le \tau; \\ d_{k-1}, & \text{Otherwise,} \end{cases}$$

and

$$s_k = \begin{cases} -\frac{g_k^T q_k}{q_k^T B_k q_k}\|q_k\|, & \text{if } k=0; \\ \max\{-\frac{g_k^T q_k}{q_k^T B_k q_k}\|q_k\|, \lambda\Delta_{k-1}\}, & \text{if } k \ge 1, \end{cases}$$

where, $\lambda > 1$, B_k is updated by a modified BFGS method,

$$\Delta_k := c^{p_k} \min\{s_k, \bar{\Delta}\}, c \in (0,1),$$

and $\bar{\Delta}$ is a real-valued nonnegative constant. Also, p_k is the smallest nonnegative integer for which $\rho_k \ge \mu$, where $\mu \in (0,1)$.

Later, since this method suffers from some disadvantages, Kamandi and Amini [18] offered another trust region scheme in such a way that the radius of trust region in the rejected steps is regulated using a radius dependent shrinkage parameter. As we know, often nonmonotone trust region algorithms have better convergence behavior in comparison with the monotone case. So, in order to achieve good computational experiments, the nonmonotone strategies have been proposed in the structure of the trust region method [15, 19]. At first, Chamberlain et al. suggested the watch-dog approach [6]. Then, Grippo et al. in [15] introduced a nonmonotone line search method for Newton's method. Grippo's method suffers from some disadvantages [1, 31]. Hence, Ahookhosh and Amini in [1] suggested an efficient nonmonotone term as follows:

$$R_k = \epsilon_k f_{\ell(k)} + (1-\epsilon_k) f_k, \tag{3}$$

where $f_k = f(x_k)$, $\epsilon_k \in [\epsilon_{\min}, \epsilon_{\max}] \subset [0,1]$ and $f_{\ell(k)}$ is the Grippo's nonmonotone term which is defined by

$$f_{\ell(k)} = \max_{0 \le j \le A(k)} f_{k-j}, \tag{4}$$

where $A(0) = 0$ and, for $k \ge 1$, $A(k) = \min\{k, A\}$, for given $A > 0$. Tarzanagh et al., [4] provided a new variant of nonmonotone parameter as $(1+\omega_k)R_k$ where R_k is defined by (3) and ω_k is defined as the following positive sequence:

$$\omega_k = \begin{cases} \eta_k, & if\ R_k > 0, \\ 0, & if\ R_k \le 0, \end{cases}$$

and it holds

$$\sum_{k=1}^{\infty} \omega_k \le \omega < \infty, \tag{5}$$

In line search methods, at kth iteration, the new point is achieved by:

$$x_{k+1} = x_k + \alpha_k d_k,$$

where, d_k and $\alpha_k > 0$ are a (descent) direction and the step length, respectively. One of the most practical step length is proposed by Barzilai and Borwein [5]. Suppose that $g_k = \nabla f(x_k)$. Also, consider $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. The new point in Barzilai-Borwein relation is computed by $x_{k+1} = x_k - U_k g_k$ in which,

$$U_k = \alpha_k I, \tag{6}$$

In order to satisfy the quasi Newton property by U_k, we compute α_k as the solution of $\min \|s_{k-1} - U_k y_{k-1}\|^2$ and also by symmetry, we can solve $\min \|U_k^{-1} s_{k-1} - y_{k-1}\|^2$ to compute α_k. So, the step size α_k is achieved from a two point approximation to the secant equation as the following relations:

$$\alpha_k^{BB1} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \quad \alpha_k^{BB2} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}. \tag{7}$$

For the first time, Filter technique has been proposed by Fletcher and Leyffer [12] based on multi objective optimization. In this manner, the points in the filter set is accepted when it is not dominated by any of the points inside the filter. When a point is accepted, all of the points dominated by the new point are deleted from the filter. Fatemi and Madavi-Amiri [10] introduced a filter that can control the size of the filter with special acceptance condition. Arzani and Peyghami [3] employed the finite filter in Barzilai-Borwein gradient method to suggest a modiyed version of Barzilai-Borwein algorithm using a relaxed nonmonotone line search techniques.

In this paper, we introduce an effective adaptive nonmonotone trust region method with a filter technique to solve the unconstrained optimization problems. Our approach uses a scaled memoryless BFGS relation for updating the approximation of the Hessian matrix at each iteration. We also develop a modified version of the adaptive rule in [26] for the proposed algorithm. The global convergence property is established under some mild assumptions. The new method is applied on some test problems and the results show the efficiency of the proposed approach.

The remainder of this paper is as follows: in Section 2, we introduce the structure of the new nonmonotone adaptive trust region method. Section 3 is discusses the convergence property, as well as the rate of convergence of the new algorithm. Finally, we apply the new algorithm on some test problems and we compare the results in Section 4.

2. The New Algorithm

In this section, we introduce the new method. This algorithm is divided in two parts, the inner and the outer loops. In the inner loop, using the trust region method, we want to find the trial step d_k. We then apply an efficient strategy for updating the matrix B_k to propose an improved trust region method in the outer loop. As we know, the BFGS formula is more efficient than some other quasi Newton approaches. But in the case of nonconvex functions, BFGS method cannot guarantee that the Hessian matrix B_k holds the positive definite condition. In order to solve this difficulty, scaled memoryless BFGS update formula was proposed as follows [2]

$$B_{k+1} = \theta_k I - \theta_k \frac{d_k d_k^T}{d_k^T d_k} + \frac{y_k y_k^T}{d_k^T y_k},$$
$$y_k = g_{k+1} - g_k, \quad \theta_k = \frac{d_k^T y_k}{\|d_k\|^2}.$$

The numerical experiments show that the algorithm proposed by [2] is not efficient. This is because in Formula (8) if $d_k^T y_k > 0$, then B_{k+1}, is a positive defenite matrix; otherwise, we set $B_{k+1} = B_k$. Overcoming this problem, Li and Fukushima [20] suggested a modified version of BFGS formula

$$B_{k+1} = B_k - \frac{B_k d_k d_k^T B_k}{d_k^T d_k} + \frac{y_k^*(y_k^*)^T}{d_k^T y_k^*}, \tag{8}$$
$$y_k^* = y_k + \|g_k\| \left(1 - \frac{d_k^T y_k}{\|d_k\|^2}\right) d_k.$$

Based on this discussion, Xue et al., [30] proposed the following relation to update the B_{k+1} matrix:

$$B_{k+1} = \begin{cases} \theta_k I - \theta_k \frac{d_k d_k^T}{d_k^T d_k} + \frac{y_k y_k^T}{d_k^T y_k}, & \text{if } d_k^T y_k > 0; \\ B_k - \frac{B_k d_k d_k^T B_k}{d_k^T d_k} + \frac{y_k^*(y_k^*)^T}{d_k^T y_k^*}, & \text{Otherwise.} \end{cases} \tag{9}$$

We here introduce the inner loop of our algorithm to achieve the set of points in each iteration. The nonmonotone ratio is calculated by [4]:

$$\rho_k = \frac{(1+\omega_k)R_k - f(x_k + d_k)}{Pred_k}, \tag{10}$$

where R_k is defined by (3) and $Pred_k = q_k(0) - q_k(d_k)$. The trial step d_k is accepted whenever $\rho_k \geq \mu$ where $\mu \in (0,1)$. The radius is updated by:

$$\Delta_k = \min\left\{\varpi_k \frac{\|g_k\|}{\|B_k\|}, \Delta_{\max}\right\},$$

where $\Delta_{\max} > 0$ is a threshold value for the radii and ϖ_{k+1} is updated by:

$$\varpi_{k+1} = \begin{cases} \sigma_0 \varpi_k, & \rho_k < \mu_1, \\ \varpi_k, & \mu_1 \leq \rho_k \leq \mu_2, \\ \min\{\sigma_1 \varpi_k, \varpi_{\max}\}, & \rho_k > \mu_2, \end{cases} \tag{11}$$

where $0 < \sigma_0 < 1 < \sigma_1$, $0 < \mu_1 < \mu_2 \leq 1$ and $\varpi_{\max} > 0$ are given numbers. The new point is defined by $x_{k+1} = x_k + d_k$ if $\rho_k \geq \mu$; otherwise, we set $x_{k+1} = x_k$. The inner loop algorithm is given as bellow:

The Inner Loop Algorithm.

Input $x_0 \in \mathbb{R}^n$, $0 < \mu < \mu_1 < \mu_2 \leq 1$, $0 < \sigma_0 < 1 < \sigma_1$, $0 < \epsilon_{\min} < \epsilon_{\max} < 1$, $\pi, A, \varpi_{\max}, \Delta_{\max} > 0$, $0 < \beta$ and $0 < \upsilon$.

Step 0 Set $k = 0$, $B_0 := B(x_0) = 1$, $g_0 = g(x_0)$, $\varpi_0 = 1$ and $\Delta_0 = \min\left\{\varpi_0 \frac{\|g_0\|}{\|B_0\|}, \Delta_{\max}\right\}$.

Step 1 If $\|g_k\| \leq \pi$, then Stop.

Step 2 Solve (approximately) the subproblem (2) to determine the trial step d_k and compute ρ_k using (10).

Step 3 If $\rho_k < \mu$, then set $\Delta_k = \sigma_0 \Delta_k$ and go to Step 2.

Step 4 Set $x_{k+1} = x_k + d_k$.

Step 5 Calculate B_{k+1} from (9).
If $\|B_{k+1}\| \leq \beta$, or $\|B_{k+1}\| \geq \frac{1}{\beta}$, then set $\|B_{k+1}\| = \upsilon$.

Step 6 Update ϖ_{k+1} from (11) and set $\Delta_{k+1} = \min\left\{\varpi_{k+1} \frac{\|g_{k+1}\|}{\|B_{k+1}\|}, \Delta_{\max}\right\}$. Set $k =: k+1$ and go to Step 1.

We assume that the search direction d_k is satisfied in the conditions proposed by [16] at kth iteration of the outer loop algorithm. That is, there exist c_1 and c_2, such that for all k we have:

$$\|d_k\| \leq c_1 \|g_k\|, \qquad -\frac{d_k^T g_k}{\|g_k\|} \geq c_2 \|g_k\|. \tag{12}$$

It is worth mentioning that the relations (12) hold for the Barzilai-Borwein direction (6) and $d_k = -g_k$. For given N, tentative points $y_k^i, i = 0, \ldots, N$ are computed as bellow:

$$\begin{cases} y_k^0 = x_k, \\ p_k^0 = d_k, \\ y_k^{i+1} = y_k^i + p_k^i, \qquad i = 1, \ldots, m, \end{cases}$$

where $m \leq N$ and p_k^i is obtained from the following relation:

$$p_k^i = -\frac{1}{\alpha_k}\nabla f(y_k^i), \qquad i = 1, 2, \ldots, m,$$

in which, considering $s_k = y_k^i - y_k^{i-1}$ and $y_k = \nabla f(y_k^i) - \nabla f(y_k^{i-1})$, α_k is achieved from (7). At first, the obtained points are considered. If these points do not justify the filter condition, then we check the nonmonotone method. When the point y_k^i is accepted by the filter condition, we set $x_{k+1} = y_k^i$ and we add this point to the filter. Also, if the nonmonotone condition is satisfied we consider $x_{k+1} = y_k^i$. This procedure is continued at most N times. Otherwise, when none of the above conditions are established, we execute a nonmonotone line search in the direction of d_k, and the new point is $x_{k+1} = x_k + \lambda_k d_k$.

The Adaptive Nonmonotone Trust Region with Filter Technique (Original Algorithm).

Step 0 Given $x_0 \in R^n$, $f_{sup} > 0$ and the integer number $N \geq 1$ and $A \geq 0$, $\sigma > 0$, $\rho_1 > 0$, $\rho_2 > 0$, $\epsilon > 0, 0 < \theta_1 < \theta_2$, $\delta > 0$ and $0 \leq \tau_1 < \tau_2 < \frac{1}{\sqrt{n}}$. Set $\mathcal{F}_0 = \emptyset$, $i = 0$ and $k = 0$.

Step 1 If $\|g_k\| \leq \epsilon$ then stop.

Step 2 If $i \leq N$, then compute the point y_k^i, using the inner loop algorithm. Otherwise, go to Step 4.

Step 3 If y_k^i is accepted by filter and $f(y_k^i) \leq f_{sup}$, then set $x_{k+1} = y_k^i$ and add y_k^i to the filter $\mathcal{F}_k$. Update the filter by $\mathcal{F}_{k+1}$ and set $k = k + 1$ and go to Step 1.
Else if

$$f(y_k^i) \leq (1 + \omega_k)R_k - \max_{1 \leq h \leq i}\{\sigma \left\|y_k^h - x_k\right\|\}, \tag{13}$$

then set $x_{k+1} = y_k^i$, $\mathcal{F}_{k+1} = \mathcal{F}_k$ and $k = k + 1$ and go to Step 1.
Otherwise, set $i = i + 1$ and go to Step 2.

Step 4 Imply the line search method in the direction of d_k and compute the step size λ_k. Set $x_{k+1} = x_k + \lambda_k d_k$, $k = k + 1$ and go to Step 1.

Remark 1. It is worth to mention that $\epsilon > \pi > 0$. Also, in the algorithm we use the nonmonotone line search scheme.

3. Convergence Analysis

First of all, we divide the set of iterations into two sets C and $\acute{C}$ as follows:

$$C = \{k | x_k \in \mathcal{F}_k\}.$$

The set $\acute{C}$ is the set of iterations that satisfy the relation (13) or are achieved from Step 4 of the original algorithm. It is clear that $C \cap \acute{C} = \emptyset$.

Also, the following assumptions are necessary to be considered:

A1 The objective function $f(x)$ is a continuously differentiable function on $\mathbb{R}^n$.

A2 The level set $\hat{L} = \{x \in \mathbb{R}^n | f(x) \leq e^{\omega} f(x_0)\}$ is a closed bounded set, in which ω, satisfiies (5).

A3 The generated sequence $\{x_k\}$ by the proposed algorithm, remains in a closed bounded set $\Omega \subset \mathbb{R}^n$.

In order to analyze the global convergence property, we should intend the behavior of the generated points $\{x_k\}$ and the sequence $\{y_k^i\}$ produced by the inner loop procedure. If the original Algorithm stops at a finite iteration such as k_0, then we get the stationary point. Otherwise, the infinite sequence of points are produced by the algorithm. So, for the cardinality of C and $\acute{C}$, we have $|C| = \infty$ or $\left|\acute{C}\right| = \infty$.

In the following lemma, what happens to the convergence after adding many iterates to the filter infinitely.

Lemma 2.1. *Let assumptions* $A1 - A3$ *hold and* $|C| = \infty$. *Then,* [3, 25]

$$\lim_{k \in C} \|g_{k+1}\| = 0.$$

Proof. [11, Lemma 5.4]. □

Lemma 2.2. *If* $|\acute{C}| = \infty$, *then for the infinite sequence* $\{x_k\}_{k \in \acute{C}}$ *there exists a limit point that*

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

Proof. In order to prove this lemma, we should consider some other notes that is similar to [3]. □

Theorem 2.3. *Let assumptions* **A1**, **A2** *and* **A3**. *Then, the original algorithm either stops at a stationary point of* (1) *or for the sequence* $\{x_k\}_{k \in \acute{C}_1}$, *we have* [1, 3]*:*

$$\liminf_{k \longrightarrow \infty} \|g_k\| = 0.$$

Proof. By contradiction, assume that the original algorithm does not stop at a stationary point. Assume that the sequence $\{x_k\}_{k\in\acute{C}_1}$, achieved from the original algorithm, considering the following items [3],

1 $\{x_k\} \subseteq L^0$.

2 If $\{x_k\}_{k\in\acute{C}_1}$ is an infinite sequence, then for a given $A > 0$(integer), $\acute{C}_1$ can be written as below:

$$\acute{C}_1 = \{LA + b \mid L \in \mathbb{N}\bigcup\{0\}, 0 \leq b \leq A-1\}.$$

3 For some $\sigma > 0$, we have

$$f_{k+1} \leq |f_0| \prod_{i=0}^{k} (1+\omega_i) - \max_{1\leq h\leq i_k} \{\sigma \left\|y_k^h - x_k\right\|\},$$

such that $i_k > 0$ is the largest integer number such that $x_{k+1} = y_k^{i_k}$.

4 Consider A from (4). Then

$$\forall k \in \acute{C}_1, \exists L > 0 \;\&\; 0 \leq b \leq A-1 : k = LA + b,$$

and

$$\begin{aligned} f_{k+1} &= f_{LA+b+1} \leq |f_0| \textstyle\prod_{i=0}^{LA+b}(1+\omega_i) \\ &- \textstyle\sum_{i=0}^{L} \min_{iA\leq j\leq (i+1)A-1} \max_{1\leq h\leq i_k} \{\sigma \left\|y_k^h - x_k\right\|\},\ L = 0,1,2,\ldots. \end{aligned}$$

The above items are discussed in detail in [3]. For every $k = LA + b \in \acute{C}_1$, we have:

$$\begin{aligned} &\sum_{i=0}^{L} \min_{iA\leq j\leq (i+1)A-1} \max_{1\leq h\leq i_k} \{\sigma \left\|y_k^h - x_k\right\|\} \\ &\leq |f_0| \prod_{i=0}^{LA+b-1} (1+\omega_i) - f_{LA+b} \\ &\leq e^{\eta}|f_0| - f_{LA+b}. \end{aligned}$$

To summarize, we set

$$\tilde{\omega}_i = \min_{iA\leq j\leq (i+1)A-1} \max_{1\leq h\leq i_k} \{\sigma \left\|y_k^h - x_k\right\|\}.$$

Then, from assumptions **A1** and **A2** we have:

$$\begin{aligned} &\lim_{L\longrightarrow\infty} \sum_{i=0}^{L} \tilde{\omega}_i < \infty \\ &\Longrightarrow \lim_{i\longrightarrow\infty} \tilde{\omega}_i = 0. \end{aligned}$$

Based on the description of $\tilde{\omega}_i$,

$$\exists 0 \leq \tilde{b} \leq A-1 : \tilde{\omega}_i = \omega_{iA+\tilde{b}}.$$

Suppose that $\beta_i = iA + \tilde{b}$, then

$$\lim_{i \longrightarrow \infty} \omega_{\beta_i} = 0.$$

From the definition of ω_i, we have:

$$0 \leq \sigma \left\| y_{\beta_i}^h - x_{\beta_i} \right\| \leq \omega_{\beta_i}, \qquad \forall 0 \leq h \leq i_{\beta_i}.$$

So, as $i \longrightarrow \infty$, $\omega_{\beta_i} \longrightarrow 0$ can be concluded that

$$\left\| y_{\beta_i}^h - x_{\beta_i} \right\| \longrightarrow 0, \qquad \forall 0 \leq h \leq i_{\beta_i},$$

in which $\sigma > 0$ is a constant. Specially, if $h = 1$ we have $\|d_{\beta_i}\| = \left\| y_{\beta_i}^1 - x_{\beta_i} \right\| \longrightarrow 0$. Since d_{β_i} fulfills (12), $\|d_{\beta_i}\| \longrightarrow 0$ results in

$$\lim_{i \longrightarrow \infty} \| g_{\beta_i} \| = 0.$$

It implies that $\{x_{\beta_i}\}_{\beta_i \in \acute{C}_1}$ of $\{x_k\}_{k \in \acute{C}_1}$ converges to a stationary point of f. Thus,

$$\liminf_{k \to \infty, k \in \acute{C}_1} \|\nabla f(x_k)\| = 0.$$

□

Considering the above disscutions, one can achieve the following theorem:

Theorem 2.4. *Let assumptions* **A1**, **A2** *and* **A3**. *Then, the original algorithm either stops at a stationary point of* (1) *or*

$$\liminf_{k \longrightarrow \infty} \|g_k\| = 0.$$

4. Numerical Experiments

Now, we compare the performance profile of the new algorithm, FTRB algorithm, with the other algorithms as bellow:

AINART Algorithm 1.1 in [30].

TRFBB Algorithm 2.1 in [24].

FATRA Algorithm 3.1 in [25].

MATLAB software has been used for implementation of all the considered algorithms (MATLAB v9.9.0 R2020b environment on a PC with CPU Intel core i5 8500, 3.00 GHz and 16GB RAM memory and double precision format). The test problems are those selected from CUTEr collection [14]. The following parameters are considered in considered algorithms:

$$\mu = 0.1, \mu_1 = 0.25, \mu_2 = 0.75, \epsilon_{\min} = 10^{-6}, \epsilon_{\max} = 10^{+6}.$$

For all the considered algorithms, stop condition is $\|g_k\| \leq 10^{-6}$. The algorithm is failed when the number of iterations passes 10000 and the number of function evaluations exceed 50000. In Table 1, we named the test problems and their dimensions. It is worth to mention that, we applied the benefit of Dolan and Moré's perfomance profile [8].

Table 1. Test problems and dimensions [14].

Problem	n	Problem	n
BDQRTIC	1000	BDQRTIC	5000
CRAGGLVY	1000	CRAGGLVY	5000
FMINSURF	1024	FREUROTH	1000
FREUROTH	5000	LIARWHD	1000
LIARWHD	5000	MOREBV	1000
MOREBV	5000	MOREBV	5000
NCB20	1000	NCB20B	1000
NCB20B	2000	NONCVXUN	1000
NONDIA	1000	NONDQUAR	1000
POWELLSG	1000	POWELLSG	5000
POWELLSG	10000	POWER	1000
DIXMAANA	3000	DIXMAANB	3000
DIXMAANC	3000	DIXMAAND	3000
DIXMAANE	3000	DIXMAANF	3000
DIXMAANG	3000	DIXMAANH	3000
DIXMAANI	3000	DIXMAANJ	3000
DIXMAANK	3000	DIXMAANL	3000
ARWHEAD	5000	BRYBND	5000
BRYBND	10000	DQRTIC	1000
DQRTIC	5000	EDENSCH	2000
ENGVAL1	5000		

From Figure 1, we found that almost 89% of the test problems is solved successfully by the FTRB algorithm, but AINART, TRFBB and FATRA algorithms solve roughly $85\%, 83\%$ and 86% of problems, respectively. Also, the FTRB algorithm solves about 61% of problems at the lowest value of n_i that shows the efficiency of this algorithm compared to the other one. At a glance to Figure 2, one can realize that between the FTRB, AINART, TRFBB and FATRA algorithms, the

FTRB algorithm could solve roughly 57% of the test problems in the lowest value of n_f. Additionally, the results in Figure 3 show that about 67% of the problems in the minimum value of n_g is solved by FTRB algorithm, while this number for AINART, TRFBB and FATRA algorithms are $66\%, 56\%$ and 53%, respectively.

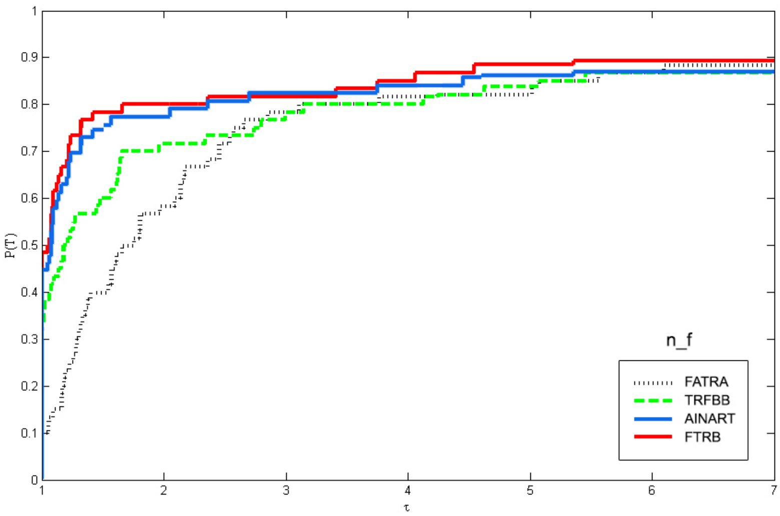

Figure 1: Performance of considered algorithms based on n_f.

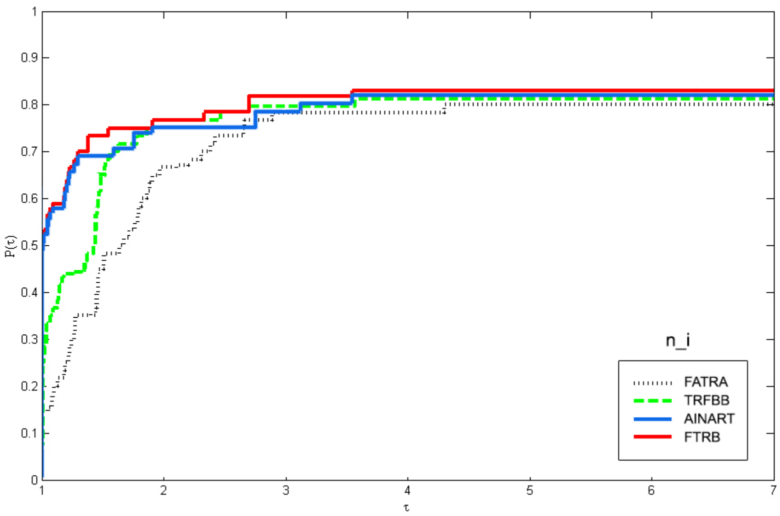

Figure 2: Performance of considered algorithms based on n_i.

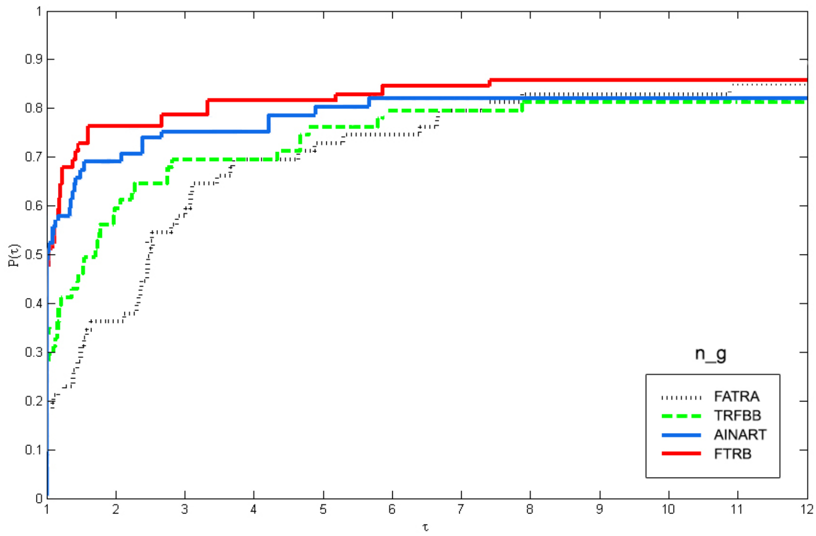

Figure 3: Performance of considered algorithms based on n_g.

Acknowledgments. The authors would like to thank the University of Kashan for supporting this work.

Conflicts of Interest. The authors declare that there are no conflicts of interest regarding the publication of this article.

References

[1] M. Ahookhosh and K. Amini, An efficient nonmonotone trust-region method for unconstrained optimization, *Numer. Algor.* **59** (2011) 523 – 540.

[2] N. Andrei, Accelerated scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization, *Eur. J. Oper. Res.* **204** (2010) 410 – 420.

[3] F. Arzani and M. R. Peyghami, A new nonmonotone fillter Barzilai-Borwein method for solving unconstrained optimization problems, *Int. J. Comput. Math.* (2014).

[4] D. Ataee Tarzanagh, Z. Saeedian, M. R. Peyghami and H. Mesgarani, A new trust region method for solving least-squares transformation of system of equalities and inequalities, *Math. Program. Stud.* **9** (2015) 283 – 310.

[5] J. Barzilai and J. M. Borwein, Two-point step size gradient methods, *IMA J. Numer. Anal.* **8** (1988) 141 – 148.

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

[7] A. Conn, N. Gould and Ph. L.Toint, *Trust Region Methods*, MPSSIAM Series on Optimization, SIAM, Philadelphia, 2000.

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

[9] J. Y. Fan and Y. X. Yuan, A new trust region algorithm with trust region radius converging to zero, *Proc.* 5*th Inter. Conf. Optim. Tech. Appl.*, Hong Kong, 15-17 December 2001, 786 – 794.

[10] M. Fatemi and N. Mahdavi-Amiri, A filter trust-region algorithm for unconstrained optimization with strong global convergence properties, *Comput. Optim. Appl.* **52** (2012) 239 – 266.

[11] M. Fatemi and N. Mahdavi-Amiri, A non-monotone trust region algorithm for unconstrained optimization with dynamic reference iteration updates using filter, *Optimization* **61** (6) (2012) 733 – 763.

[12] R. Fletcher and S. Leyer, Nonlinear programming without a penalty function, *Math. Program. Ser. A* **91** (2002) 239 – 269.

[13] N. I. M. Gould, D. Orban, A. Sartenaer and PhL. Toint, Sensitivity of trust region algorithms to their parameters, *Q. J. Oper. Res.* **3** (2005) 227 – 241.

[14] N. I. M. Gould, D. Orban and P. L. Toint, Cuter and sifdec: a constrained and unconstrained testing environment, revisited, *ACM Trans. Math. Softw.* **29** (4) (2002) 373 – 394.

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

[16] L. Grippo and M. Sciandrone, Nonmontone globalization techniques for the Barzilai-Borwein gradient method, *Comput. Optim. Appl.* **23** (2002) 143–169.

[17] A. Kamandi, K. Amini and M. Ahookhosh, An improved adaptive trust-region algorithm, *Optim. Lett.* **11** (3) (2017) 555 – 569.

[18] A. Kamandi and K. Amini, A new nonmonotone adaptive trust region algorithm, *Appl. Math.* **67** (2022) 233 – 250.

[19] L. Grippo, F. Lampariello and S. Lucidi, A truncated Newton method with nonmonotone line search for unconstrained optimization, *J. Optim. Theory Appl.* **60** (3) (1989) 401 – 419.

[20] D. Li and M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization, *J. Comput. Appl. Math.* **129** (2001) 1535.

[21] J. J. Moré, B. S. Garbow and K. E. Hilstron, Testing unconstrained optimization software, *ACM Trans. Math. Softw.* **7** (1981) $17-41$.

[22] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer Series in Operations Research and Financial Engineering, Springer, NewYork, 2007.

[23] J. Nocedal and Y. Yuan, Combining trust region and line search techniques, In: Y. Yuan (ed.), *Advanced in Nonlinear Programming, Kluwer Academic*, Dordrecht, (1996) $153-175$.

[24] Z. Saeidian and F. Arzani (in Persian), Solving the Unconstrained Optimization Problems Using the Combination of Nonmonotone Trust Region Algorithm and Filter Technique, *J. Oper. Res. Appl.* **17** (1) (2020) $85-101$.

[25] Z. Saeidian and M. R. Peyghami, An adaptive nonmonotone trust region method for unconstrained optimization problems based on a simple subproblem, *Iran. J. Numer. Anal. Optim.* **5** (2) (2015) $95-117$.

[26] Z. Sang and Q. Sun, A self-adaptive trust region method with line search based on a simple subproblem model, *J. Appl. Math. Comput.* **232** (2) (2009) $514-522$.

[27] A. Sartenaer, Automatic determination of an initial trust region in nonlinear programming, *SIAM J. Sci. Comput.* **18** (6) (1997) $1788-1803$.

[28] Z. Shi and J. Guo, A new trust region method for unconstrained optimization, *J. Comput. Appl. Math.* **213** (2008) $509-520$.

[29] Z. J. Shi and H. Q. Wang, A new self-adaptive trust region method for unconstrained optimization, Technical Report, College of Operations Research and Management, Qufu Normal University, 2004.

[30] Y. Xue, H. Liu and Z. Liu, An improved nonmonotone adaptive trust region method, *Appl. Math.* **64** (3) (2019) $335-350$.

[31] H. C. Zhang and W. W. Hager, A nonmonotone line search technique and its application to unconstrained optimization, *SIAM J. Optim.* **14** (2004) $1043-1056$.

[32] X. S. Zhang, J. L. Zhang and L. Z. Liao, An adaptive trust region method and its convergence, *Sci. China.* **45** (2002) $620-631$.

Zeinab Saeidian
Department of Applied Mathematics,
University of Kashan,
Kashan, I. R. Iran
e-mail: saeidian@kashanu.ac.ir

Farshid Arzani
Scientific Computations in Optimization and Systems Engineering (SCOPE),
K. N. Toosi University of Technology,
Tehran, I. R. Iran
e-mail: farzani@dena.kntu.ac.ir