Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 14, No. 4, pp. 621–633, 2018

AN ADAPTIVE RETROSPECTIVE TRUST REGION METHOD BASED ON A HYBRIDIZATION OF THE MONOTONE AND NONMONOTONE ASPECTS

Saeed Rezaee and Saman Babaie–Kafaki*

Abstract: Based on an eigenvalue analysis conducted on the scaled memoryless DFP updating formula, we propose an adaptive trust region radius. Then, based on a retrospective approach, we use a convex combination of the monotone and nonmonotone strategies to develop an adaptive trust region algorithm. Under proper conditions, it is briefly shown that the proposed algorithm is globally and locally superlinearly convergent. Numerical experiments are done on a set of unconstrained optimization test problems of the CUTEr collection, using the Dolan–Moré performance profile. They demonstrate efficiency of the proposed algorithm.

Key words: *unconstrained optimization; trust region method; quasi–Newton update; eigenvalue; adaptive radius; global convergence; superlinear convergence*

Mathematics Subject Classification: *65K05; 90C53; 49M37*

1 Introduction

Consider an unconstrained optimization problem in the following form:

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

in which $f:\mathbb{R}^n\to\mathbb{R}$ is a continuously differentiable function and analytic expression of its gradient is available. As a class of efficient techniques for solving (1.1), recently trust region (TR) algorithms have attracted especial attentions. In each iteration of the methods, a neighborhood is defined around the current iterate, called the trust region, and then, a step is taken to the minimum of an approximation of the objective function within the region. Iterative formula of the TR methods is generally in the following form:

$$x_0\in\mathbb{R}^n,\ x_{k+1}=x_k+s_k,\ k=0,1,...,\tag{1.2}$$

where s_k is the step taken from x_k. In the classical TR method, s_k is often an approximate solution of the following constrained quadratic subproblem:

$$\begin{array}{ll}\min\limits_{s\in\mathbb{R}^n} & m_k(s)=g_k^Ts+\dfrac{1}{2}s^TB_ks,\\ s.t. & ||s||\le\Delta_k,\end{array}\tag{1.3}$$

*Corresponding author

where $g_k = \nabla f(x_k)$, B_k is an approximation of the Hessian $\nabla^2 f(x_k)$, $\Delta_k > 0$ is the TR radius and $||.||$ stands for the Euclidean norm.

To the best of our knowledge, TR methods are much influenced by the local agreement between $f(x_k + s)$ and $m_k(s)$. In order to measure consistency between the exact and the approximate models, the following basic ratio is defined:

$$\rho_k^B = \frac{f(x_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)}. \tag{1.4}$$

If ρ_k^B is close to 1, then the quadratic model (1.3) reasonably approximates the objective function in the current region. So, it is better to increase Δ_k for the next iteration. However, if ρ_k^B is close to zero or negative, then the approximate model (1.3) is far from the exact model. In such situation, Δ_k should be decreased and consequently, the subproblem (1.3) should be resolved. As the final case, if $0 \ll \rho_k^B \ll 1$, then, although the trial step s_k is acceptable, the radius Δ_k should not changed.

As known, successive reduction of the objective function value generated by the monotone iterative methods in the form of (1.2) may slow down the convergence speed, especially in the presence of narrow curved valleys which appear in the strongly nonlinear problems [21]. To overcome this defect, nonmonotone TR methods have been developed based on the nonmonotone line search approaches [10]. As a successful attempt, Toint [22] proposed a TR method with the following nonmonotone version of the ratio (1.4):

$$\rho_k^{NB} = \frac{f_{l(k)} - f(x_k + s_k)}{m_k(0) - m_k(s_k)},$$

in which

$$f_{l(k)} = \max_{0 \le j \le q(k)} \{f_{k-j}\}, \tag{1.5}$$

where $f_i = f(x_i)$, $q(0) = 0$, and for all $k \ge 1$, $0 \le q(k) \le \min\{q(k-1) + 1, N\}$, for some $N \in \mathbb{N}$. Then, based on the fact that the best convergence behavior of the iterative method (1.2) is obtained by employing stronger nonmonotone strategy far from the solution and weaker one near the solution [24], Ahookhoosh et al. [2] proposed a nonmonotone TR method with the following ratio:

$$\hat{\rho}_k^{NB} = \frac{R_k - f(x_k + s_k)}{m_k(0) - m_k(s_k)}, \tag{1.6}$$

where

$$R_k = \epsilon_k f_{l(k)} + (1 - \epsilon_k) f_k, \tag{1.7}$$

in which $\epsilon_k \in [\epsilon_{\min}, \epsilon_{\max}]$, with $\epsilon_{\min} \in [0, 1)$ and $\epsilon_{\max} \in [\epsilon_{\min}, 1]$. The interested reader can find further information about several other nonmonotone TR methods in [1, 5, 8].

It can be seen that in the classical TR method the radius is updated by considering how well the approximate model at x_k predicts the objective function value at x_{k+1}. To employ more available information at x_{k+1} in order to find a more appropriate region for predicting the objective function values, after the step s_k was accepted based on the value of ρ_k^B, Bastin et al. [4] applied the following retrospective ratio to update the TR radius:

$$\rho_{k+1}^R = \frac{f(x_k) - f(x_k + s_k)}{m_{k+1}(0) - m_{k+1}(s_k)}. \tag{1.8}$$

If ρ_{k+1}^R is negative or a small positive number near to zero, then Δ_{k+1} is decreased; otherwise, it is increased or left unchanged. Also, Ataee Tarzanagh et al. [20] proposed a nonmonotone

adaptive retrospective TR method based on the following nonmonotone versions of the standard and the retrospective ratios:

$$\tilde{\rho}_k^{NB} = \frac{(1+\phi_k)R_k - f(x_k+s_k)}{m_k(0) - m_k(s_k)},$$

$$\tilde{\rho}_{k+1}^{NR} = \frac{(1+\phi_k)R_k - f(x_k+s_k)}{m_{k+1}(0) - m_{k+1}(s_k)}, \tag{1.9}$$

where R_k is defined by (1.7), and ϕ_k is given by

$$\phi_k = \begin{cases} \eta_k, & R_k \geq 0, \\ 0, & R_k < 0, \end{cases}$$

in which $\{\eta_k\}_{k=0}^{\infty}$ is a positive sequence satisfying

$$\sum_{k=0}^{\infty} \eta_k < \infty.$$

Here, we propose another adaptive TR method by hybridizing monotone and nonmonotone approaches in a retrospective scheme. This work is organized as follows. In Section 2, we suggest an adaptive choice for the TR radius using the scaled memoryless DFP updating formula. Then, we deal with an adaptive TR algorithm together with its global and superlinear convergence properties in Section 3. Comparative numerical results are reported in Section 4 and conclusions are drawn in Section 5.

2 An Adaptive Formula for the Trust Region Radius

As seen in (1.4), the gradient or the Hessian information is not explicitly employed to compute the radius Δ_k in the classical TR method. To overcome this defect, using the steepest descent direction, Fan and Yuan [7] proposed the following adaptive TR radius:

$$\Delta_k = \upsilon_k ||g_k||,$$

in which υ_k is a positive parameter computed according to the magnitude of the ratio ρ_k^B. Using the quasi–Newton search direction [19], Zhang et al. [25] employed both of the first and the second order information of the objective function to suggest another adaptive choice for the TR radius as follows:

$$\Delta_k = t^p ||\hat{B}_k^{-1}||.||g_k||, \tag{2.1}$$

where $t \in (0,1)$ is a constant, p is a nonnegative integer and $\hat{B}_k = B_k + iI$, with some $i \in \mathbb{N} \cup \{0\}$, is a positive definite matrix approximation of the Hessian. Despite effectiveness of the approach of [25], computing Δ_k by (2.1) may require a matrix inverse estimation which causes extra computational cost. In order to overcome this possible drawback, Shi and Wang [18] dealt with the following updating formula for the TR radius:

$$\Delta_k = t^p \frac{||g_k||^3}{g_k^T \hat{B}_k g_k}, \tag{2.2}$$

with t, p and $\hat{B}_k$ are defined as above. In another effort, Shi and Guo [17] proposed an extension of (2.2) as follows:

$$\Delta_k = -t^p \frac{g_k^T q_k}{q_k^T \hat{B}_k q_k} ||q_k||, \tag{2.3}$$

with t, p and $\hat{B}_k$ as defined for (2.1), and the vector parameter $q_k \in \mathbb{R}^n$ satisfying the angle condition [19], i.e.

$$-\frac{g_k^T q_k}{||g_k||.||q_k||} \geq \tau, \tag{2.4}$$

for some constant $\tau \in (0, 1]$. Also, recently Peyghami and Ataee Tarzanagh [15] suggested a truncated version of (2.3) given by

$$\Delta_k = \min \left\{ -\upsilon_k \frac{g_k^T q_k}{q_k^T \hat{B}_k q_k} ||q_k||, \Delta_{max} \right\},$$

in which Δ_{max} is a positive constant and υ_k is computed based on the magnitude of the TR ratio.

Here, conducting an eigenvalue analysis on the scaled memoryless DFP updating formula, we suggest another adaptive TR radius. In this context, the following preliminaries are needed.

As a class of line search–based techniques, quasi–Newton methods are of particular performance for solving (1.1) since they do not require explicit expressions of the second derivatives and are often globally and locally superlinearly convergent [19]. Iterative formula of the methods is given by

$$x_0 \in \mathbb{R}^n, \ x_k = x_{k-1} + \alpha_{k-1} d_{k-1}, \ k = 1, 2, ...,$$

where α_{k-1} is a step length to be computed by a line search strategy and d_{k-1} is the search direction to be calculated by

$$d_{k-1} = -H_{k-1} g_{k-1},$$

in which $H_{k-1} \in \mathbb{R}^{n \times n}$ is an approximation of the inverse Hessian; more precisely, $H_{k-1} \approx \nabla^2 f(x_{k-1})^{-1}$. The methods are characterized by the fact that H_{k-1} is effectively updated to achieve a new matrix H_k as an approximation of $\nabla^2 f(x_k)^{-1}$ satisfying the secant equation [19], i.e.

$$H_k y_{k-1} = s_{k-1}, \tag{2.5}$$

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

Among the well–known and effective quasi–Newton updating formulas there is the DFP formula [19] given by

$$H_k^{DFP} = H_{k-1} + \frac{s_{k-1} s_{k-1}^T}{s_{k-1}^T y_{k-1}} - \frac{H_{k-1} y_{k-1} y_{k-1}^T H_{k-1}}{y_{k-1}^T H_{k-1} y_{k-1}}. \tag{2.6}$$

It can be seen that if H_{k-1} is a positive definite matrix and the line search ensures that $s_{k-1}^T y_{k-1} > 0$, as guaranteed by the popular Wolfe conditions [23], then H_k is also a positive definite matrix [19] and consequently, the search direction $d_k = -H_k g_k$ is a descent direction. Also, under convexity assumption on the objective function, the DFP method has been shown to be globally and locally superlinearly convergent [19].

In order to achieve an ideal distribution of the eigenvalues of the DFP updating formula, improving the condition number of successive approximations of the inverse Hessian and

consequently, increasing numerical stability of the method, the scaled DFP update has been developed [19], replacing H_{k-1} by $\theta_{k-1}H_{k-1}$ in (2.6) as follows:

$$H_k^{SDFP} = \theta_{k-1}H_{k-1} + \frac{s_{k-1}s_{k-1}^T}{s_{k-1}^T y_{k-1}} - \theta_{k-1}\frac{H_{k-1}y_{k-1}y_{k-1}^T H_{k-1}}{y_{k-1}^T H_{k-1}y_{k-1}}, \tag{2.7}$$

in which $\theta_{k-1} > 0$ is called the scaling parameter, can be computed by the Oren–Spedicato formula [14], i.e.

$$\theta_{k-1} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T H_{k-1}y_{k-1}}. \tag{2.8}$$

Although the scaled DFP method is numerically efficient [14], as an important defect the method needs to save the matrix $H_{k-1} \in \mathbb{R}^{n\times n}$ in each iteration, being improper for solving large–scale problems. Hence, replacing H_{k-1} by the identity matrix in (2.7), scaled memoryless DFP updating formula has been proposed as follows:

$$H_k^{\theta} = \theta_{k-1}I + \frac{s_{k-1}s_{k-1}^T}{s_{k-1}^T y_{k-1}} - \theta_{k-1}\frac{y_{k-1}y_{k-1}^T}{y_{k-1}^T y_{k-1}},$$

with the following memoryless version of the scaling parameter (2.8):

$$\theta_{k-1} = \frac{s_{k-1}^T y_{k-1}}{||y_{k-1}||^2}. \tag{2.9}$$

As shown in [3], eigenvalues of H_k^{θ} are θ_{k-1} with multiplicity $n-2$, and two other scalars λ_{k-1}^- and λ_{k-1}^+. Also, when $s_{k-1}^T y_{k-1} > 0$, it can be seen that

$$\lambda_{k-1}^- + \lambda_{k-1}^+ = \theta_{k-1} + \frac{||s_{k-1}||^2}{s_{k-1}^T y_{k-1}} > \theta_{k-1}.$$

So,

$$||H_k^{\theta}|| = \max\{\theta_{k-1}, \lambda_{k-1}^-, \lambda_{k-1}^+\} \leq \theta_{k-1} + \frac{||s_{k-1}||^2}{s_{k-1}^T y_{k-1}}.$$

Now, when θ_{k-1} is computed by (2.9), we get

$$||H_k^{\theta}|| \leq \frac{s_{k-1}^T y_{k-1}}{||y_{k-1}||^2} + \frac{||s_{k-1}||^2}{s_{k-1}^T y_{k-1}}.$$

Hence, based on the approach of [25] which leads to (2.1), here we suggest the following adaptive choice for the TR radius:

$$\check{\Delta}_k = \nu_k||g_k||\left(\frac{s_{k-1}^T y_{k-1}}{||y_{k-1}||^2} + \frac{||s_{k-1}||^2}{s_{k-1}^T y_{k-1}}\right), \tag{2.10}$$

where ν_k is computed based on the magnitude of the TR ratio.

Note that inequality $s_{k-1}^T y_{k-1} > 0$ may not always hold in the TR methods. So, to ensure positiveness of the TR radius (2.10), here we replace $s_{k-1}^T y_{k-1}$ by its absolute value. Also, the formula (2.10) is not well–defined for $k = 0$. Hence, the following modified version of (2.10) is now immediate:

$$\Delta_k = \nu_k \delta_k,$$

where

$$\delta_k = ||g_k|| \begin{cases} 1, & k = 0, \\ \dfrac{|s_{k-1}^T y_{k-1}|}{||y_{k-1}||^2} + \dfrac{||s_{k-1}||^2}{|s_{k-1}^T y_{k-1}|}, & k > 0. \end{cases} \tag{2.11}$$

3 A Hybrid Adaptive Trust Region Algorithm

Here, using the adaptive TR radius (2.11), we describe structure of our adaptive retrospective TR algorithm as a modified version of the TR algorithm proposed in [20].

As the main scheme of the algorithm, we use the nonmonotone ratio $\hat{\rho}_k^{NB}$ given by (1.6) to decide whether the trial step s_k is acceptable or not, while we employ a convex combination of $\hat{\rho}_k^{NB}$ and the retrospective monotone ratio ρ_{k+1}^R given by (1.8) to update the TR radius. So, as an inheritance of the nonmonotone TR methods, a good move from x_k to x_{k+1} is possible. Also, the retrospective monotone ratio ρ_{k+1}^R provides more reasonable information about the relationship between the exact and the approximate models at x_{k+1}. Moreover, since local information are more useful to measure consistency between the exact and the approximate models, it is more reasonable to use available information from the current iteration in contrast to using the information available from more than one previous step. Hence, we give a chance to the monotone ratio ρ_{k+1}^R to play a role in computing the TR radius.

To describe with more details, at the kth iteration of the algorithm we assume that the current point x_k, the parameter ν_k and the radius Δ_k are available. So, we solve the TR subproblem (1.3) to find the trial step s_k and then, we compute the ratio $\hat{\rho}_k^{NB}$ by (1.6) to accept or reject the trial step. If $\hat{\rho}_k^{NB}$ is negative or a small positive number, then we set $x_{k+1} = x_k$ and shrink the radius as $\Delta_{k+1} = \min\{\gamma_0||s_k||, \nu_{k+1}\delta_{k+1}\}$, where $\gamma_0 \in (0,1)$ is a prespecified constant, δ_{k+1} is computed by (2.11) and $\nu_{k+1} = \sigma_0\nu_k$, with a given constant $\sigma_0 \in (0,1)$. In such situation, the TR subproblem should be solved again. Otherwise, we set $x_{k+1} = x_k + s_k$, compute ρ_{k+1}^R by (1.8) and based on the approach of [20], we define the following ratio, being a convex combination of $\hat{\rho}_k^{NB}$ and ρ_{k+1}^R:

$$\rho_{k+1}^C = \lambda\hat{\rho}_k^{NB} + (1-\lambda)\rho_{k+1}^R, \tag{3.1}$$

where $\lambda \in [0,1]$ is a constant. Then, we update the radius by $\Delta_{k+1} = \min\{\tilde{\Delta}_{k+1}, \Delta_{max}\}$ in which $\tilde{\Delta}_{k+1}$ is computed as follows:

$$\tilde{\Delta}_{k+1} = \begin{cases} \nu_{k+1}\delta_{k+1}, & \rho_{k+1}^C \geq \mu_1, \\ \min\{\gamma_0||s_k||, \nu_{k+1}\delta_{k+1}\}, & \rho_{k+1}^C < \mu_1, \end{cases} \tag{3.2}$$

with γ_0 and δ_{k+1} are defined as above, the constant $\mu_1 \in (0,1)$ and the parameter ν_{k+1} updated by

$$\nu_{k+1} = \begin{cases} \min\{\sigma_1\nu_k, \nu_{max}\}, & \rho_{k+1}^C > \mu_2, \\ \nu_k, & \mu_1 \leq \rho_{k+1}^C \leq \mu_2, \\ \sigma_0\nu_k, & \rho_{k+1}^C < \mu_1, \end{cases} \tag{3.3}$$

in which $\sigma_1 > 1$ and $\mu_2 \in (\mu_1, 1)$ are prespecified constants and σ_0 is defined as above. Considering these preliminaries, now we are in a position to describe our algorithm in details.

Algorithm 3.1. An adaptive retrospective trust region method based on a hybridization of the monotone and nonmonotone aspects (ARMNMTR)

Step 0 {Initialization} Choose an initial point $x_0 \in \mathbb{R}^n$, a symmetric matrix $B_0 \in \mathbb{R}^{n\times n}$, and the constants $0 < \mu_1 < \mu_2 < 1$, $0 < \sigma_0 < 1 < \sigma_1$, $0 < \epsilon_{min} < \epsilon_{max} < 1$, $0 < \gamma_0 < 1$, $\nu_0 > 0$, $\nu_{max} > 0$, $\Delta_{max} > 0$, $N > 0$, and $\varepsilon > 0$. Compute f_0 and set $k = 0$.

Step 1 {Radius update} **If** $k = 0$, **then** set $\Delta_k = \min\{\nu_k\delta_k, \Delta_{max}\}$ and **goto** Step 2. **If** $\hat{\rho}_{k-1}^{NB} < \mu_1$, **then** set $\nu_k = \sigma_0\nu_{k-1}$ and $\Delta_k = \min\{\gamma_0||s_{k-1}||, \nu_k\delta_k\}$; **else** compute ρ_k^R and ρ_k^C respectively by (1.8) and (3.1), update ν_k using (3.3), and set $\Delta_k = \min\{\tilde{\Delta}_k, \Delta_{max}\}$ where $\tilde{\Delta}_k$ is given by (3.2).

Step 2 {Stopping criterion} **If** $||g_k|| < \varepsilon$, **then stop**.

Step 3 Solve the subproblem (1.3) to find the trial step s_k and compute $\hat{\rho}_k^{NB}$ by (1.6).

Step 4 **If** $\hat{\rho}_k^{NB} \geq \mu_1$, **then** set $x_{k+1} = x_k + s_k$; **else** set $x_{k+1} = x_k$.

Step 5 Compute the new Hessian approximation B_{k+1} using a quasi–Newton updating formula, set $k = k + 1$ and **goto** Step 1.

Now, we discuss well–definiteness as well as convergence of Algorithm 3.1. In this context, the following standard assumptions are needed.

Assumption 3.1.

H1 The objective function f is bounded below, the level set $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n | f(x) \leq f_0\}$ is bounded, and ∇f is uniformly continuous on a compact convex set Ω that contains the level set $\mathcal{L}(x_0)$.

H2 The matrix B_k is uniformly bounded, i.e. there exists a positive constant M_1 such that $||B_k|| \leq M_1$, $\forall k \in \mathbb{N} \cup \{0\}$.

H3 There exists a positive constant m such that $s^T B_k s \geq m||s||^2$, $\forall s \in \mathbb{R}^n$, $\forall k \in \mathbb{N} \cup \{0\}$.

If f is a twice continuously differentiable function, then H1 implies that $||\nabla^2 f||$ is uniformly continuous and bounded on Ω. Hence, there exists a positive constant L such that $||\nabla^2 f(x)|| \leq L$, $\forall x \in \Omega$. So, from the mean value theorem, we have

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

which ensures that ∇f is Lipschitz continuous on Ω.

To simplify the discussion, we define the following two index sets:

$$I_1 = \{k | \hat{\rho}_k^{NB} \geq \mu_1\}, \ I_2 = \{k | \hat{\rho}_k^{NB} < \mu_1\}.$$

The following results are now immediate, showing well–definiteness of the steps of Algorithm 3.1.

Lemma 3.2. *If s_k is a solution of (1.3), then*

$$m_k(0) - m_k(s_k) \geq \theta||g_k|| \min\left\{\Delta_k, \frac{||g_k||}{||B_k||}\right\}, \ \forall k \geq 0,$$

where $\theta \in (0, 1)$ is a constant.

Proof. The proof is similar to the proof of Theorem 4.4 of [12] and here is omitted. □

Lemma 3.3. *Suppose that Assumptions 3.1 hold. For the sequence $\{x_k\}_{k\geq 0}$ generated by Algorithm 3.1, assume that there exists a constant $\varepsilon \in (0,1)$ such that*

$$||g_k|| \geq \varepsilon, \ \forall k \geq 0. \tag{3.4}$$

For each k, there exists a nonnegative integer p such that x_{k+p+1} is a successful iteration point in the sense that $k+p \in I_1$.

Proof. At the contrary, suppose that there exists an iteration k so that for all nonnegative integers $p \geq 0$, x_{k+p+1} is an unsuccessful iteration point, i.e.

$$\hat{\rho}_{k+p}^{NB} < \mu_1.$$

Now, from Step 1 of Algorithm 3.1, Assumptions 3.1, and since the quasi–Newton updates (used in Step 5 of Algorithm 3.1) satisfy the secant equation (2.5), we have

$$\begin{aligned}
\Delta_{k+p+1} &\leq \sigma_0^{p+1}\nu_k||g_k||\left(\frac{||s_{k-1}||^2}{|s_{k-1}^T y_{k-1}|}+\frac{|s_{k-1}^T y_{k-1}|}{||y_{k-1}||^2}\right)\\
&\leq \sigma_0^{p+1}\nu_k||g_k||\left(\frac{||s_{k-1}||^2}{|s_{k-1}^T B_k s_{k-1}|}+\frac{|s_{k-1}^T B_k s_{k-1}|}{||B_k s_{k-1}||^2}\right)\\
&\leq \sigma_0^{p+1}\nu_k||g_k||\left(\frac{||s_{k-1}||^2}{m||s_{k-1}||^2}+\frac{||s_{k-1}||^2|s_{k-1}^T B_k s_{k-1}|}{|s_{k-1}^T B_k s_{k-1}|^2}\right)\\
&\leq \sigma_0^{p+1}\nu_k||g_k||\left(\frac{1}{m}+\frac{||s_{k-1}||^2}{m||s_{k-1}||^2}\right)\\
&\leq \sigma_0^{p+1}\nu_k||g_k||\left(\frac{1}{m}+\frac{1}{m}\right)\\
&\leq \frac{2\sigma_0^{p+1}\nu_{max}}{m}||g_k||.
\end{aligned}$$

Setting $\phi_k = 0$ in (1.9), the remainder of the proof is similar to the proof of Lemma 4.2 of [20] and here is omitted. □

Remark 3.4. As a consequence of Lemma 3.3, one can realize that whenever Algorithm 3.1 does not stop in finite number of iterations, then the index set I_1 is infinite. In this situation, there exists a bijection between I_1 and $\{\bar{L}N + r | \bar{L} \in \mathbb{N}\bigcup\{0\}, 1 \leq r \leq N\}$, with the same positive integer N as in (1.5). Thus, without loss of generality we can assume that $I_1 = \{\bar{L}N + r | \bar{L} \in \mathbb{N}\bigcup\{0\}, 1 \leq r \leq N\}$.

In what follows, we present some results which are necessary to establish convergence of Algorithm 3.1.

Lemma 3.5. *Assume that $\{x_k\}_{k\geq 0}$ generated by Algorithm 3.1 is an infinite sequence and N is the same integer constant as in (1.5). Then, for every $\bar{L}N + r \in I_1, 1 \leq r \leq N$, we have*

$$f_{\bar{L}N+r} \leq |f_0| - \sum_{i=0}^{\bar{L}} \omega_{s(i)}, \ \bar{L} = 0, 1, ...,$$

where

$$\omega_{s(i)} = \min_{iN \le j \le (i+1)N-1} \omega_j, \tag{3.5}$$

in which $\omega_j = \theta\mu_1 ||g_j|| \min\left\{\Delta_j, \dfrac{||g_j||}{||B_j||}\right\}$.

Proof. Setting $\phi_k = 0$ in (1.9), the proof is similar to the proof of Lemma 4.6 of [20] and here is omitted. □

Theorem 3.6. *If Assumptions 3.1 hold, then Algorithm 3.1 either stops at a stationary point of (1.1) or generates an infinite sequence* $\{x_k\}_{k\ge 0}$ *such that*

$$\liminf_{k\to\infty} ||g_k|| = 0. \tag{3.6}$$

Proof. The proof is similar to the proof of Theorem 4.1 of [20] and here is omitted. □

Theorem 3.7. *Suppose that Assumptions 3.1 hold and Algorithm 3.1 generates an infinite sequence* $\{x_k\}_{k\ge 0}$ *which converges to the optimal solution* x^*. *Consider an arbitrary function* $\tilde{H} : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ *which is Lipschitz continuous in a neighborhood of* x^* *such that* $\tilde{H}(x^*)$ *is a positive definite matrix. If*

$$\lim_{k\to\infty} \frac{||g_k + \tilde{H}(x^*)s_k||}{||s_k||} = 0,$$

then the convergence rate of $\{x_k\}_{k\ge 0}$ *is superlinear, i.e.* $||x_{k+1} - x^*|| = o(||x_k - x^*||)$.

Proof. The proof is similar to the proof of Theorem 4.1 of [5] and here is omitted. □

4 Numerical Experiments

Here, we present some numerical results obtained by applying MATLAB 7.14.0.739 (R2012a) implementations of the ARMNMTR method (Algorithm 3.1), a version of ARMNMTR using the adaptive radius proposed in [16] instead of the radius (2.11), here called ARMNMTR–V2, the adaptive retrospective nonmonotone TR algorithm proposed in [20], here called ARNMTR, and a modified version of ARNMTR (based on our hybridization approach) in which the nonmonotone ratio (1.9) is replaced by the monotone ratio (1.8), here called MARNMTR. The runs were performed on a set of 68 unconstrained optimization test problems of the CUTEr collection [9] with the minimum dimension being equal to 100, as specified in Table 1, using a computer Intel(R) Core(TM) 2 Duo CPU 2.00 GHz with 1.50 GB of RAM.

The ARNMTR method has been run with the same parameter values as specified in [20]. Also, adopting suggestions of [20], for ARMNMTR, ARMNMTR-V2 and MARNMTR we set $\mu_1 = 0.05$, $\mu_2 = 0.9$, $\sigma_0 = 0.2$, $\sigma_1 = 5$, $\lambda = 0.5$, $\gamma_0 = 0.25$, $\nu_0 = 0.1$, $\nu_{max} = 2$, $\eta_k = \dfrac{10}{(k+1)^2}$, $\Delta_{max} = 100$, $N = 2n$ if $n < 5$, and $N = 10$, otherwise. Moreover, starting from $\epsilon_0 = 0.85$, we set

$$\epsilon_k = \begin{cases} \epsilon_0/2, & k = 1, \\ (\epsilon_{k-1} + \epsilon_{k-2})/2, & k \ge 2. \end{cases}$$

Among the wide scope of the choices of q_k satisfying (2.4), for ARNMTR and MARNMTR we set $q_k = -B^{-1}g_k$. In all the algorithms, we used the following scaled memoryless DFP approximation of the Hessian [19]:

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

Table 1: Test problems data

Function	n	Function	n	Function	n
ARGLINA	200	DIXMAANH	3000	MSQRTBLS	1024
ARGLINB	200	DIXMAANI	3000	NCB20	5010
ARWHEAD	5000	DIXMAANJ	3000	NCB20B	5000
BDEXP	5000	DIXMAANK	3000	NONCVXU2	5000
BDQRTIC	5000	DIXMAANL	3000	NONDIA	5000
BIGGSB1	5000	DIXON3DQ	10000	NONDQUAR	5000
BOX	10000	DQDRTIC	5000	PENALTY1	1000
BROWNAL	200	DQRTIC	5000	PENALTY2	200
BROYDN7D	5000	EDENSCH	2000	POWELLSG	5000
BRYBND	5000	EG2	1000	POWER	10000
CHAINWOO	4000	ENGVAL1	5000	QUARTC	5000
COSINE	10000	EXTROSNB	1000	SCHMVETT	5000
CRAGGLVY	5000	FLETCBV2	5000	SENSORS	100
CURLY10	10000	FLETCHCR	1000	SINQUAD	5000
CURLY20	10000	FMINSRF2	5625	SPARSQUR	10000
CURLY30	10000	FMINSURF	5625	SPMSRTLS	4999
DIXMAANA	3000	FREUROTH	5000	SROSENBR	5000
DIXMAANB	3000	GENHUMPS	5000	TOINTGSS	5000
DIXMAANC	3000	GENROSE	500	TQUARTIC	5000
DIXMAAND	3000	LIARWHD	5000	TRIDIA	5000
DIXMAANE	3000	MANCINO	100	VARDIM	200
DIXMAANF	3000	MOREBV	5000	WOODS	4000
DIXMAANG	3000	MSQRTALS	1024		

where $\theta_k^B = \dfrac{s_k^T y_k}{||s_k||^2}$ (see also [3, 13, 14]). Note that we set $B_{k+1} = B_k$ whenever the curvature condition (i.e. $s_k^T y_k > 0$) does not hold. Considering relationship between the DFP and the BFGS updates [19], the inverse of B_{k+1} can be written as

$$H_{k+1} = \theta_k^H I - \theta_k^H \frac{s_k y_k^T + y_k s_k^T}{s_k^T y_k} + (1 + \theta_k^H \frac{||y_k||^2}{s_k^T y_k}) \frac{s_k s_k^T}{s_k^T y_k},$$

with $\theta_k^H = \dfrac{s_k^T y_k}{||y_k||^2}$. All attempts for finding an approximation of the solution were terminated by reaching maximum of 20000 iterations or achieving a solution with $||g_k||_\infty < 10^{-6}(1 + |f(x_k)|)$.

Efficiency comparisons were drawn using the Dolan–Moré performance profile [6] on the running time and the total number of function and gradient evaluations being equal to $N_f + 3N_g$, where N_f and N_g respectively denote the number of function and gradient evaluations [11]. Performance profile gives, for every $\omega \geq 1$, the proportion $p(\omega)$ of the test problems that each considered algorithmic variant has a performance within a factor of ω of the best.

Figures 1 and 2 demonstrate the results of comparisons. As the figures show, although the algorithms are approximately competitive with respect to the total number of function and

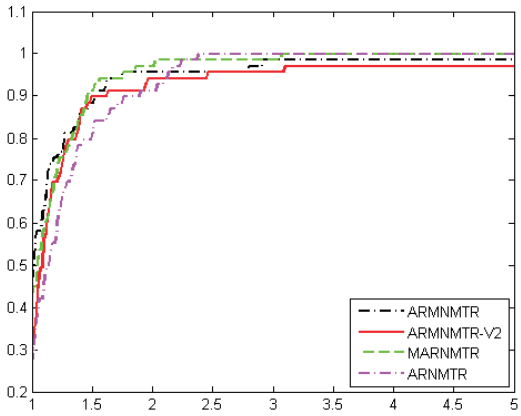

Figure 1: Total number of function and gradient evaluations performance profiles

gradient evaluations, ARMNMTR outperforms ARMNMTR–V2, ARNMTR and MARNMTR with respect to the running time. This seems reasonable since computing the TR radius in ARMNMTR is low–cost in contrast to the radius computation of the other three algorithms. Also, the figures show that MARNMTR is preferable to ARNMTR, demonstrating effectiveness of our hybridization of the monotone and nonmonotone strategies.

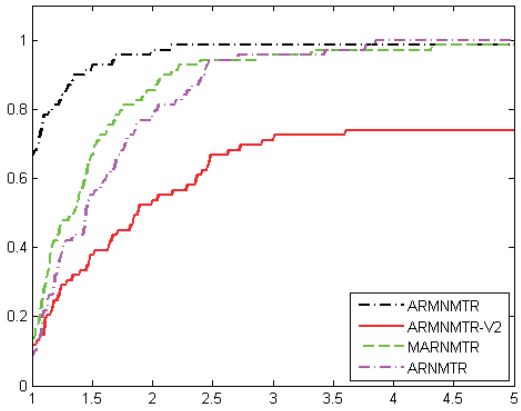

Figure 2: CPU time performance profiles

5 Conclusions

Hybridizing monotone and nomonotone strategies, a retrospective trust region algorithm has been proposed. An eigenvalue analysis which has been carried out on the scaled memoryless DFP updating formula plays an important role in computing the trust region radius adaptively, in a reasonable computational cost. The method has been briefly shown to

be globally and locally superlinearly convergent. Numerical experiments showed that our hybridization scheme is practically effective. Also, they showed efficiency of the propose algorithm, especially with respect to the running time.

Acknowledgements

The authors thank the Research Council of Semnan University for its support. They are also grateful to the anonymous referees for their valuable comments and suggestions helped to improve the quality of this work.

References

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

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

[3] S. Babaie-Kafaki, On optimality of the parameters of self–scaling memoryless quasi–Newton updating formulae, *J. Optim. Theory Appl.* 167 (2015) 91–101.

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

[5] Z. Cui and B. Wu, A new modified nonmonotone adaptive trust region method for unconstrained optimization, *Comput. Optim. Appl.* 53 (2012) 795–806.

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

[7] J.Y. Fan and Y.X. Yuan, A new trust region algorithm with trust region radius converging to zero, in *Proceedings of the 5th International Conference on Optimization: Techniques and Applications*, D. Li di (ed.), Hong Kong, 2001, pp. 786–794.

[8] J.H. Fu and W.Y. Sun, Nonmonotone adaptive trust region method for unconstrained optimization problems, *Appl. Math. Comput.* 163 (2005) 489–504.

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

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

[11] W.W.Hager and H. Zhang, Algorithm 851: CG_Descent, a conjugate gradient method with guaranteed descent, *ACM Trans. Math. Softw.* 32 (2006) 113–137.

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

[13] S.S Oren and D.G. Luenberger, Self–scaling variable metric (SSVM) algorithms. I. Criteria and sufficient conditions for scaling a class of algorithms, *Management Sci.* 20 (1974) 845–862.

[14] S.S. Oren and E.J. Spedicato, Optimal conditioning of self–scaling variable metric algorithms, *Math. Program.* 10 (1976) 70–90,

[15] M.R. Peyghami and D. Ataee Tarzanagh, A relaxed nonmonotone adaptive trust region method for solving unconstrained optimization problems, Comput. Optim. Appl. 61 (2015) 321–341.

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

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

[18] 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.

[19] W. Sun and Y.X. Yuan, Optimization Theory and Methods: Nonlinear Programming, Springer, New York, 2006.

[20] D. Ataee Tarzanagh, M. Peyghami and F. Bastin, A new nonmonotone adaptive retrospective trust region method for unconstrained optimization problems, *J. Optim. Theory Appl.*167 (2015)676–692.

[21] Ph. L. Toint, An assessment of nonmonotone line search techniques for unconstrained Optimization, *SIAM J. Sci. Comput.* 17 (1996) 725–739.

[22] Ph. L. Toint, Nonmonotone trust region algorithms for nonlinear optimization subject to convex constraints, *SIAM J. Sci. Comput.* 17 (1996) 725–739.

[23] P. Wolfe, Convergence conditions for ascent methods, *SIAM Rev.* 11 (1969) 226–235.

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

[25] X.S. Zhang, J.L. Zhang and L.Z. Liao, An adaptive trust region method and its convergence, *Sci. China Ser. A Math.* 45 (2002) 620–631.

Manuscript received 8 December 2016
revised 10 May 2017
accepted for publication 1 June 2017

Saeed Rezaee
Department of Mathematics, Faculty of Mathematics
Statistics and Computer Science
Semnan University, P.O. Box: 35195–363, Semnan, Iran
E-mail address: s.rezaee@semnan.ac.ir

Saman Babaie–Kafaki
Department of Mathematics, Faculty of Mathematics
Statistics and Computer Science
Semnan University, P.O. Box: 35195–363, Semnan, Iran
E-mail address: sbk@semnan.ac.ir