RAIRO-Oper. Res. 53 (2019) 829–839
https://doi.org/10.1051/ro/2017057

RAIRO Operations Research
www.rairo-ro.org

AN ADAPTIVE NONMONOTONE TRUST REGION METHOD BASED ON A MODIFIED SCALAR APPROXIMATION OF THE HESSIAN IN THE SUCCESSIVE QUADRATIC SUBPROBLEMS

Saeed Rezaee[1] and Saman Babaie-Kafaki[1,*]

Abstract. Based on a modified secant equation, we propose a scalar approximation of the Hessian to be used in the trust region subproblem. Then, we suggest an adaptive nonmonotone trust region algorithm with a simple quadratic model. 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.

Mathematics Subject Classification. 65K05, 90C53, 49M37.

Received January 14, 2017. Accepted July 23, 2017.

1. Introduction

Here, we consider the following unconstrained optimization problem:

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

in which $f:\mathbb{R}^n\to\mathbb{R}$ is a continuously differentiable function and analytic expression of its gradient is available. Iterative methods for solving (1.1) are essentially divided into two categories: line search methods and trust region (TR) methods. Line search methods compute an appropriate step length along a descent direction passing through the previous iterate. On the other hand, in an iteration of a TR method a neighborhood is defined around the current iterate, called the trust region, and then, a trial step is taken to the minimum of an approximation of the objective function within the region.

Iterative formula of a TR method is generally given by

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

Keywords. Unconstrained optimization, trust region method, secant equation, adaptive radius, global convergence, superlinear convergence.

[1] Department of Mathematics, Faculty of Mathematics, Statistics and Computer Science, Semnan University, P.O. Box: 35195–363, Semnan, Iran.
[*]Corresponding author: sbk@semnan.ac.ir

in which the trial step s_k is often an approximate solution of the following constrained quadratic subproblem:

$$\begin{array}{ll} \min\limits_{s\in\mathbb{R}^n} & m_k(s) = g_k^T s + \frac{1}{2} s^T B_k s, \\ s.t. & ||s|| \leq \Delta_k, \end{array} \tag{1.2}$$

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. The trial step s_k may be accepted or rejected due to the agreement of $f(x_k + s_k)$ and $m_k(s_k)$. So, in order to measure consistency between the exact and the approximate models, the following classical ratio is defined:

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)}. \tag{1.3}$$

If ρ_k is close to 1, then the quadratic model (1.2) is a good predictor of the objective function behavior over the current region. In such situation, the trial step s_k is accepted and the next TR radius can be increased. However, if ρ_k is close to zero or negative, the trial step s_k is rejected and the TR subproblem is solved again within a smaller region. Moreover, when $0 \ll \rho_k \ll 1$, although the trial step s_k is acceptable, the radius should not be changed.

Solving subproblem (1.2) plays a crucial role in the TR methods. Several efficient techniques for finding some approximate solutions of (1.2) have been proposed in the literature [18]. However, the methods often need to compute and save the matrix $B_k \in \mathbb{R}^{n\times n}$ in each iteration, that is costly, especially in large-scale cases. To overcome this defect, finding simple approximations for the Hessian has recently attracted especial attentions. For example, Zhu *et al.* [24] and also, Sang and Sun [17] proposed a positive definite diagonal form for B_k that needs low memory storage and is computationally low cost. Moreover, using Taylor expansion, Zhou *et al.* [21] suggested the following simple quadratic model with a scalar approximation for the matrix B_k:

$$\begin{array}{ll} \min\limits_{s\in\mathbb{R}^n} & m_k(s) = g_k^T s + \frac{1}{2} \gamma(x_k) s^T s, \\ s.t. & ||s|| \leq \Delta_k, \end{array} \tag{1.4}$$

where

$$\gamma(x_k) = \begin{cases} \hat{\gamma}(x_k), & \hat{\gamma}(x_k) > 0, \\ \dfrac{2\delta}{s_{k-1}^T s_{k-1}}, & \hat{\gamma}(x_k) \leq 0, \end{cases} \tag{1.5}$$

in which δ is a small positive constant and

$$\hat{\gamma}(x_k) = \frac{2\left(f(x_{k-1}) - f(x_k) + g_k^T s_{k-1}\right)}{s_{k-1}^T s_{k-1}}.$$

Then, as a modified version of (1.5), Zhou and Zhang [22] suggested another formula for $\gamma(x_k)$ in (1.4) as follows:

$$\gamma(x_k) = \frac{2\left(f(x_{k-1}) - f(x_k) + (1+\eta_k) g_k^T s_{k-1}\right)}{s_{k-1}^T s_{k-1}},$$

in which

$$\eta_k = \frac{f(x_k) - f(x_{k-1}) - g_k^T s_{k-1} + \delta}{g_k^T s_{k-1}},$$

with the scalar δ as defined in (1.5). More recently, Saeidian and Peyghami [16] dealt with the following choice for $\gamma(x_k)$:

$$\gamma(x_k) = \bar{\gamma}(x_k) + \eta_k \frac{g_k^T s_{k-1}}{s_{k-1}^T s_{k-1}},$$

where

$$\bar{\gamma}(x_k) = \frac{4(f(x_{k-1}) - f(x_k)) + 3g_k^T s_{k-1} + g_{k-1}^T s_{k-1}}{s_{k-1}^T s_{k-1}},$$

and

$$\eta_k = \begin{cases} \bar{\gamma}(x_k) + \dfrac{\delta}{g_k^T s_{k-1}}, & \bar{\gamma}(x_k) < 0, \\ 0, & \bar{\gamma}(x_k) \geq 0. \end{cases}$$

In another guideline, based on scalar approximations for the tensor of the third derivatives, efforts have been made to achieve simple cubic approximations for the objective function in the neighborhood of x_k which can be used in the TR subproblem. In this context, Nesterov and Polyak [13] proposed the following local approximation of the objective function:

$$f(x_k + s) \approx f(x_k) + g_k^T s + \frac{1}{2} s^T \nabla^2 f(x_k) s + \frac{1}{6} M ||s||^3, \tag{1.6}$$

where M is a positive parameter, using a cubic regularization of the Newton's method. Subsequently, Nesterov [12] dealt with more sophisticated schemes to improve the complexity bounds of the approach of [13] for the convex problems. Also, Cartis *et al.* [6] suggested the following adaptive version of (1.6):

$$f(x_k + s) \approx f(x_k) + g_k^T s + \frac{1}{2} s^T B_k s + \frac{1}{3} \sigma_k ||s||^3,$$

in which σ_k is a positive parameter.

Here, we suggest another scalar approximation for the matrix B_k based on a two-point approximation of the modified secant equation proposed by Li and Fukushima [10] (see also [11]), to be used in the quadratic TR subproblem. This work is organized as follows. In Section 2, we describe our approach for computing the scalar $\gamma(x_k)$ in (1.4). Then, using the corresponding simple quadratic model, we deal with an adaptive nonmonotone TR algorithm as well as 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. A scalar approximation for the Hessian in the trust region subproblem using a two-point approximation of a modified secant equation

In this section, at first we briefly discuss secant equations and then, we propose a scalar approximation for the matrix B_k in (1.2).

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 [18]. Iterative formula of the methods is given by

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

where α_{k-1} is a step length to be computed by performing a line search along the search direction d_{k-1} which is calculated by solving the following system of linear equations:

$$B_{k-1} d_{k-1} = -g_{k-1}.$$

The methods are characterized by the fact that B_{k-1} is effectively updated to achieve a new matrix B_k as an approximation of $\nabla^2 f(x_k)$ (often) satisfying the standard secant equation [18], *i.e.*

$$B_k s_{k-1} = y_{k-1}, \tag{2.1}$$

where $y_{k-1} = g_k - g_{k-1}$.

In spite of computational efficiency of the quasi-Newton methods, as an important defect the methods need to compute the matrix $B_k \in \mathbb{R}^{n\times n}$ in each iteration, being improper for solving large-scale problems. Although memoryless quasi-Newton methods have been developed to overcome the mentioned defect [2, 3], in another approach scalar approximations have been proposed for the matrix B_k using a two-point approximation of the secant equation (2.1) [5]. More exactly, $B_k \approx \gamma(x_k)I$ in which $\gamma(x_k)$ is computed as a solution of the following least–squares problem:

$$\min_{\gamma>0} ||\gamma s_{k-1} - y_{k-1}||,$$

which yields

$$\gamma(x_k) = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}\cdot \tag{2.2}$$

Although being positive for strictly convex objective functions, the formula (2.2) generally fails to yield a positive value for $\gamma(x_k)$. Hence, we employ the modified secant equation proposed by Li and Fukushima [10] which is appropriate even for nonconvex objective functions; that is

$$B_k s_{k-1} = \bar{y}_{k-1},$$

in which

$$\bar{y}_{k-1} = y_{k-1} + h_{k-1}||g_{k-1}||^r s_{k-1},$$

where $r > 0$, and $h_{k-1} > 0$ is defined by

$$h_{k-1} = C + \max\{-\frac{s_{k-1}^T y_{k-1}}{||s_{k-1}||^2}, 0\}||g_{k-1}||^{-r},$$

with some positive constant C (see also [23]). More exactly, here we solve the following least-squares problem to obtain $\gamma(x_k)$:

$$\min_{\gamma>0} ||\gamma s_{k-1} - \bar{y}_{k-1}||,$$

which yields

$$\gamma(x_k) = \frac{s_{k-1}^T \bar{y}_{k-1}}{s_{k-1}^T s_{k-1}}\cdot \tag{2.3}$$

It is worth noting that without convexity assumption on the objective function, $\gamma(x_k)$ given by (2.3) is positive. Also, in computational point of view, (2.3) presents a low cost formula for computing $\gamma(x_k)$ in contrast to the analogous formulas discussed in Section 1. Thus, it can be effectively used in the quadratic model (1.4).

3. An adaptive nonmonotone trust region algorithm

Here, using (2.3) we propose a nonmonotone TR algorithm in which a simple quadratic model is employed. Then, we briefly discuss its convergence properties.

To describe a general iteration of our algorithm, firstly we define the following simple quadratic model:

$$\begin{array}{ll} \min\limits_{s\in\mathbb{R}^n} & m_k(s) = g_k^T s + \frac{1}{2}\gamma(x_k)s^T s, \\ s.t. & ||s|| \le \Delta_k, \end{array} \tag{3.1}$$

where

$$\gamma(x_k) = \begin{cases} \frac{s_{k-1}^T \bar{y}_{k-1}}{s_{k-1}^T s_{k-1}}, & k > 0, \\ 1, & k = 0. \end{cases} \tag{3.2}$$

Then, based on the approach of [20], we introduce the following adaptive choice for the TR radius:

$$\Delta_k = t^p \frac{||g_k||}{\gamma(x_k)}, \tag{3.3}$$

where $t \in (0,1)$ is a constant and p is a nonnegative integer. Afterwards, by setting $p = 0$ we solve (3.1) to find the trial step s_k. In order to measure consistency between the exact and the approximate models, similar to [21] we apply the effective nonmonotone version of the TR ratio (1.3) proposed by Toint [19], *i.e.*

$$\hat{\rho}_k = \frac{f_{l(k)} - f(x_k + s_k)}{m_k(0) - m_k(s_k)}, \tag{3.4}$$

in which

$$f_{l(k)} = \max_{0 \le j \le q(k)} \{f_{k-j}\}, \tag{3.5}$$

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 $N \in \mathbb{N}$. If $\hat{\rho}_k$ is large enough, then we set $x_{k+1} = x_k + s_k$; otherwise, we set $p = p+1$ to resolve the subproblem (3.1) with a smaller radius. Considering these preliminaries, now we are in a position to describe our algorithm in details.

Algorithm 3.1. An adaptive nonmonotone trust region algorithm with a simple quadratic model (ANTRSQM).

Step 0: {Initialization} Choose an initial point $x_0 \in \mathbb{R}^n$ and the positive constants $t \in (0,1)$, $\mu \in (0,1)$, $\theta \in (0,1)$, C, r, N, and ϵ. Compute f_0 and set $\gamma(x_0) = 1$ and $k = 0$.
Step 1: **If** $||g_k|| < \epsilon$, **then stop**; **else** set $p = 0$.
Step 2: Compute Δ_k by (3.3) and solve the subproblem (3.1) to find the trial step s_k.
Step 3: Compute $\hat{\rho}_k$ by (3.4). **If** $\hat{\rho}_k \ge \mu$, **then** set $x_{k+1} = x_k + s_k$ and **goto** Step 5.
Step 4: Set $p = p + 1$ and **goto** Step 2.
Step 5: Compute $\gamma(x_{k+1})$ by (3.2) and set

$$\gamma(x_{k+1}) = \min\left\{\frac{1}{\theta}, \max\{\gamma(x_{k+1}), \theta\}\right\}. \tag{3.6}$$

Step 6: Set $k = k + 1$ and **goto** Step 1.

Note that the subproblem (3.1) can be solved by a reasonable strategy as described in [17]. More exactly, the trial step s_k can be simply computed by

$$s_k = -\frac{t^p}{\gamma(x_k)} g_k. \tag{3.7}$$

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 (3.1), then*

$$m_k(0) - m_k(s_k) \ge \frac{t^p}{2\gamma(x_k)} ||g_k||^2. \tag{3.8}$$

Proof. From (3.7), we get

$$\begin{aligned} m_k(0) - m_k(s_k) &= m_k(0) - m_k(-\frac{t^p}{\gamma(x_k)} g_k) \\ &= \frac{t^p}{\gamma(x_k)}||g_k||^2 - \frac{t^{2p}}{2\gamma(x_k)}||g_k||^2 \ge \frac{t^p}{2\gamma(x_k)}||g_k||^2. \end{aligned}$$

□

Lemma 3.3. *Steps 2–4 of Algorithm 3.1 terminate finitely.*

Proof. Firstly, we prove that if p is sufficiently large, then $\rho_k \geq \mu$. In this context, let s_k^p be the solution of (3.1) corresponding to an integer value of p. We can write

$$m_k(s_k^p) = g_k^T s_k^p + \frac{1}{2}\gamma(x_k)||s_k^p||^2 \leq m_k(0) = 0.$$

Now, at the contrary suppose that there exists an iteration k so that for all nonnegative integers $p \geq 0$, we have

$$\frac{f_k - f(x_k + s_k^p)}{m_k(0) - m_k(s_k^p)} < \mu.$$

So, from Taylor expansion, for some $\xi_k \in (0,1)$ we have

$$\begin{aligned} -g_k^T s_k^p - \frac{1}{2} s_k^{pT} \nabla^2 f(x_k + \xi_k s_k^p) s_k^p &= f(x_k) - f(x_k + s_k^p) \\ &< \mu \left(-g_k^T s_k^p - \frac{1}{2}\gamma(x_k)||s_k^p||^2 \right). \end{aligned}$$

Also, since $\gamma(x_k)$ defined by (3.6) is bounded and the trial step s_k is computed by (3.7), we have

$$(1-\mu)\frac{t^p}{\gamma(x_k)}||g_k||^2 < \frac{t^{2p}}{2\gamma(x_k)^2}\left(g_k^T \nabla^2 f(x_k + \xi_k s_k^p) g_k - \mu\gamma(x_k)||g_k||^2\right),$$

which yields

$$(1-\mu)||g_k||^2 < \frac{t^p}{2\gamma(x_k)}\left(g_k^T \nabla^2 f(x_k + \xi_k s_k^p) g_k - \mu\gamma(x_k)||g_k||^2\right).$$

Now, if $p \longrightarrow \infty$, then right-hand side of the preceding inequality tends to zero and consequently, we get $||g_k|| < 0$ which is a contradiction. So, for sufficiently large values of p we have $\rho_k \geq \mu$. Now, using (3.5), we get

$$\hat{\rho}_k = \frac{f_{l(k)} - f(x_k + s_k)}{m_k(0) - m_k(s_k)} \geq \frac{f(x_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)} \geq \mu,$$

which completes the proof. □

Here, we discuss global convergence of Algorithm 3.1. In this context, the following standard assumptions are needed.

Assumption 3.4. The objective function f is bounded below and the level set $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n | f(x) \leq f_0\}$ is bounded.

Lemma 3.5. *For the sequence $\{x_k\}_{k\geq 0}$ generated by Algorithm 3.1, if Assumptions 3.4 hold, then the sequence $\{f_{l(k)}\}_{k\geq 0}$ is decreasing and $\lim_{k\to\infty} f(x_{l(k)}) = \lim_{k\to\infty} f(x_k)$.*

Proof. The proof is similar to the proofs of Lemmas 2.5 and 3.2 of [1] and here is omitted. □

Theorem 3.6. *Suppose that Assumptions 3.4 hold. For the sequence $\{x_k\}_{k\geq 0}$ generated by Algorithm 3.1, if $||g_k|| \neq 0$, $\forall k \geq 0$, then*

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

Proof. When $||g_k|| \neq 0$, for all $k \geq 0$, Lemma 3.3 ensures that the algorithm generates an infinite sequence $\{x_k\}_{k\geq 0}$ satisfying $\hat{\rho}_k \geq \mu$. So, from (3.4) and (3.8), we obtain

$$f_{l(k)} - f(x_k + s_k) \geq \mu(m_k(0) - m_k(s_k)) \geq \frac{t^p}{2\gamma(x_k)}\mu||g_k||^2.$$

Furthermore, Step 5 of Algorithm 3.1 implies that $\theta \leq \gamma(x_k) \leq \theta^{-1}$, for all $k \geq 0$. Also, considering Lemma 3.3, p is finite in each iteration. So, from Lemma 3.5, the proof is complete. □

Theorem 3.7. *Suppose that Assumptions 3.4 hold and the sequence $\{x_k\}_{k\geq 0}$ generated by Algorithm 3.1 converges to the optimal solution x^* for which the Hessian $\nabla^2 f(x^*)$ is positive definite. Moreover, assume that $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of x^*. If*

$$\lim_{k\to\infty} \frac{||(\nabla^2 f(x) - \gamma(x_k)I)s_k||}{||s_k||} = 0,$$

and

$$\frac{||\gamma(x_k)s_k + g_k||}{||g_k||} \leq \xi_k,$$

for a positive sequence $\{\xi_k\}_{k\geq 0}$ converging to zero, then the sequence $\{x_k\}_{k\geq 0}$ converges to x^* superlinearly.

Proof. The proof is similar to the proof of Theorem 4.1 of [22] 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 ANTRSQM method (Algorithm 3.1), the TR method SLN with the quadratic simple subproblem given in [17], and an adaptive version of the nonmonotone TR method proposed in [21] in which the radius is computed by

$$\varDelta_k = \frac{t^p}{\gamma(x_k)}||g_k||,\ \ k = 0, 1, \ldots,$$

with $\gamma(x_k)$ given by (1.5), here called ASNTR. Also, we have compared ANTRSQM with the TR methods in which the cubic models of [6, 12, 13] have been employed in the subproblems; the corresponding methods are respectively called ARC, ACRN and CRN.

For the ASNTR, SLN and ARC algorithms we adopted the parameter values suggested in [6, 17, 21], respectively. For ANTRSQM, based on the suggestions of [21] we set $\mu = 0.1$ and $t = 0.5$. The other parameters have been set to be equal to $\theta = 10^{-20}$, $C = 10^{-3}$, $N = 10$, $r = 1$ if $||g_k|| \geq 1$, and $r = 3$, otherwise [23]. Also, for ACRN and CRN we adopted the parameter values of ANTRSQM and to update the TR radius in both of the methods, we applied the approach of [20]. We used the following scaled memoryless DFP approximation of the Hessian [18] for ARC, ACRN and CRN:

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

with $\theta_k^B = \dfrac{s_k^T y_k}{||s_k||^2}$ (see also [4, 14, 15]). Note that we set $B_{k+1} = B_k$ whenever the curvature condition (*i.e.* $s_k^T y_k > 0$) did not hold. All attempts for finding an approximation of the solution were terminated by reaching maximum of 20 000 iterations or achieving a solution with $||g_k||_\infty < 10^{-6}(1 + |f(x_k)|)$.

Table 1. Test problems data.

Function	n	Function	n	Function	n
ARWHEAD	5000	ENGVAL1	5000	FLETCHCR	1000
ARGLINA	200	EXTROSNB	1000	FMINSRF2	5625
BDEXP	5000	GENROSE	500	FMINSURF	5625
BDQRTIC	5000	LIARWHD	5000	FREUROTH	5000
BIGGSB1	5000	NONDIA	5000	MANCINO	100
BOX	10 000	PENALTY1	1000	MOREBV	5000
BROWNAL	200	TQUARTIC	5000	MSQRTBLS	1024
BROYDN7D	5000	SPARSQUR	10 000	NCB20	5010
BRYBND	5000	SPMSRTLS	4999	NCB20B	5000
COSINE	10 000	SROSENBR	5000	NONDQUAR	5000
DIXMAANA	3000	TOINTGSS	5000	POWELLSG	5000
DIXMAANB	3000	TRIDIA	5000	POWER	10 000
DIXMAANC	3000	VARDIM	200	ARGLINB	200
DIXMAAND	3000	VAREIGVL	50	PENALTY3	200
DIXMAANE	3000	WOODS	4000	CHAINWOO	4000
DIXMAANF	3000	CHNROSNB	50	BA-L1LS	57
DIXMAANG	3000	CRAGGLVY	5000	CHNROSNB	50
DIXMAANH	3000	TOINTPSP	50	CHNRSNBM	50
DIXMAANI	3000	TOINTQOR	50	FLETBV3M	5000
DIXMAANJ	3000	TOINTGOR	50	FLETCBV3	5000
DIXMAANK	3000	DECONVU	63	FLETCHBV	5000
DIXMAANL	3000	DIXON3DQ	10 000	INDEF	5000
DQDRTIC	5000	EDENSCH	2000	SBRYBND	5000

The experiments were performed on a set of CUTEr unconstrained optimization test problems [8] with the dimensions $n \in [50, 10^4]$, based on the considerations of [9]. More exactly, we considered all the test problems that are twice continuously differentiable and then deleted a problem in any of the following cases:

(D1) the problem was small (dimension less than 50);
(D2) the problem could be solved in, at most, 0.001 S by any of the solvers;
(D3) the cost function seemed to have no lower bound;
(D4) the cost function generated a 'NaN' for what seemed to be a reasonable choice for the input;
(D5) the problem could not be solved by any of the solvers;
(D6) different solvers converged to different local minimizers (that is, the optimal costs were different).

Therefore, 110 and 7 test problems were deleted by (D1) and (D2), respectively. Moreover, 10 and 12 test problems were discarded by (D4) and (D5), respectively. After the deletion process, we were left with 69 problems, as specified in Table 1.

Efficiency comparisons were drawn using the Dolan-Moré performance profile [7] 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 [9]. 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 1–4 demonstrate the results of comparisons. As Figures 1 and 2 show, ANTRSQM outperforms ASNTR and SLN with respect to the total number of function and gradient evaluations, and the running time. Also, as shown by Figures 3 and 4, ANTRSQM is generally preferable to ARC and CRN. Furthermore, although the figures show that ANTRSQM and ACRN are approximately competitive and at times ACRN is preferable with respect to the total number of function and gradient evaluations, ANTRSQM outperforms ACRN with

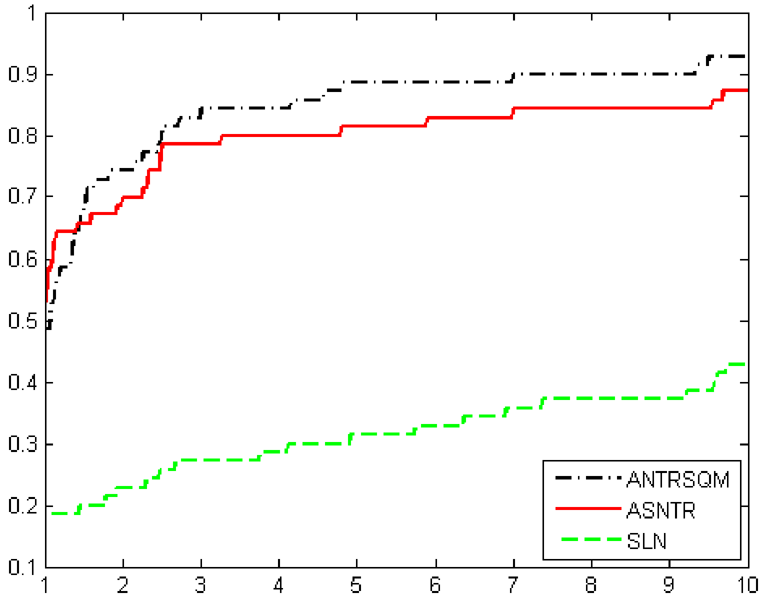

FIGURE 1. Total number of function and gradient evaluations performance profiles for ANTRSQM, ASNTR and SLN.

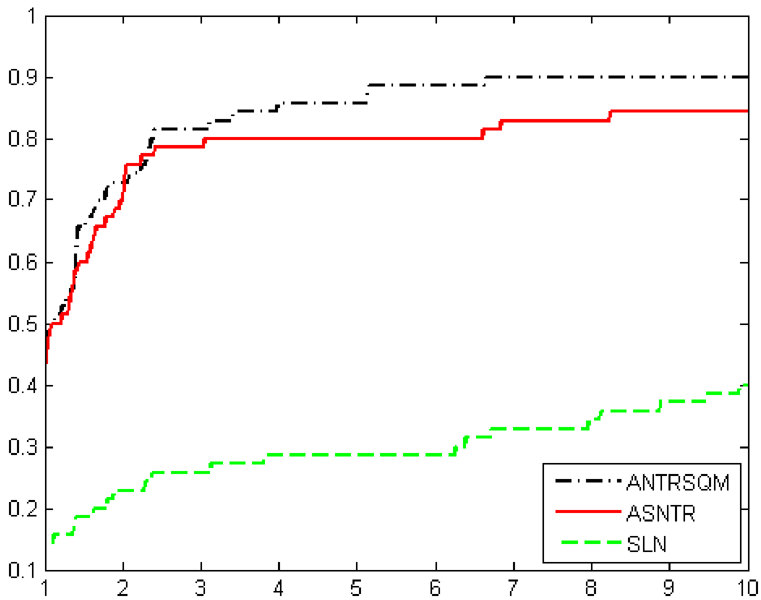

FIGURE 2. CPU time performance profiles for ANTRSQM, ASNTR and SLN.

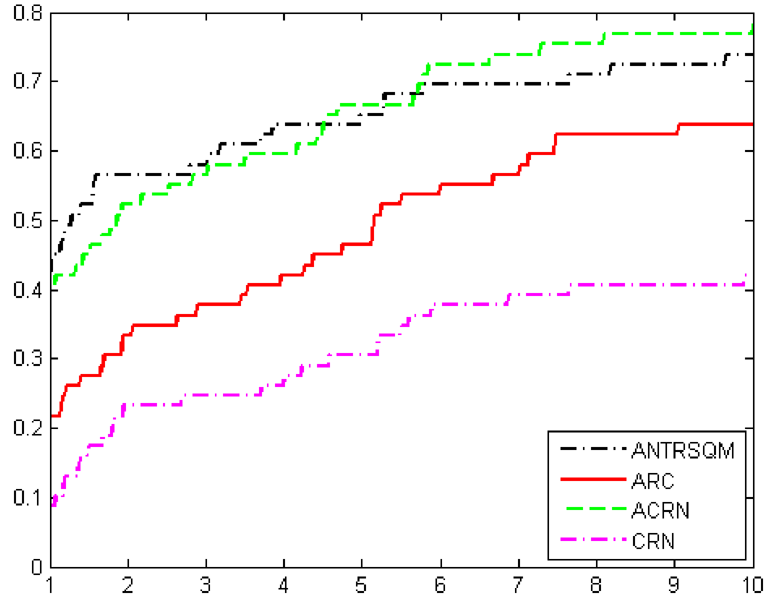

FIGURE 3. Total number of function and gradient evaluations performance profiles for ANTRSQM, ARC, ACRN and CRN.

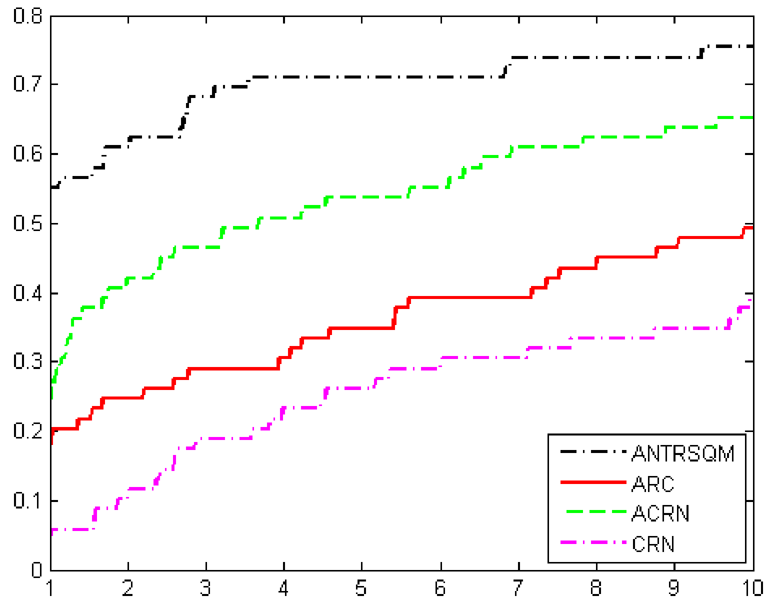

FIGURE 4. CPU time performance profiles for ANTRSQM, ARC, ACRN and CRN.

respect to the running time. This seems reasonable because the TR subproblem of ANTRSQM is solved simpler than the TR subproblem of ACRN.

5. Conclusions

Employing a modified scalar approximation of the Hessian in trust region subproblem, a nonmonotone adaptive trust region method has been developed. It is worth noting that the trust region subproblem of the algorithm is solved more simply in contrast to the many other trust region methods proposed in the literature. The method has been shown to be globally and locally superlinearly convergent. Preliminary numerical experiments showed that our scalar approximation seem to be practically effective.

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

References

[1] M. Ahookhoosh and K. Amini, A nonmonotone trust region method with adaptive radius for unconstrained optimization problems. *Comput. Math. Appl.* **60** (2010) 411–442.
[2] N. Andrei, Accelerated scaled memoryless BFGS preconditioned conjugate gradient algorithm for unconstrained optimization. *Eur. J. Oper. Res.* **204** (2010) 410–420.
[3] S. Babaie-Kafaki, A modified scaled memoryless BFGS preconditioned conjugate gradient method for unconstrained optimization. *4OR* **11** (2013) 361–374.
[4] S. Babaie-Kafaki, On optimality of the parameters of self-scaling memoryless quasi-Newton updating formulae. *J. Optim. Theory Appl.* **167** (2015) 91–101.
[5] J. Barzilai and J.M. Borwein, Two-point stepsize gradient methods. *IMA J. Numer. Anal.* **8** (1988) 141–148.
[6] C. Cartis, N.I.M. Gould and Ph.L. Toint, Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Math. Program.* **127** (2011) 245–295.
[7] E.D. Dolan and J.J. Moré, Benchmarking optimization software with performance profiles. *Math. Program.* **9** (2002) 201–213.
[8] 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.
[9] 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.
[10] D.H. Li and M. Fukushima, A modified BFGS method and its global convergence in nonconvex minimization. *J. Comput. Appl. Math.* **129** (2001) 15–35.
[11] D.H. Li and M. Fukushima, On the global convergence of the BFGS method for nonconvex unconstrained optimization problems. *SIAM J. Optim.* **11** (2001) 1054–1064.
[12] Y. Nesterov, Accelerating the cubic regularization of Newton's method on convex problems. *Math. Program.* **112** (2008) 159–181.
[13] Y. Nesterov and B.T. Polyak, Cubic regularization of Newton's method and its global performance. *Math. Program.* **108** (2006) 177–205.
[14] 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.
[15] S.S. Oren and E. Spedicato, Optimal conditioning of self-scaling variable metric algorithms. *Math. Program.* **10** (1976) 70–90.
[16] 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** (2015) 95–117.
[17] Z. Sang and Q. Sun, A self-adaptive trust region method with line search based on a simple subproblem model. *J. Comput. Appl. Math.* **232** (2009) 514–522.
[18] W. Sun and Y.X. Yuan. Optimization Theory and Methods: Nonlinear Programming. Springer, New York (2006).
[19] Ph.L. Toint, Nonmonotone trust region algorithms for nonlinear optimization subject to convex constraints. *SIAM J. Sci. Comput.* **17** (1996) 725–739.
[20] 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.
[21] Q. Zhou, J. Chen and Z. Xie, A nonmonotone trust region method based on simple quadratic models. *J. Comput. Appl. Math.* **272** (2014) 107–115.
[22] Q. Zhou and C. Zhang, A new nonmonotone adaptive trust region method based on simple quadratic models. *J. Appl. Math. Comput.* **40** (2012) 111–123.
[23] W. Zhou and L. Zhang, A nonlinear conjugate gradient method based on the MBFGS secant condition. *Optim. Methods Softw.* **21** (2006) 707–714.
[24] M. Zhu, J.L. Nazareth and H. Wolkowicz, The quasi-Cauchy relation and diagonal updating. *SIAM J. Optim.* **9** (1999) 1192–1204.