4OR-Q J Oper Res
DOI 10.1007/s10288-016-0305-3

RESEARCH PAPER

A line search trust-region algorithm with nonmonotone adaptive radius for a system of nonlinear equations

Keyvan Amini[1] · Mushtak A. K. Shiker[1,2] · Morteza Kimiaei[3]

Received: 10 January 2015 / Revised: 27 November 2015

Abstract In this paper, a trust-region procedure is proposed for the solution of nonlinear equations. The proposed approach takes advantages of an effective adaptive trust-region radius and a nonmonotone strategy by combining both of them appropriately. It is believed that selecting an appropriate adaptive radius based on a suitable nonmonotone strategy can improve the efficiency and robustness of the trust-region frameworks as well as decrease the computational cost of the algorithm by decreasing the required number subproblems that must be solved. The global convergence and the local Q-quadratic convergence rate of the proposed approach are proved. Preliminary numerical results of the proposed algorithm are also reported which indicate the promising behavior of the new procedure for solving the nonlinear system.

Keywords Nonlinear equations · Trust-region · Adaptive radius · Nonmonotone technique · Armijo-type line search

Mathematics Subject Classification 90C30 · 93E24 · 34A34

✉ Keyvan Amini
kamini@razi.ac.ir

Mushtak A. K. Shiker
mmttmmhh@yahoo.com

Morteza Kimiaei
morteza.kimiaei@gmail.com

[1] Department of Mathematics, Faculty of Science, Razi University, Kermanshah, Iran

[2] Department of Mathematics, University of Babylon, Babylon, Iraq

[3] Department of Mathematics, Asadabad Branch, Islamic Azad University, Asadabad, Iran

Published online: 27 January 2016

1 Introduction

In this paper we consider the nonlinear system of equations

$$F(x) = 0, \quad x \in \mathbf{R}^n, \tag{1}$$

where $F : \mathbf{R}^n \to \mathbf{R}^n$ is a continuously differentiable mapping in the form $F(x) := (F_1(x), F_2(x), \ldots, F_n(x))^T$. Suppose that $F(x)$ has a zero. Then, it is well known that every solution x^* of the given problem (1) is also a solution of the following nonlinear least-squares problem

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}\|F(x)\|^2 \\ \text{s.t.} & x \in \mathbf{R}^n, \end{array} \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm. Conversely, if x^* is a minimum of (2) and $f(x^*) = 0$, then x^* solves (1). The trust-region framework for solving system of nonlinear equations (1) is a popular class of iterative procedures that generates a trial step d_k, in each iterate, by solving the following subproblem

$$\begin{array}{ll} \min & m_k(x_k + d) := \frac{1}{2}\|F_k + J_k d\|^2 = f_k + d^T g_k + \frac{1}{2} d^T J_k^T J_k d \\ \text{s.t.} & d \in \mathbf{R}^n \text{ and } \|d\| \leq \Delta_k, \end{array} \tag{3}$$

where $f_k := f(x_k)$, $F_k := F(x_k)$, $J_k := F'(x_k)$, Jacobian of $F(x)$ at x_k, $g_k := J_k^T F_k$ and $\Delta_k > 0$ is the trust-region radius. The ratio r_k of the actual reduction to the predicted reduction is defined by

$$r_k := \frac{f(x_k) - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)}. \tag{4}$$

Obviously, it can be concluded that the model will have a good agreement with the original problem at the current iterate x_k whenever r_k is sufficiently close to 1. If r_k is greater than a positive constant μ, the trial step d_k will be accepted, leading to $x_{k+1} := x_k + d_k$, and the trust-region radius can be expanded or kept the same. Otherwise, the trust-region radius must be diminished and the subproblem (3) will be solved again to possibly find an acceptable trial point in the sequel of the process Nocedal and Wright (2006).

To modify the trust-region methods, many important techniques are presented that can improve the efficiency of the trust-region methods. A basic technique for improving the trust-region methods, namely the line search, has been developed in order to prevent resolving the trust-region subproblem when the current trial step is rejected. In their simplest form, line-search methods produce each iterate by searching for an acceptable value of x along a line passing through the previous iterate. Toint (1982) presents a linear search to find a lower value of the objective function at every iterate, but it does not impose a sufficient decrease condition on the line search. In addition, Nocedal and Yuan (1998) used a backtracking line search when the new value of the objective function is less than the previous value, however they didn't impose a sufficient decrease condition on this line search. Gertz (1999) developed a trust-region method with a monotone backtracking line search. The monotone backtracking method employs a sufficient decrease criterion at every iteration and has the appropriate convergence properties but has some drawbacks. Some researchers showed that utilizing monotone techniques may lead to decreasing the rate of convergence and increasing the possibility of finding the global optimum (Ahookhosh et al. 2012; Fasano et al. 2006; Grippo et al. 1986, 1989; Grippo and Sciandrone 2007; La Cruz and Raydan 2003; Zhang and Hager 2004. In order to avoid these drawbacks of the Armijo-type line search globalization techniques, the first nonmonotone strategy was introduced by Grippo et al. (1986) for unconstrained optimization problems. In particular, they changed the Armijo rule to accept the steplength α_k by

$$f(x_k + \alpha_k d_k) \le f_{l(k)} + \delta \alpha_k g_k^T d_k, \tag{5}$$

where $\delta \in (0, 1)$ and

$$f_{l(k)} = \max_{0 \le j \le n(k)} \{f_{k-j}\}, \quad k \in \mathbf{N} \cup \{0\}, \tag{6}$$

in which $n(0) := 0$ and $0 \le n(k) \le \min\{n(k-1)+1, N\}$ with $N \ge 0$. Recently, Ahookhosh et al. (2012), introduced a new nonmonotone backtracking strategy for unconstrained optimization and exploited it into a modified trust-region framework in order to prevent resolving the trust-region subproblem. They used advantages of new nonmonotone line search with low computational cost whenever the trial step d_k is rejected.

Another approach to improve the trust-region methods is the adaptive radius that can prevent increasing and decreasing the radius by controlling the size of the trust-region radius. If the trust-region radius Δ_k is very large, then the number of subproblems will be increased and so computational costs to solve the problem may be increased, too. On the other hand, if Δ_k is very small, then the total number of iterations is increased and efficiency of the method will be possibly reduced. Sartenaer (1997) developed an elaborate strategy that can automatically determine an initial trust-region radius. The basic idea is to determine a maximal initial radius through many repeated trials in the steepest descent direction in order to guarantee a sufficient agreement between the model and the objective function. Zhang et al. (2002) proposed another adjustable strategy to determine the trust-region radius based on information of g_k and Hessian matrix B_k in current iteration. They introduced the following adaptive formula

$$\Delta_{k+1} = \begin{cases} c\Delta_k & \text{if } r_k < \mu, \\ \|g_k\| \|\widehat{B}_k^{-1}\| & \text{if } r_k \ge \mu, \end{cases}$$

in which $0 < c < 1$ and $\widehat{B}_k := B_k + iI$ is a positive definite matrix for some $i \in \mathbf{N}$ and I is identity matrix. Zhang and Wang (2003) proposed an adaptive radius technique for solving the system of nonlinear equations. This method updates the radius of trust-region as follows

$$\Delta_{k+1} = \begin{cases} c\Delta_k & \text{if } r_k < \mu, \\ \|F_k\|^{\delta} & \text{if } r_k \ge \mu, \end{cases} \tag{7}$$

where $0 < c < 1$ and $0.5 < \delta < 1$ are constants. Although this method can somehow prevent the trust-region radius staying too large, it has the following disadvantages

- The sequence generated by this method is superlinearly convergent with the convergence order 2δ.
- The efficiency of the numerical results is largely dependent on the choice of δ.
- This method can not prevent adequately generating the intensely small trust-region radius.

Another interesting work on the trust-region radius to overcome some drawbacks (7) was proposed by Fan and Pan (2010). They introduced the following adaptive trust-region radius

$$\Delta_{k+1} = \begin{cases} c\Delta_k & \text{if } r_k < \mu, \\ M\|F_k\| & \text{if } r_k \geq \mu, \end{cases} \tag{8}$$

with a constant M and $c \in (0, 1)$. The two disadvantages of the method (7) can be overcome almost by the adaptive radius (8). But, when $\|F_k\|$ is large, because of the selection of large constant M, the radius is very large and will increase the total number of solving subproblem. Therefore, the adaptive radius (8) does not sufficiently create a suitable agreement between the constant M and $\|F_k\|$. Consequently, they have a high computational cost.

In this paper, we introduce a new adaptive radius strategy based on the nonmonotone Grippo et al. (1986) which we can overcome some disadvantages of (7) and (8). One of the most interesting advantages of this method is an attractive relation between the nonmonotone line search and the adaptive radius that can increase the efficiency of the method. The global convergence to first-order critical points together with superlinear and quadratic convergence are investigated. The preliminary numerical results exhibit the efficiency and the robustness of the proposed method for solving the system of nonlinear equations.

The rest of this paper is organized as follows. In Sect. 2, a new adaptive trust-region radius is described and then the new trust-region algorithm will be introduced. In Sect. 3, the global convergence and the quadratic convergence of the new algorithm under some suitable assumptions are investigated. Preliminary numerical results are reported in Sect. 4. Finally, some conclusions are outlined in Sect. 5.

2 Motivation and algorithmic structure

A trust-region-based algorithm for solving a system of nonlinear equations will be introduced in this section. After proposing a nonmonotone adaptive trust-region radius and establishing a nonmonotone line search approach, we incorporate these strategies into trust-region framework to construct a more effective procedure for solving the nonlinear system.

If iterate is unsuccessful, the trust-region radius is produced based on the obtained information of nonmonotone line search technique (α_k) that will lead to decrease the total number of function evaluations. Also, if iterate is successful or very successful, then the radius of trust-region takes advantages from nonmonotone technique to control the size of radius trust-region. Consider the step acceptance constants $0 < \mu_1 < \mu_2 <$

1 and the trust-region scaling parameters $0 < \eta_1 < 1 \leq \eta_2$. Using the relation (6), we define the new adaptive radius by

$$\Delta_{k+1} := \begin{cases} \eta_1 \alpha_k \Delta_k & \text{if } r_k < \mu_1, \\ NF_{l(k+1)} & \text{if } \mu_1 \leq r_k < \mu_2, \\ \eta_2 NF_{l(k+1)} & \text{if } r_k \geq \mu_2, \end{cases} \tag{9}$$

in which

$$NF_{l(k)} := \max_{0 \leq j \leq n(k)} \{\|F_{k-j}\|\}, \quad k \in \mathbf{N} \cup \{0\}, \tag{10}$$

in which $n(0) := 0$ and $0 \leq n(k) \leq \min\{n(k-1)+1, N\}$ with $N \geq 0$. Because, the elements of the new sequence generated by $\{NF_{l(k)}\}_{k \geq 0}$ are always larger than the elements of $\{\|F_k\|\}_{k \geq 0}$, the trust-region radius cannot become too small as possible whenever iterates are not near the optimum. On the other hand, this sequence is decrease and so it prevents the radius of trust-region staying too large whenever iterates are not far away from the optimum. If iteration is very successful, we increase the radius appropriately in a scale of $\{NF_{l(k)}\}_{k \geq 0}$ and so we find the optimum within the greater region that will decrease the total number of iterations. Let x_k denote the current iterate. The first step in our algorithm is to compute a trial step d_k and then obtain the ratio (4). If $r_k \geq \mu_1$, then we accept the trial step and set $x_{k+1} := x_k + d_k$. Otherwise, for computing α_k, we use an approximately nonmonotone line search technique (5) until the trust-region step reduces the objective function value and set $x_{k+1} := x_k + \alpha_k d_k$.

3 Convergence theory

In this section, we will investigate the global and the quadratic convergence results of the proposed algorithm given in Sect. 2.

To verify the convergence analysis of the proposed algorithm, the following assumptions are required:

(H1) The level set $L(x_0) := \{x \in \mathbf{R}^n \mid f(x) \leq f(x_0)\}$ is bounded for the initial point $x_0 \in \mathbf{R}^n$ and $F(x)$ is continuously differentiable on a compact convex set Ω containing the level set $L(x_0)$.

(H2) The matrix $J(x)$ is bounded on Ω, i.e., there exist a constant $M_1 > 0$ such that

$$\|J(x)\| \leq M_1, \quad \forall x \in \Omega, \tag{11}$$

see Fan and Pan (2010), Li and Fukushima (2000a), Li and Fukushima (2000b), Yuan et al. (2011), Yuan (1998), Zhang and Wang (2003).

(H3) The matrix $J(x)$ is uniformly nonsingular on Ω, i.e., there exist a constant $M_0 > 0$ such that

$$M_0\|F(x)\| \leq \|J(x)^T F(x)\| = \|g(x)\|, \quad \forall x \in \Omega, \tag{12}$$

see Li and Fukushima (2000b).

Algorithm 1: LSTR (Line Search Trust-Region Algorithm)

Input: An initial point $x_0 \in \mathbf{R}^n$ and parameters $0 < \mu_1 < \mu_2 < 1$, $0 < \eta_1 < 1 \leq \eta_2$, $0 < \sigma_1 < \sigma_2 < 1$, $N > 0$ and $\epsilon > 0$;
Output: x_b, f_b;

1 **begin**
2 $J_0 := J(x_0)$; $\Delta_0 := NF_0$; $NF_0 := \|F_0\|$; $f_0 := 1/2\|F_0\|^2$; $F_0 := F(x_0)$; $n(0) := 0$; $k := 0$;
3 **while** $\|F_k\| \geq \epsilon$ **do**
4 compute d_k by solving the subproblem (3);
5 compute $F(x_k + d_k)$;
6 $f(x_k + d_k) := 1/2\ \|F(x_k + d_k)\|^2$;
7 determine r_k using (4);
8 **if** $r_k \geq \mu_1$ **then**
9 $x_{k+1} := x_k + d_k$;
10 **else**
11 set $\sigma \in [\sigma_1, \sigma_2]$;
12 $\alpha_k := 1$;
13 **while** $f(x_k + \alpha_k d_k) > f_{l(k)} + \gamma \alpha_k g_k^T d_k$ **do**
14 $\alpha_k := \sigma \alpha_k$;
15 compute $F(x_k + \alpha_k d_k)$;
16 $f(x_k + \alpha_k d_k) := 1/2\ \|F(x_k + \alpha_k d_k)\|^2$;
17 **end**
18 $x_{k+1} := x_k + \alpha_k d_k$;
19 **end**
20 $F_{k+1} := F(x_{k+1})$; $f_{k+1} := f(x_{k+1})$; $J_{k+1} := J(x_{k+1})$;
21 compute $n(k+1)$ and $NF_{l(k+1)}$ according with (10);
22 set $f_{l(k+1)} := 1/2 NF^2_{l(k+1)}$ and determine Δ_{k+1} using (9);
23 $k \leftarrow k + 1$;
24 **end**
25 $x_b := x_k$; $f_b := f_k$;
26 **end**

The cycle starting from Line 3 to Line 25 is called the outer cycle, and the cycle starting from Line 13 to Line 17 is called the backtraking loop. In addition, if $r_k \geq \mu_1$ (Line 8), it is called a successful iteration.

Remark 1 At each iteration, strong theoretical and numerical results for the proposed algorithm can be obtained if the step d_k satisfies

$$m_k(x_k) - m_k(x_k + d_k) \geq \beta\ \|g_k\|\ \min\left[\Delta_k, \frac{\|g_k\|}{\|J_k^T J_k\|}\right], \tag{13}$$

and

$$g_k^T d_k \leq -\beta \|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|J_k^T J_k\|}\right], \tag{14}$$

for all $k \in \mathbf{N} \cup \{0\}$ where $0 < \beta < 1$ is a constant. Similar to Ahookhosh et al. (2012), Nocedal and Yuan (1998), the trust-region subproblem can be solved such that (13) and (14) hold.

Remark 2 For $k \in I_1$, if $r_k \geq \mu_1$, then $f(x_k + d_k) \leq f_k$, so we can conclude that $x_k + d_k \in \Omega$. Otherwise, since the backtraking loop is well-defined, $f(x_k + \alpha_k d_k) \leq f_{l(k)}$, we have $x_k + \alpha_k d_k \in \Omega$. In both cases, according to (H2), $J(x)$ is uniformly bounded on segments $[x_k, x_k + d_k]$ and $[x_k, x_k + \alpha_k d_k]$, respectively, i.e., there exists a constant $M_1 > 0$ such that

$$d_k^T J^T(x) J(x) d_k \leq M_1^2 \|d_k\|^2, \quad \text{for all} \quad k \in \mathbf{N} \cup \{0\} \text{ and } x \in [x_k, x_k + \alpha_k d_k].$$

To start the convergence analysis, we define two index sets

$$\mathcal{I}_1 := \{k \mid r_k \geq \mu_1\} \text{ and } \mathcal{I}_2 := \{k \mid r_k < \mu_1\},$$

while $\mathcal{I}_1$ is the set of iterations that don't need line search and $\mathcal{I}_2$ is the set of iterations that need to use the line search.

Lemma 1 *Suppose that the sequence* $\{x_k\}$ *is generated by Algorithm* 1*. Then, for all* $k \in \mathbf{N} \cup \{0\}$*, we have* $x_k \in L(x_0)$ *and the sequences* $\{NF_{l(k)}\}$ *and* $\{f_{l(k)}\}$ *are decreasing and convergent.*

Proof Using the definition of $NF_{l(k)}$, we have

$$NF_{l(0)} = \|F_0\| \quad \text{and} \quad \|F_k\| \leq NF_{l(k)}.$$

By induction, we show that $x_k \in L(x_0)$, for all $k \in \mathbf{N}$. In that sense, we let $x_i \in L(x_0)$ for $i = 1, 2, \ldots, k$. To do so, we consider two cases.

(i) $k \in \mathcal{I}_1$. We have

$$\frac{NF_{l(k)}^2}{2} - \frac{\|F(x_k + d_k)\|^2}{2} \geq f_k - f_{k+1} \geq \mu_1(m_k(x_k) - m_k(x_k + d_k)) > 0,$$

so

$$\|F_{k+1}\| \leq NF_{l(k)} \leq \|F_0\|.$$

(ii) $k \in \mathcal{I}_2$. Using (6) and (14), we have

$$f(x_k + \alpha_k d_k) \leq f_{l(k)} + \gamma \alpha_k g_k^T d_k \leq f_{l(k)}.$$

This inequality, along with (10), shows that

$$\|F_{k+1}\| \leq NF_{l(k)}.$$

Thus, the sequence $\{x_k\}$ is contained in $L(x_0)$. Now, we prove that the sequence $\{NF_{l(k)}\}$ is a decreasing sequence. We divide the proof into two cases.

(i) $k \geq N$. In this case, $n(k) = N$, for all $k \geq N$. This fact that $\|F_{k+1}\| \leq NF_{l(k)}$ and the definition of $NF_{l(k)}$ result

$$\begin{aligned} NF_{l(k+1)} &= \max_{0\leq j\leq N}\{\|F_{k+1-j}\|\} \leq \max\left\{\max_{0\leq j\leq N}\{\|F_{k-j}\|\}, \|F_{k+1}\|\right\} \\ &= \max\{NF_{l(k)}, \|F_{k+1}\|\} = NF_{l(k)}. \end{aligned}$$

(ii) $k < N$. In this case, $n(k) = k$. Using an inductive approach, we can see that

$$NF_{l(k)} = F_0, \quad \forall k.$$

Both cases show that the sequence $\{NF_{l(k)}\}$ is a decreasing one. According to Assumption (H1) and $x_k \in L(x_0)$, then $\{NF_{l(k)}\}$ is convergent. Since $f_{l(k)} = 1/2NF^2_{l(k)}$, we can easily conclude that $\{f_{l(k)}\}$ is also convergent. □

Lemma 2 *Suppose that the sequence $\{x_k\}$ is generated by Algorithm 1 while d_k satisfies in (13) and (14). Then, the backtraking loop in Algorithm 1 is well-defined.*

Proof The proof is straight for $r_k \geq \mu_1$. So, let $r_k < \mu_1$. We show that the line search process terminates in the finite number of steps. By contradiction, assume that there exists $k \in \mathcal{I}_2$ such that

$$f(x_k + \sigma^i \alpha_k d_k) > f_{l(k)} + \gamma\sigma^i\alpha_k g_k^T d_k, \quad \forall i \in \mathbf{N} \cup \{0\}. \tag{15}$$

From (6), we have $f_k \leq f_{l(k)}$. This fact, along with (15), implies that

$$\frac{f(x_k + \sigma^i\alpha_k d_k) - f_k}{\sigma^i\alpha_k} > \gamma g_k^T d_k, \quad \forall i \in \mathbf{N} \cup \{0\}.$$

Since f is a differentiable function, by taking a limit, as $i \to \infty$, we obtain

$$g_k^T d_k \geq \gamma g_k^T d_k.$$

Using the fact that $\gamma \in (0, \frac{1}{2})$, this inequality leads us to $g_k^T d_k \geq 0$ which contradicts (14). □

Lemma 3 *Suppose that Assumptions (H2) and (H3) hold, the sequence $\{x_k\}$ is generated by Algorithm 1 and d_k is a solution of the subproblem (3). Then we have*

$$\beta\, \|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|J_k^T J_k\|}\right] \geq L_k \|F_k\|^2,$$

where $L_k := \beta M_0 \min\left\{\alpha_{k-1}\eta_1, \frac{M_0}{M_1^2}\right\}$.

Proof Using the fact that the $(k-1)$-th iteration is successful along with (9), we can see that

$$\Delta_k = \begin{cases} \alpha_{k-1}\eta_1 NF_{l(k-1)} & \text{if } \mu_1 \le r_{k-2} < \mu_2 \text{ and } r_{k-1} < \mu_1, \\ \alpha_{k-1}\eta_1\eta_2 NF_{l(k-1)} & \text{if } r_{k-2} \ge \mu_2 \text{ and } r_{k-1} < \mu_1, \\ NF_{l(k)} & \text{if } \mu_1 \le r_{k-1} < \mu_2, \\ \eta_2 NF_{l(k)} & \text{if } r_{k-1} \ge \mu_2. \end{cases} \tag{16}$$

This fact, along with Lemma 1, implies that

$$\Delta_k \ge \alpha_{k-1}\eta_1 NF_{l(k-1)}. \tag{17}$$

Using Assumptions (H2) and (H3), Remarks 1, 2 and the above inequality, we have

$$\begin{aligned} \beta\|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|J_k^T J_k\|}\right] &\ge \beta M_0\|F_k\| \min\left[\alpha_{k-1}\eta_1 NF_{l(k-1)}, \frac{M_0\|F_k\|}{M_1^2}\right] \\ &\ge \beta M_0\|F_k\| \min\left[\alpha_{k-1}\eta_1 NF_{l(k)}, \frac{M_0\|F_k\|}{M_1^2}\right] \\ &\ge \beta M_0\|F_k\| \min\left[\alpha_{k-1}\eta_1 \|F_k\|, \frac{M_0\|F_k\|}{M_1^2}\right] \\ &= \beta M_0 \min\left[\alpha_{k-1}\eta_1, \frac{M_0}{M_1^2}\right]\|F_k\|^2 \\ &= L_k\|F_k\|^2, \end{aligned}$$

for all $k \in \mathbf{N} \cup \{0\}$, where $L_k = \beta M_0 \min\left\{\alpha_{k-1}\eta_1, \frac{M_0}{M_1^2}\right\}$. Therefore, the proof is completed. □

Lemma 4 *Suppose that Assumptions (H2) and (H3) hold, the sequence $\{x_k\}$ is generated by Algorithm 1 and d_k is a solution of the subproblem (3). Then, we have*

$$g_k^T d_k \le -L_k\|F_k\|, \tag{18}$$

and

$$m_k(x_k) - m_k(x_k + d_k) \ge L_k\ \|F_k\|^2, \tag{19}$$

where $L_k := \beta M_0 \min\left\{\alpha_{k-1}\eta_1, \frac{M_0}{M_1^2}\right\}$.

Proof By Lemma 3 and relation (14), we can obtain

$$g_k^T d_k \le -\beta\|g_k\| \min\left[\Delta_k, \frac{\|g_k\|}{\|J_k^T J_k\|}\right] \le -L_k\|F_k\|^2$$

and

$$m_k(x_k) - m_k(x_k + d_k) \geq \beta \|g_k\| \min \left[\Delta_k, \frac{\|g_k\|}{\|J_k^T J_k\|} \right] \geq L_k \|F_k\|^2,$$

where $L_k = \beta M_0 \min \left\{ \alpha_{k-1} \eta_1, \frac{M_0}{M_1^2} \right\}$. □

Lemma 5 *Suppose that* $\{x_k\}$ *is generated by Algorithm 1 and there exists a positive constant* $\kappa > 0$ *such that* $\|d_k\| \leq \kappa \|g_k\|$. *Then, we have*

$$\lim_{k \to \infty} NF_{l(k)} = \lim_{k \to \infty} \|F(x_k)\|. \tag{20}$$

Proof There are two cases to consider.

Case 1 $k \in \mathcal{I}_1$. It is followed from the definition of x_{k+1} and $f_{l(k)} \geq f_k$ that

$$\frac{f_{l(k)} - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)} \geq \frac{f_k - f(x_k + d_k)}{m_k(x_k) - m_k(x_k + d_k)} \geq \mu_1.$$

Now, similar to the proof of Theorem 3.2 in Ahookhosh and Amini (2010), we can deduce that

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{I}_1}} NF_{l(k)} = \lim_{\substack{k \to \infty \\ k \in \mathcal{I}_1}} \|F(x_k)\|.$$

Case 2 $k \in \mathcal{I}_2$. For $k > N$, using (5) and (14), we obtain

$$f_{l(k)} = f(x_{l(k)-1} + \alpha_{l(k)-1} d_{l(k)-1}) \leq f_{l(k)-1} + \gamma \alpha_{l(k)-1} g_{l(k)-1}^T d_{l(k)-1}.$$

This inequality along with Lemma 1 and (17) implies that

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{I}_2}} \alpha_{l(k)-1} g_{l(k)-1}^T d_{l(k)-1} = 0. \tag{21}$$

On the other hand, from $\|d_k\| \leq \kappa \|g_k\|$, (H2) and (17), we can conclude that

$$\begin{aligned} g_k^T d_k &\leq -L_k \|F_k\|^2 \\ &= -\frac{L_k}{M_1^2} (M_1 \|F_k\|)^2 \\ &\leq -\frac{L_k}{M_1^2} \|g_k\|^2 \\ &\leq -\frac{L_k}{\kappa_1^2 M_1^2} \|d_k\|^2. \end{aligned}$$

This fact along with (21) implies that

$$\lim_{\substack{k \to \infty \\ k \in \mathcal{I}_2}} \alpha_{l(k)-1} \|d_{l(k)-1}\| = 0.$$

The rest of the proof, for $k \in \mathcal{I}_2$, can follow exactly similar to [Fan (2011), Page 4, 5]. □

Lemma 6 *Suppose that the sequence x_k is generated by Algorithm 1 and there exists a positive constant $\kappa > 0$ such that $\|d_k\| \le \kappa \|g_k\|$. Then, for sufficiently large $k \in \mathcal{I}_2$, the steplength α_k satisfies*

$$\alpha_k > \frac{2\sigma(1-\gamma)L_k}{\eta_2 M_1^2}.$$

Proof Let $\alpha = \alpha_k/\sigma$. Because the backtraking loop of Algorithm 1 implies

$$f_{l(k)} + \gamma\alpha g_k^T d_k < f(x_k + \alpha d_k),$$

using (6) concludes

$$\gamma\alpha g_k^T d_k < f(x_k + \alpha d_k) - f_k. \tag{22}$$

Furthermore, Taylor's theorem implies that there is a $\xi \in [x_k, x_k + \alpha d_k]$ such that

$$f(x_k + \alpha d_k) - f_k = \alpha g_k^T d_k + \frac{1}{2}\alpha^2 d_k^T J(\xi)^T J(\xi) d_k,$$

while Remark 2 implies that there is a positive scaler M_1 such that

$$\frac{1}{2} d_k^T J(\xi)^T J(\xi) d_k \le \frac{M_1^2}{2}\|d_k\|^2,$$

for any $\xi \in [x_k, x_k + \alpha d_k]$. These facts along with (22) imply that

$$\gamma g_k^T d_k < g_k^T d_k + \frac{1}{2} M_1^2 \alpha \|d_k\|^2,$$

or, equivalently,

$$-(1-\gamma) g_k^T d_k < \frac{1}{2}\alpha M_1^2 \|d_k\|^2.$$

On the other hand, from (17), we obtain

$$(1-\gamma) L_k \|g_k\|^2 < \frac{M_1^2}{2}\frac{\alpha_k}{\sigma}\|d_k\|^2.$$

By using (16), we have

$$\|d_k\| \le \Delta_k \le \eta_2 N F_{l(k)},$$

hence

$$\alpha_k > \frac{2\sigma(1-\gamma)L_k\|F_k\|^2}{M_2^2\|d_k\|^2} \geq \frac{2\sigma(1-\gamma)L_k\|F_k\|^2}{M_2^2\eta_2 NF_{l(k)}^2}.$$

This fact along with $\lim_{\substack{k\to\infty \\ k\in\mathcal{I}_2}} NF_{l(k)} = \lim_{\substack{k\to\infty \\ k\in\mathcal{I}_2}} \|F(x_k)\|$, for sufficiently large k, results that

$$\alpha_k > \frac{2\sigma(1-\gamma)L_k}{\eta_2 M_2^2},$$

which completes the proof of the lemma. □

At this point, the global convergence of Algorithm 1 based on the mentioned assumptions of this section can be investigated.

Theorem 1 *Suppose that Assumptions (H1)–(H3) hold. Then Algorithm 1 either stops at a stationary point of $f(x)$ or generates an infinite sequence $\{x_k\}$ such that*

$$\lim_{k\to\infty} \|F_k\| = 0. \tag{23}$$

Proof By contradiction, for all sufficiently large k, assume that there exist a constant $\epsilon > 0$ and an infinite subset $K \subseteq \mathbf{N} \cup \{0\}$ such that

$$\|F_k\| > \epsilon, \quad \text{for all } k \in K. \tag{24}$$

Now, we consider the following two cases:

Case 1 $k \in \mathcal{I}_1$. Using (19), (24) and $r_k > \mu_1$, it can be written

$$f_k - f(x_k + d_k) \geq \mu_1[m_k(x_k) - m_k(x_k + d_k)] \geq \mu_1 L_k \|F_k\|^2 \geq \mu_1\epsilon^2 L_k.$$

Case 2 $k \in \mathcal{I}_2$. Using (6), (18) and Lemma 6, we obtain

$$f(x_k + \alpha_k d_k) \leq f_{l(k)} + \gamma\alpha_k g_k^T d_k \leq f_{l(k)} - \gamma\frac{2\sigma(1-\gamma)L_k}{\eta_2 M_1^2} L_k\|F_k\|^2$$
$$\leq f_{l(k)} - \gamma\frac{2\sigma(1-\gamma)L_k}{\eta_2 M_1^2} L_k\epsilon^2.$$

In each two cases, taking a limit from both sides of the above inequalities, as $k \to \infty$, give $\lim_{k\to\infty} L_k = 0$. This clearly contradicts with Lemma 2. Therefore, (24) is incorrect and the proof is completed. □

This theorem guarantees that the stoping criterion of Algorithm 1, that is $\|F_k\| < \epsilon$, is eventually held.

To establish the quadratic convergence rate of the sequence generated by Algorithm 1, some additional assumptions are required, see Fan and Pan (2010), Yuan et al. (2011), Yuan (1998), Zhang and Wang (2003), Esmaeili and Kimiaei (2014), Yamashita and Fukushima (2001). We assume x^* is a solution (1). These conditions can be stated as follows:

(H4) There exist constants $c_0 > 0$ and $\rho_1 \in (0, 1)$ such that

$$\|F(x) - F(y) + J(y)(x - y)\| \le c_0\|x - y\|^2, \quad \text{for all} \quad x, y \in N(x_*, \rho_1),$$

where $N(x_*, \rho_1) := \{x| \ \|x - x_*\| \le \rho_1\}$.

It is clear that (H4) holds if $F(x)$ is continuously differentiable and $J(x)$ is Lipschitz continuous.

Remark 3 Assumption (H3) and the mean value theorem conclude that there exist constants $c_1 > 0$ and $\rho_2 \in (0, 1)$ such that

$$c_1\|x - x_*\| \le \|F(x)\| = \|F(x) - F(x_*)\|, \quad \text{for all} \quad x \in N(x_*, \rho_2),$$

see Li and Fukushima (2000b).

For the purpose of our quadratic convergence, we set $\rho := \min[\rho_1, \rho_2]$.

Theorem 2 *Suppose that Assumptions (H1)–(H4) hold, the sequence $\{x_k\}$ generated by Algorithm 1 is convergent to x_*. Then, for sufficiently large k, we have*

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

furthermore, the sequence $\{x_k\}$ is quadratically convergent to x_*.

Proof If d_k is a solution of (3) and $k \in \mathcal{I}_2$, then we first show that $x_{k+1} = x_k + \alpha_k d_k$. Therefore, it is sufficient to show that $\alpha_k = 1$, for sufficiently large k. From this the fact that d_k is feasible for the subproblem (3), the relationship (15), Lemma 1, we simply have

$$\|d_k\| \le \Delta_k \le \eta_2 N F_{l(k-1)} \to 0, \quad as \quad k \to \infty. \tag{25}$$

Remark 2 implies that there is a positive scaler M_1 such that

$$\frac{1}{2} d_k^T J(\xi)^T J(\xi) d_k \le \frac{M_1}{2}\|d_k\|^2,$$

for any $\xi \in (x_k, x_k + \alpha d_k)$. This fact along with the Taylor expansion, (16), (18), (27) and Lemma 5 implies that

$$\begin{aligned} f(x_k + d_k) - f_{l(k)} - \gamma g_k^T d_k &\le f(x_k + d_k) - f_k - \gamma g_k^T d_k \\ &\le (1 - \gamma) g_k^T d_k + \frac{1}{2} d_k^T J(\xi)^T J(\xi) d_k \\ &\le -(1 - \gamma) L_k \|F_k\|^2 + \frac{M_2^2}{2}\|d_k\|^2 \\ &\le -(1 - \gamma) L_k \|F_k\|^2 + \frac{(M_2\eta_2)^2}{2} N F_{l(k)}^2 \to 0, \\ &\quad \text{as} \quad k \to \infty. \end{aligned} \tag{26}$$

for any $\xi \in (x_k, x_k + \alpha d_k)$. Thus, for all sufficiently large k, $\alpha_k = 1$ is taken by Algorithm 1, i.e., $x_{k+1} = x_k + d_k$.

At this point, the quadratic convergence of the sequence $\{x_k\}$ generated by Algorithm 1 is investigated. Regarding the mean value theorem, one can easily deduce that there is a ξ_k is between x_k and x_* such that

$$\|F_k\| = \|F_k - F(x_*)\| \leq \|J(\xi_k)\| \, \|x_k - x_*\|,$$

for all $x_k \in N(x_*, \rho)$. By (H2), it is derived that

$$\|F_k\| \leq M_1 \|x_k - x_*\|. \tag{27}$$

Lemma 5 results that the sequence $\{NF_{l(k)}\}_{k\geq 0}$ satisfies $|NF_{l(k)} - \|F_k\|| \leq \varepsilon$ for all $k \in \mathbf{N} \cup \{0\}$ sufficiently large. As a result of this fact and (27), as $k \to \infty$, it can be concluded that

$$NF_{l(k)} \leq \|F_k\| + \varepsilon \leq M_1 \|x_k - x_*\| + \varepsilon, \quad \forall\, \varepsilon > 0,$$

leading to

$$NF_{l(k)} \leq M_1 \|x_k - x_*\|, \quad \text{as} \quad k \to \infty,$$

and so

$$\|d_k\| \leq \Delta_k \leq \eta_2 NF_{l(k)} \leq \eta_2 M_1 \, \|x_k - x_*\| = O(\|x_k - x_*\|). \tag{28}$$

Now, we consider the following two cases:

Case 1 If $\|x_k - x_*\| \leq \|d_k\|$, since $x_k - x_*$ is a feasible point of problem (3), then (H4) results that

$$\begin{aligned}
\frac{1}{2}\|F_k + J_k d_k\|^2 = m_k(x_k + d_k) \leq m_k(x_k + (x_* - x_k)) &= \frac{1}{2}\|F_k + J_k(x_k - x_*)\|^2 \\
&= \frac{1}{2}\|F_k - F_* + J_k(x_k - x_*)\|^2 \\
&\leq \frac{c_0^2}{2}\|x_k - x_*\|^4.
\end{aligned} \tag{29}$$

Case 2 If $\|x_k - x_*\| > \|d_k\|$, then $\frac{\|d_k\|}{\|x_k - x_*\|}(x_* - x_k)$ is a feasible point of problem (3). This fact along with (H4) implies that

$$\begin{aligned}
\frac{1}{2}\|F_k + J_k d_k\|^2 &= m_k(x_k + d_k) \leq m_k\Big(x_k + \frac{\|d_k\|}{\|x_k - x_*\|}(x_* - x_k)\Big) \\
&= \frac{1}{2}\|F_k + \frac{\|d_k\|}{\|x_k - x_*\|} J_k(x_k - x_*)\|^2 \\
&\leq \frac{1}{2}\frac{\|d_k\|^2}{\|x_k - x_*\|^2}\|F_k - F_* + J_k(x_k - x_*)\|^2
\end{aligned}$$

$$\leq \frac{1}{2}\frac{\|d_k\|^2}{\|x_k - x_*\|^2}c_0^2\|x_k - x_*\|^4$$
$$= \frac{c_0^2}{2}\|d_k\|^2\|x_k - x_*\|^2$$
$$\leq \frac{c_0^2}{2}\|x_k - x_*\|^4. \tag{30}$$

By Remark 3, (28), (29) and (30), we conclude that

$$\begin{aligned} c_1\|x_{k+1} - x_*\| \leq \|F(x_{k+1})\| = \|F(x_k + d_k)\| &\leq \|F_k + J_k d_k\| + O(\|d_k\|^2) \\ &\leq c_0^2\|x_k - x_*\|^2 + O(\|x_k - x_*\|^2) \\ &= O(\|x_k - x_*\|^2). \end{aligned}$$

So

$$\|x_{k+1} - x_*\| = O(\|x_k - x_*\|^2),$$

that shows the sequence $\{x_k\}$ generated by Algorithm 1 is quadratically convergent. Therefore, the proof is completed. □

4 Preliminary numerical experiments

We now firstly report the results obtained by running Algorithm 1 (LSTR) in comparison with the traditional trust-region algorithm (TTR), the adaptive trust-region algorithm from Zhang and Wang (2003) (ATRZ), the adaptive trust-region algorithm of Fan and Pan (2010) (ATRF) on the set of some nonlinear system of equations. For all of these codes, the trust-region subproblems are coded due to Steihaug-Toint procedure, see Conn et al. (2000). The Steihaug-Toint algorithm terminates at $x_k + d$ when

$$\|\nabla f_k(x_k + d)\| \leq 0.1 \min\left\{\frac{1}{k+1}, \|\nabla f_k(x_k)\|\right\} \|\nabla f_k(x_k)\|,$$

holds. The Jacobian matrix J_k can be either evaluated analytically by a user-supplied function or approximated using finite-differences formula provided by the code. More precisely, in the latter case, the Jacobian matrix J_k is approximated as follows :

$$[J_k]_{.j} \sim \frac{1}{h_j}(F(x_k + h_j e_j) - F_k),$$

where $[J_k]_{.j}$ denotes the j-th column of J_k, e_j is the j-th vector of the canonic basis and

$$h_j := \begin{cases} \sqrt{\epsilon_m} & \text{if } x_{k_j} = 0, \\ \sqrt{\epsilon_m} sign(x_{k_j}) \max\left\{|x_{k_j}|, \frac{\|x_k\|_1}{n}\right\} & \text{otherwise,} \end{cases}$$

for more details see Bellavia et al. (2004). All codes are written in MATLAB 9 programming environment with double precision format in the same subroutine. In our numerical experiments, the algorithms are stopped whenever the total number of iterates exceeds 1000, or when

$$\|F_k\| \leq 10^{-5}\sqrt{n}.$$

During implementations, we verified whether the different codes converged to the same point, and data is provided only for problems in which all algorithms converged to the identical point.

LSTR takes advantages of the parameters $\mu_1 = 0.1$, $\mu_2 = 0.9$, $\eta_1 = 0.25$, $\eta_2 = 3$, $\gamma = 10^{-4}$, $M = 10$, $\sigma_1 = 0.1$, $\sigma_2 = 0.5$ and at Step 2 the scalar σ is computed by means of a quadratic interpolation formula. For LSTR, the trust-region radius is updated by

$$\Delta_{k+1} := \begin{cases} \eta_1 \alpha_k \|d_k\| & \text{if } r_k < \mu_1, \\ NF_{l(k+1)} & \text{if } \mu_1 \leq r_k < \mu_2, \\ \eta_2 NF_{l(k+1)} & \text{if } r_k \geq \mu_2, \end{cases}$$

and $n(k+1)$ is updated by $n(k+1) := \min\{n(k)+1, N\}$ where $n(0) := 0$. TTR employs the parameters $\mu_1 = 0.1$, $\mu_2 = 0.9$ where the trust-region radius like Conn et al. (2000) is computed by the following formula

$$\Delta_{k+1} := \begin{cases} c_1 \|d_k\| & r_k < \mu_1, \\ \Delta_k & \mu_1 \leq r_k \leq \mu_2, \\ c_2 \Delta_k & r_k \geq \mu_2, \end{cases}$$

where $c_1 = 0.25$ and $c_2 = 3$. We also decide to follow the literature Toint (1986) in exploiting $\Delta_0 = 1$ as an initial trust-region radius for TTR. The parameters of ATRZ and ATRF are chosen the same as what proposed in articles Zhang and Wang (2003) and Fan and Pan (2010), respectively. The results for considered algorithms are summarized in Table 1. In this table, N_i and N_f respectively indicate the total number of iterates and the total number of function evaluations. Test problems were selected from wide range of literatures: problems 1–40 are taken from La Cruz et al. (2004) and problems 41–46 are selected from Lukšan and Vlček (1999).

It is followed from Table 1 that in most cases the total number of iterates and function evaluations of the LSTR are less than the other presented algorithms, and the algorithms solve all of test functions successfully. In spite of the fact that it generally seems that the performance of LSTR is better than other presented algorithms. In this point, to demonstrate the overall behavior of the algorithms and get more insight about the performances, use the performance profile proposed by Dolan and Moré in Dolan and Moré (2002) and show performance of algorithms, based on both N_i and N_f, in Figs. 1 and 2 respectively. In this procedure, the profile of each code is measured considering the ratio of its computational outcome versus the best numerical outcome of all codes. This profile offers a tool for comparing the performance of iterative processes in statistical structure. In the following figures, P designates the percentage of problems which are solved within a factor τ of the best solver.

Figure 1 clearly indicates that LSTR outperforms ATRZ and ATRF regarding the total number of iterates. In particular, LSTR has the most wins in nearly 89 % of the tests with the greatest efficiency. Meanwhile, in the sense of the ability of completing a run successfully, it is the best among considered algorithms because it grows up faster

Table 1 Numerical results

Problem name	Dim	TTR N_i/N_f	ATRZ N_i/N_f	ATRF N_i/N_f	LSTR N_i/N_f
1. Exponential 1	500	3/4	3/4	3/4	3/4
2. Exponential 2	500	2/3	2/3	2/3	2/3
3. Extended Rosenbrock	500	15/20	13/14	10/17	9/10
4. Chandrasekhars H-equation	500	5/6	9/10	3/4	4/5
5. Trigonometric	100	Failed	Failed	Failed	9/13
6. Singular	500	17/18	33/34	14/15	14/15
7. Logarithmic	500	6/7	8/9	4/5	4/5
8. Broyden tridiagonal	500	5/6	4/5	4/5	4/5
9. Trigexp	500	20/28	8/21	8/39	11/15
10. Variable band 1	500	10/12	38/40	8/18	9/11
11. Variable band 2	500	12/17	61/63	10/25	10/12
12. Function 15	500	12/15	16/17	8/9	8/9
13. Strictly convex 1	500	6/7	6/7	4/5	4/5
14. Strictly convex 2	500	9/10	11/12	7/8	7/8
15. Penalty	500	5/6	71/72	4/5	27/28
16. Zero Jacobian	500	16/17	13/14	13/14	13/14
17. Geometric programming	100	13/14	144/145	39/40	13/14
18. Function 21	501	7/8	12/13	5/6	5/6
19. Linear function-full rank 1	500	9/10	51/52	2/3	2/3
20. Linear function-full rank 2	500	2/3	2/3	2/3	2/3
21. Brown almost linear	500	2/3	2/3	2/3	2/3
22. Variable dimensioned	500	21/22	20/21	20/21	20/21
23. Geometric	100	10/11	128/129	33/34	10/11
24. Extended Powel singular	500	1/2	1/2	1/2	1/2
25. Function 27	500	16/17	13/14	13/14	13/14
26. Tridimensional valley	501	9/10	7/8	6/7	6/7
27. Complementary	500	7/8	11/12	6/7	6/7
28. Hanbook	500	3/4	3/4	3/4	3/4
29. Tridiagnal system	500	60/72	56/113	60/356	21/22
30. Five-diagonal system	500	Failed	Failed	Failed	16/18
31. Seven-diagonal system	500	81/89	Failed	Failed	21/23
32. Extended Freudentein and Roth	500	17/18	13/14	13/14	13/14
33. Extended cragg and levy	500	23/24	90/91	21/38	18/19
34. Extended Wood	500	6/7	4/5	4/5	4/5
35. Triadiagnal exponential	500	5/6	2/3	2/3	2/3
36. Brent	500	13/14	11/12	11/12	11/12
37. Thorech	500	9/12	8/13	8/26	9/11
38. Trigonometric system	500	2/3	2/3	2/3	2/3
39. Broyden banded	500	6/7	5/6	5/6	5/6

Table 1 continued

Problem name	Dim	TTR N_i/N_f	ATRZ N_i/N_f	ATRF N_i/N_f	LSTR N_i/N_f
40. Discrete integral equation	500	2/3	2/3	2/3	2/3
41. Countercurrent reactors 1	504	14/19	82/83	10/12	9/10
42. Singular Broyden	500	10/11	11/12	9/10	9/10
43. Structured Jacobian	500	11/14	15/16	8/15	8/9
44. Extended Powell Singular	500	12/13	30/31	11/21	11/12
45. Generalized Broyden banded	500	6/7	5/6	5/6	5/6
46. Extended powell badly scaled	500	81/117	756/757	102/146	17/18

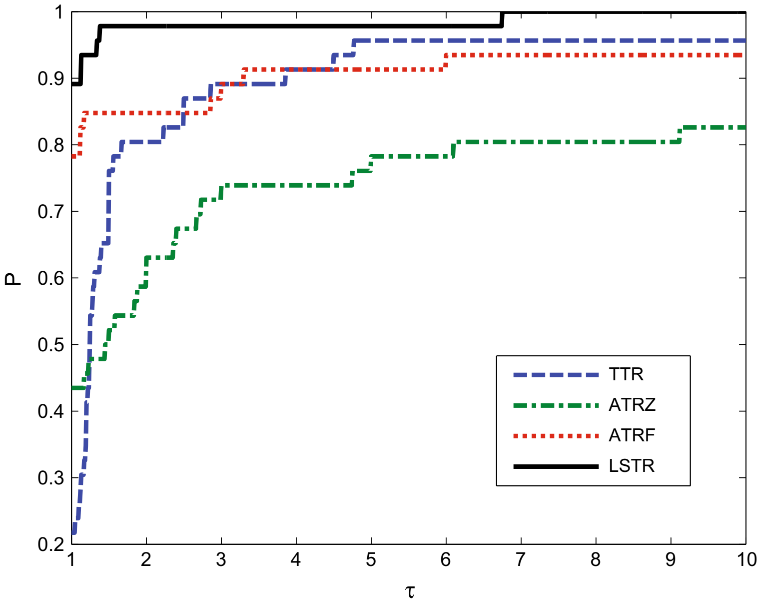

Fig. 1 Iterates performance profile for the presented algorithms

than the others and reaches 1 more rapidly. However, as illustrated in Fig. 2, LSTR implements remarkably better than the others where it has most wins in approximately 96 % of performed tests concerning the total number of function evaluations. Furthermore, Figs. 1 and 2 show similar patterns in the sense of the ability of completing a run successfully. As a result, this fact directly implies that the total number of solving the trust-region subproblems is notably decreased thanks to using the LSTR algorithm.

Summarizing our discussion, we employ a large set of problems that occur in applications while the obtained results suggest that LSTR constitutes an efficient and robust approach for solving the nonlinear system of equations which outperforms some well-known codes in this field.

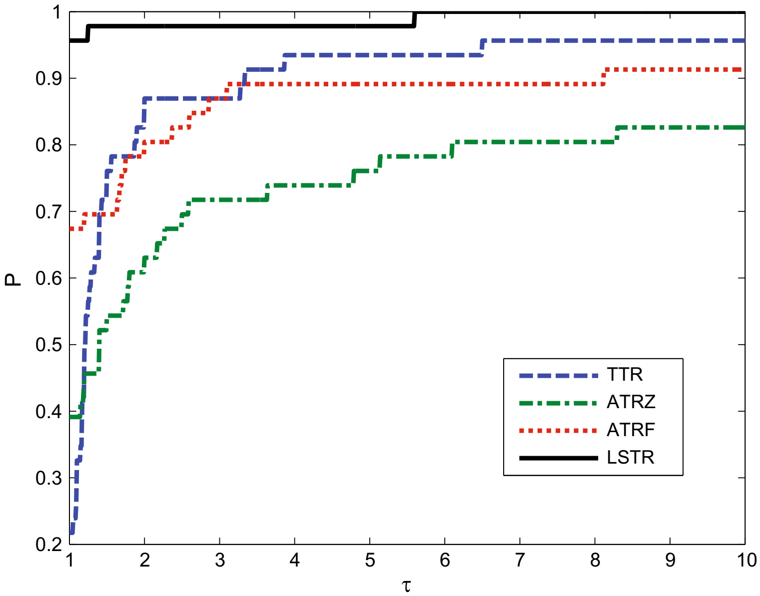

Fig. 2 Function evaluations performance profile for the presented algorithms

5 Concluding remarks

The present paper proposes a new trust-region algorithm for solving a system of nonlinear equations by combining two techniques of adaptive radius and nonmonotone line search. The adaptive technique is used to decrease the total number of iterations, because of the optimum found within the greater region. The nonmonotone line search technique is applied to prevent breaking the trail step that in each case declines the number of solving subproblems leading to decreasing computational costs. Nevertheless, these modifications in the traditional trust-region procedure are favorably encouraging, the global and the quadratic convergence properties of the proposed algorithms are established. Preliminary numerical results on large set of nonlinear systems indicate that the method proposed will have significant profits in computational costs.

References

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

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

Bellavia S, Macconi M, Morini B (2004) STRSCNE: a scaled trust-region solver for constrained nonlinear equations. Comput Optim Appl 28:31–50

Conn AR, Gould NIM, Toint PhL (2000) Trust-region methods. Society for Industrial and Applied Mathematics SIAM, Philadelphia

Dolan ED, Moré JJ (2002) Benchmarking optimization software with performance profiles. Math Program 91:201–213

Esmaeili H, Kimiaei M (2014) A new adaptive trust-region method for system of nonlinear equations. Appl Math Model 38:3003–3015

Esmaeili H, Kimiaei M (2015) An efficient adaptive trust-region method for systems of nonlinear equations. Int J Comput Math 92(1):151–166

Fan JY (2005) Convergence rate of the trust region method for nonlinear equations under local error bound condition. Comput Optim Appl 34:215–227

Fan JY(2011)An improved trust region algorithmfor nonlinear equations.ComputOptimAppl 48(1):59–70

Fan JY, Pan JY (2010) A modified trust region algorithm for nonlinear equations with new updating rule of trust region radius. Int J Comput Math 87(14):3186–3195

Fasano G, Lampariello F, Sciandrone M (2006) A truncated nonmonotone Gauss–Newton method for large-scale nonlinear least-squares problems. Comput Optim Appl 34(3):343–358

Fischer A, Shukla PK, Wang M (2010) On the inexactness level of robust Levenberg–Marquardt methods. Optimization 59(2):273–287

Gertz EM (1999) Combination trust-region line-search methods for unconstrained optimization. University of California San Diego, San Diego

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

Grippo L, Lampariello F, Lucidi S (1989) A truncated Newton method with nonmonotone linesearch for unconstrained optimization. J Optim Theory Appl 60(3):401–419

Grippo L, Lampariello F, Lucidi S (1991) A class of nonmonotone stabilization method in unconstrained optimization. Numer Math 59:779–805

Grippo L, Sciandrone M (2007) Nonmonotone derivative-free methods for nonlinear equations. Comput Optim Appl 37:297–328

La Cruz W, Raydan M (2003) Nonmonotone spectral methods for large-scale nonlinear systems. Optim Methods Softw 18(5):583–599

La Cruz W, Venezuela C, Martínez JM, Raydan M (2004) Spectral residual method without gradient information for solving large-scale nonlinear systems of equations: theory and experiments. In: Technical report RT-04-08, July 2004

Li DH, Fukushima M (2000a) A derivative-free line search and global convergence of Broyden-like method for nonlinear equations. Optim Methods Softw 13:181–201

Li DH, Fukushima M (2000b) A globally and superlinearly convergent Gauss–Newton-Based BFGS method for symmetric nonlinear equations. SIAM J Numer Anal 37(1):152–172

Lukšan L, Vlček J (1999) Sparse and partially separable test problems for unconstrained and equality constrained optimization. In: Technical report, no 767

Nocedal J, Yuan YX (1998) Combining Trust-region and line-search techniques. Optimization Technology Center mar OTC 98(04) 1998

Nocedal J, Wright SJ (2006) Numerical optimization. Springer, New York

Sartenaer A (1997) Automatic determination of an initial trust region in nonlinear programming. SIAM J Sci Comput 18(6):1788–1803

Toint Ph L (1982) Towards an efficient sparsity exploiting Newton method for minimization. In: Sparse matrices and their uses. Academic Press, New York 1982 I. S. Duff 57–87

Toint PhL (1986) Numerical solution of large sets of algebraic nonlinear equations. Math. Comput. 46(173):175–189

Yamashita N, Fukushima M (2001) On the rate of convergence of the Levenberg–Marquardt method. Computing 15:239–249

Yuan G, Lu S, Wei Z (2011) A new trust-region method with line search for solving symmetric nonlinear equations. Int J Comput Math 88(10):2109–2123

Yuan Y (1998) Trust region algorithm for nonlinear equations. Information 1:7–21

Zhang HC, Hager WW (2004) A nonmonotone line search technique for unconstrained optimization. SIAM J Optim 14(4):1043–1056

Zhang J, Wang Y (2003) A new trust region method for nonlinear equations. Math Methods Oper Res 58:283–298

Zhang XS, Zhang JL, Liao LZ (2002) An adaptive trust region method and its convergence. Sci China 45:620–631