Journal of Inequalities and Applications
a SpringerOpen Journal

RESEARCH Open Access

A trust region spectral method for large-scale systems of nonlinear equations

Meilan Zeng[1] and Guanghui Zhou[2*]

[*]Correspondence:
163zgh@163.com
[2]School of Mathematical Sciences/Information College, Huaibei Normal University, Huaibei, 235000, China
Full list of author information is available at the end of the article

Abstract

The spectral gradient method is one of the most effective methods for solving large-scale systems of nonlinear equations. In this paper, we propose a new trust region spectral method without gradient. The trust region technique is a globalization strategy in our method. The global convergence of the proposed algorithm is proved. The numerical results show that our new method is more competitive than the spectral method of La Cruz *et al.* (Math. Comput. 75(255):1429-1448, 2006) for large-scale nonlinear equations.

MSC: 65H10; 90C06

Keywords: nonlinear equations; trust region; spectral method; large-scale problem

1 Introduction

In this paper we introduce a trust region spectral method for solving large-scale systems of nonlinear equations

$$F(x) = 0, \tag{1}$$

where $F : R^n \to R^n$ is continuously differentiable and its Jacobian matrix $J(x) \in R^{n\times n}$ is sparse, n is large. Large-scale systems of nonlinear equations have been widely applied in many aspects, such as network-flow problems, discrete boundary value problems, etc.

Many algorithms have been presented for solving the large-scale problem (1). Bouaricha *et al.* [2] proposed tensor methods. Bergamaschi *et al.* [3] proposed inexact quasi-Newton methods. The above methods need to calculate the Jacobian matrix or an approximation of it at each iteration. La Cruz and Raydan [4] introduced the spectral method for (1). The method uses the residual $\pm F(x_k)$ as a search direction and the trial point at each iteration is $x_k - \lambda_k F(x_k)$, where λ_k is a spectral coefficient. λ_k satisfies the Grippo-Lampariello-Lucidi (GLL) line search condition

$$f(x_k + \lambda_k d_k) \le \max_{0\le j\le M-1} f(x_{k-j}) + \alpha\lambda_k \nabla f(x_k)^T d_k, \tag{2}$$

where $f(x) = \frac{1}{2}\|F(x)\|^2$, M is a nonnegative integer, α is a small positive number and $d_k = \pm F(x_k)$. This method also requires one to compute a directional derivative or a very good approximation of it at every iteration. Later La Cruz *et al.* [1] proposed a spectral method without gradient information, which uses a nonmonotone line search globalization strategy

$$f(x_k + \lambda_k d_k) \leq \max_{0 \leq j \leq M-1} f(x_{k-j}) + \eta_k - \alpha \lambda_k^2 f(x_k), \tag{3}$$

where $\sum_k \eta_k \leq \eta < \infty$. Meanwhile, conjugate gradient techniques have been developed for solving large-scale nonlinear equations (see [5–7]). In fact, spectral gradient, BFGS quasi-Newton, and conjugate gradient methods can solve large-scale optimization problems and systems of nonlinear equations (see [8–13]). The advantage of spectral methods is that the storage of certain matrices associated with the Hessian of objective functions can be avoided.

The purpose of this paper is to extend the spectral method for solving large-scale systems of nonlinear equations by using the trust region technique. For the traditional trust region methods [14], at each iterative point x_k, the trial step d_k is obtained by solving the following trust region subproblem:

$$\min q_k(d) \quad \text{such that} \quad \|d\| \leq \Delta_k, \tag{4}$$

where $q_k(d) = \frac{1}{2}\|F(x_k) + J(x_k)d\|^2$.

The above trust region methods are particularly effective for small to medium-sized systems of nonlinear equations; however, the computation and storage loads can greatly increase with increased dimension.

For the large-scale problems of nonlinear equations, we use $\gamma_k I$ as an approximation of $J(x_k)$. At each iterative point x_k in our method, the trial step d_k is obtained by solving the following subproblem:

$$\min q_k(d) = \frac{1}{2}\|F_k + \gamma_k d\|^2 \quad \text{such that} \quad \|d\| \leq \Delta_k, \tag{5}$$

where γ_k is the spectral coefficient and $F_k = F(x_k)$. The classic quasi-Newton equation is

$$B_{k+1} d_k = y_k. \tag{6}$$

In (6), we left-multiply y_k^T and set $B_{k+1} = \gamma_{k+1} I$, it follows that

$$\gamma_{k+1} = \frac{y_k^T y_k}{y_k^T d_k}, \tag{7}$$

where $d_k = x_{k+1} - x_k$ and $y_k = F_{k+1} - F_k$.

The paper is organized as follows. Section 2 introduces the new algorithm. The convergence theory is presented in Section 3. Section 4 demonstrates preliminary numerical results on test problems.

2 New algorithm

In this section, we give a trust region spectral method for solving large-scale systems of nonlinear equations. Let d_k be the solution of the trust region subproblem (5). We define the actual reduction as

$$Ared_k(d_k) = f(x_k) - f(x_k + d_k), \tag{8}$$

the predict reduction as

$$Pred_k(d_k) = q_k(0) - q_k(d_k). \tag{9}$$

Now we present our algorithm for solving (1). The algorithm is given as follows.

Algorithm 1

Step 0. Choose $0 < \eta_1 < \eta_2 < 1$, $0 < \beta_1 < 1 < \beta_2$, $\epsilon > 0$. Initialize x_0, $0 < \Delta_0 < \bar{\Delta}$. Set $k := 0$.
Step 1. Evaluate F_k, if $\|F_k\| \leq \epsilon$, then terminate.
Step 2. Solve the trust region subproblem (5) to obtain d_k.
Step 3. Compute

$$r_k = \frac{Ared_k(d_k)}{Pred_k(d_k)}. \tag{10}$$

If $r_k < \eta_1$, then $\Delta_k = \beta_1 \Delta_k$, go to Step 2. Otherwise, go to Step 4.
Step 4. $x_{k+1} = x_k + d_k$;

$$\Delta_{k+1} = \begin{cases} \min\{\beta_2 \Delta_k, \bar{\Delta}\}, & \text{if } r_k \geq \eta_2, \\ \Delta_k, & \text{otherwise.} \end{cases}$$

Compute γ_{k+1} by (7). Set $k := k + 1$, go to Step 1.

3 Convergence analysis

In this section, we prove the global convergence of Algorithm 1. The global convergence of Algorithm 1 needs the following assumptions.

Assumption A

(1) The level set $\Omega = \{x \in R^n | f(x) \leq f(x_0)\}$ is bounded.
(2) The following relation holds:

$$\big\| [J_k - \gamma_k I]^T F_k \big\| = O\big(\|d_k\|\big).$$

Then we get the following lemmas.

Lemma 3.1 $|Ared_k(d_k) - Pred_k(d_k)| = O(\|d_k\|^2)$.

Proof By (8) and (9), we have

$$\begin{aligned} \big|Ared_k(d_k) - Pred_k(d_k)\big| &= \big|q_k(d_k) - f(x_k + d_k)\big| \\ &= \frac{1}{2}\big|\|F_k + \gamma_k d_k\|^2 - \big\|F_k + J_k d_k + O\big(\|d_k\|^2\big)\big\|^2\big| \\ &= \big|\gamma_k F_k^T d_k - F_k^T J_k d_k + O\big(\|d_k\|^2\big) + O\big(\|d_k\|^4\big)\big| \end{aligned}$$

$$\begin{aligned} &\leq \big\| [\gamma_k I - J_k]^T F_k \big\| \|d_k\| + O\big(\|d_k\|^2\big) \\ &= O\big(\|d_k\|^2\big). \end{aligned}$$

This completes the proof. □

Similar to Zhang and Wang [15], or Yuan *et al.* [16], we obtain the following result.

Lemma 3.2 *If d_k is a solution of* (5), *then*

$$Pred_k(d_k) \geq \frac{1}{2}\|\gamma_k F_k\| \min\left\{\Delta_k, \frac{\|F_k\|}{|\gamma_k|}\right\}. \tag{11}$$

Proof Since d_k is a solution of (5), for any $\alpha \in [0,1]$, it follows that

$$\begin{aligned} Pred_k(d_k) &= \frac{1}{2}\big(\|F_k\|^2 - \|F_k + \gamma_k d_k\|^2\big) \\ &\geq \frac{1}{2}\left(\|F_k\|^2 - \left\|F_k - \gamma_k \frac{\alpha \Delta_k}{\|\gamma_k F_k\|}\gamma_k F_k\right\|^2\right) \\ &= \alpha \Delta_k \|\gamma_k F_k\| - \frac{1}{2}\alpha^2 \Delta_k^2 \gamma_k^2. \end{aligned} \tag{12}$$

Then we have

$$\begin{aligned} Pred_k(d_k) &\geq \max_{0 \leq \alpha \leq 1}\left[\alpha \Delta_k \|\gamma_k F_k\| - \frac{1}{2}\alpha^2 \Delta_k^2 \gamma_k^2\right] \\ &\geq \frac{1}{2}\|\gamma_k F_k\| \min\left\{\Delta_k, \frac{\|F_k\|}{|\gamma_k|}\right\}. \end{aligned} \tag{13}$$

The proof is complete. □

Lemma 3.3 *Algorithm* 1 *does not circle between Step* 2 *and Step* 3 *infinitely.*

Proof If Algorithm 1 circles between Step 2 and Step 3 infinitely, then for all $i = 1, 2, \ldots$, we have $x_{k+i} = x_k$, and $\|F_k\| > \epsilon$, which implies that $r_k < \eta_1$, $\Delta_k \to 0$.

By Lemmas 3.1 and 3.2, we have

$$|r_k - 1| = \frac{|Ared_k(d_k) - Pred_k(d_k)|}{|Pred_k(d_k)|} \leq \frac{2O(\|d_k\|^2)}{\Delta_k \|\gamma_k F_k\|} \to 0. \tag{14}$$

Therefore, for k sufficiently large

$$r_k \geq \eta_1, \tag{15}$$

this contradicts the fact that $r_k < \eta_1$. □

Lemma 3.4 *Let Assumption* A *hold and $\{x_k\}$ be generated by Algorithm* 1, *then $\{x_k\} \subset \Omega$. Moreover, $\{f(x_k)\}$ converges.*

Proof By the definition of Algorithm 1, we have

$$r_k \geq \eta_1 > 0. \tag{16}$$

This implies

$$f(x_{k+1}) \leq f(x_k) \leq \cdots \leq f(x_0).$$

Therefore, $\{x_k\} \subset \Omega$. According to $f(x_k) \geq 0$, we know that $\{f(x_k)\}$ converges. □

The following theorem shows that Algorithm 1 is global convergent under the conditions of Assumption A.

Theorem 3.5 *Let Assumption* A *hold*, $\{x_k\}$ *be generated by Algorithm* 1. *Then the algorithm either stops finitely or generates an infinite sequence* $\{x_k\}$ *such that*

$$\lim_{k\to\infty} \|F_k\| = 0. \tag{17}$$

Proof Assume that Algorithm 1 does not stop after finite steps. Now we suppose that (17) does not hold, then there exist a constant $\varepsilon > 0$ and a subsequence $\{k_j\}$ satisfying

$$\|F_{k_j}\| \geq \varepsilon. \tag{18}$$

Let $K = \{k | \|F_k\| \geq \varepsilon\}$.

Let $S_0 = \{k | r_k \geq \eta_2\}$. Using Algorithm 1 and Lemma 3.2, we have

$$\sum_{k\in S_0} \bigl[f(x_k) - f(x_{k+1})\bigr] \geq \sum_{k\in S_0} \eta_2 \cdot Pred_k(d_k) \geq \sum_{k\in K} \eta_2 \cdot \frac{\varepsilon|\gamma_k|}{2} \min\left\{\Delta_k, \frac{\varepsilon}{|\gamma_k|}\right\}.$$

By Lemma 3.4, we know that $\{f(x_k)\}$ is convergent, then

$$\sum_{k\in S_0} \eta_2 \cdot \frac{\varepsilon|\gamma_k|}{2} \min\left\{\Delta_k, \frac{\varepsilon}{|\gamma_k|}\right\} < \infty.$$

Thus, we have

$$\sum_{k\in S_0} \Delta_k < \infty. \tag{19}$$

From Steps 3-4 of Algorithm 1 it follows that

$$\Delta_{k+1} \leq \Delta_k, \tag{20}$$

for all $k \notin S_0$, thus (19) means

$$\sum_{k\in K} \Delta_k < \infty. \tag{21}$$

Therefore there exists x^* such that

$$\lim_{k\to\infty} x_k = x^*. \tag{22}$$

By (21), we have $\Delta_k \to 0$, which implies

$$Pred_k(d_k) \geq \frac{\varepsilon|\gamma_k|}{2}\Delta_k$$

for all sufficiently large k. The fact that $|Ared_k(d_k) - Pred_k(d_k)| = O(\|d_k\|^2)$ indicates that

$$\lim_{k\to\infty} r_k = 1,$$

which shows that, for sufficiently large k and $k \in K$,

$$\Delta_{k+1} \geq \Delta_k.$$

The above inequality contradicts (20). Thus, the conclusion follows. □

4 Numerical experiments

In this section, the recent spectral method in [1] is called Algorithm 2. We report results of some numerical experiments of Algorithms 1 and 2. We choose 14 test functions as follows (see [4, 6, 17]).

Function 1 The trigonometric function

$$f_i(x) = n - \sum_{j=1}^{n} \cos x_j + i(1 - \cos x_i) - \sin x_i, \quad i = 1, 2, \ldots, n.$$

Initial guess: $x_0 = -(\frac{1}{n}, \ldots, \frac{1}{n})^T$.

Function 2 The discretized two-point boundary value problem

$$F(x) = Ax + \Phi(x),$$

when A is the $n \times n$ tridiagonal matrix given by

$$A = \begin{pmatrix} 8 & -1 & & & \\ -1 & 8 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 8 & -1 \\ & & & -1 & 8 \end{pmatrix},$$

and $\Phi = (\Phi_1(x), \Phi_2(x), \ldots, \Phi_n(x))^T$ with $\Phi_i(x) = \sin x_i - 1$, $i = 1, 2, \ldots, n$.

Initial guess: $x_0 = (50, 0, \ldots, 50, 0)$.

Function 3 The Broyden tridiagonal function

$$f_i(x) = (3 - 2x_i)x_i - x_{i-1} - 2x_{i+1} + 1, \quad i = 1, 2, \ldots, n,$$
$$x_0 = x_{n+1} = 0.$$

Initial guess: $x_0 = (-1, \ldots, -1)^T$.

Function 4 The Broyden banded function

$$f_i(x) = x_i\left(2 + 5x_i^2\right) + 1 - \sum_{j \in J_i} x_j(1 + x_j), \quad i = 1, \ldots, n,$$
$$J_i = \left\{j : j \neq i, \max(1, i - m_l) \leq j \leq \min(n, i + m_u)\right\}, \quad m_l = 5, m_l = 1.$$

Initial guess: $x_0 = (-1, \ldots, -1)^T$.

Function 5 The variable dimensioned function

$$f_i(x) = x_i - 1, \quad i = 1, 2, \ldots, n - 2,$$
$$f_{n-1}(x) = \sum_{j=1}^{n-2} j(x_j - 1),$$
$$f_n(x) = \left(\sum_{j=1}^{n-2} j(x_j - 1)\right)^2.$$

Initial guess: $x_0 = (1 - \frac{1}{n}, 1 - \frac{2}{n}, \ldots, 0)^T$.

Function 6 The discrete boundary value function

$$f_1(x) = 2x_1 + 0.5h^2(x_1 + h + 1)^3 - x_2,$$
$$f_i(x) = 2x_i + 0.5h^2(x_i + ih + 1)^3 - x_{i-1} + x_{i+1}, \quad i = 2, 3, \ldots, n - 1,$$
$$f_n(x) = 2x_n + 0.5h^2(x_n + nh + 1)^3 - x_{i-1},$$
$$h = \frac{1}{n+1}.$$

Initial guess: $x_0 = (h(h-1), h(2h-1), \ldots, h(nh-1))^T$.

Function 7 The logarithmic function

$$f_i(x) = \ln(x_i + 1) - \frac{x_i}{n}, \quad i = 1, 2, 3, \ldots, n.$$

Initial guess: $x_0 = (1, 1, \ldots, 1)^T$.

Function 8 The strictly convex function

$$f_i(x) = e^{x_i} - 1, \quad i = 1, 2, 3, \ldots, n.$$

Initial guess: $x_0 = (\frac{1}{n}, \frac{2}{n}, \ldots, 1)^T$.

Function 9 The exponential function

$$f_1(x) = e^{x_1-1} - 1,$$
$$f_i(x) = i\big(e^{x_i-1} - x_i\big), \quad i = 2, 3, \ldots, n.$$

Initial guess: $x_0 = (\frac{n}{n-1}, \frac{n}{n-1}, \ldots, \frac{n}{n-1})^T$.

Function 10 The extended Rosenbrock function (n is even). For $i = 1, 2, \ldots, n/2$,

$$f_{2i-1}(x) = 10\big(x_{2i} - x_{2i-1}^2\big),$$
$$f_{2i}(x) = 1 - x_{2i-1}.$$

Initial guess: $x_0 = (-1.2, 1, \ldots, -1.2, 1)^T$.

Function 11 The singular function

$$f_1(x) = \frac{1}{3}x_1^3 + \frac{1}{2}x_2^2,$$
$$f_i(x) = -\frac{1}{2}x_i^2 + \frac{i}{3}x_i^3 + \frac{1}{2}x_{i+1}^2, \quad i = 2, 3, \ldots, n-1,$$
$$f_n(x) = -\frac{1}{2}x_n^2 + \frac{n}{3}x_n^3.$$

Initial guess: $x_0 = (1, 1, \ldots, 1)^T$.

Function 12 The trigexp function

$$f_1(x) = 3x_1^3 + 2x_2 - 5 + \sin(x_1 - x_2)\sin(x_1 + x_2),$$
$$f_i(x) = -x_{i-1}e^{x_{i-1}-x_i} + x_i\big(4 + 3x_i^2\big) + 2x_{i+1} + \sin(x_i - x_{i+1})\sin(x_i + x_{i+1}) - 8,$$
$$i = 2, 3, \ldots, n-1,$$
$$f_n(x) = -x_{n-1}e^{x_{n-1}-x_n} + 4x_n - 3.$$

Initial guess: $x_0 = (0, 0, \ldots, 0)^T$.

Function 13 The extended Freudentein and Roth function (n is even). For $i = 1, 2, \ldots, n/2$,

$$f_{2i-1}(x) = x_{2i-1} + \big((5 - x_{2i})x_{2i} - 2\big)x_{2i} - 13,$$
$$f_{2i}(x) = x_{2i-1} + \big((1 + x_{2i})x_{2i} - 14\big)x_{2i} - 29.$$

Initial guess: $x_0 = (6, 3, \ldots, 6, 3)^T$.

Function 14 The Troech problem

$$f_1(x) = 2x_1 + \varrho h^2 \sin h(\varrho x_1) - x_2,$$
$$f_i(x) = 2x_i + \varrho h^2 \sin h(\varrho x_i) - x_{i-1} - x_{i+1}, \quad i = 2, 3, \ldots, n-1,$$

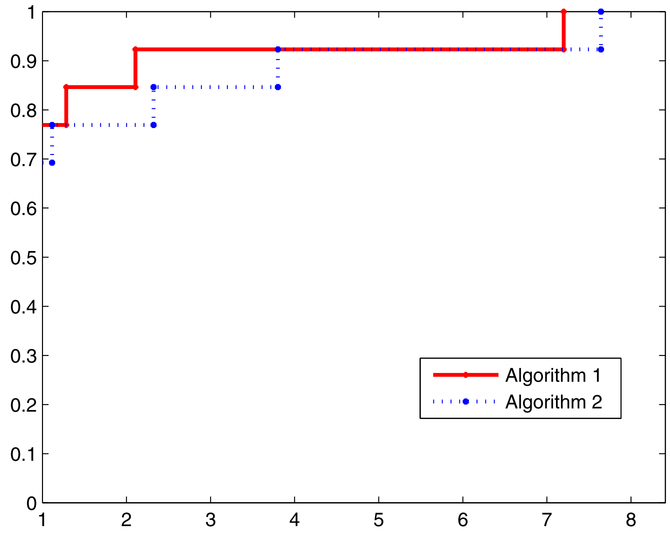

Figure 1 Performance profiles of the total number of iterations of two algorithms (*n* = 100).

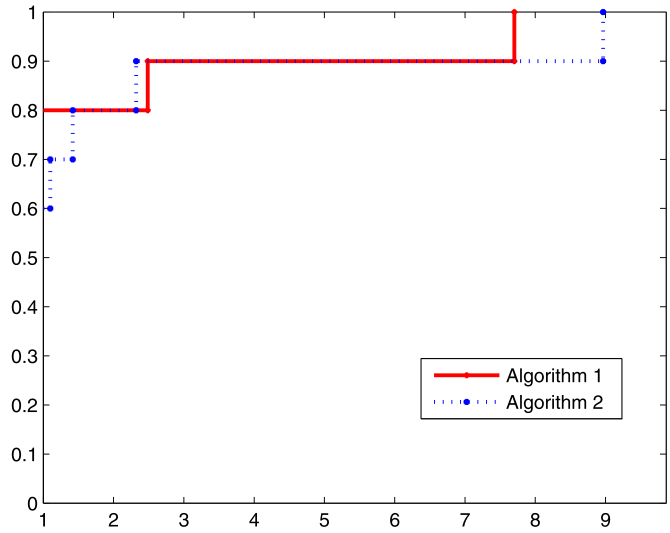

Figure 2 Performance profiles of the total number of iterations of two algorithms (*n* = 1,000).

$$f_n(x) = 2x_n + \varrho h^2 \sin h(\varrho x_n) - x_{n-1},$$

$$h = \frac{1}{n+1}, \qquad \varrho = 10.$$

Initial guess: $x_0 = (0, 0, \dots, 0)^T$.

In the experiments, the parameters are chosen as $\Delta_0 = 1$, $\bar{\Delta} = 10$, $\epsilon = 10^{-5}$, $\eta_1 = 0.001$, $\eta_2 = 0.75$, $\beta_1 = 0.5$, $\beta_2 = 2.0$, $M = 10$, $\eta_k = 1/(k+1)^2$, $\alpha = 0.5$, where ϵ is the stop criterion. The program is also stopped if the iteration number is larger than 5,000. We obtain d_k by (5) from the Dogleg method in [18]. The program is coded in MATLAB 2009a.

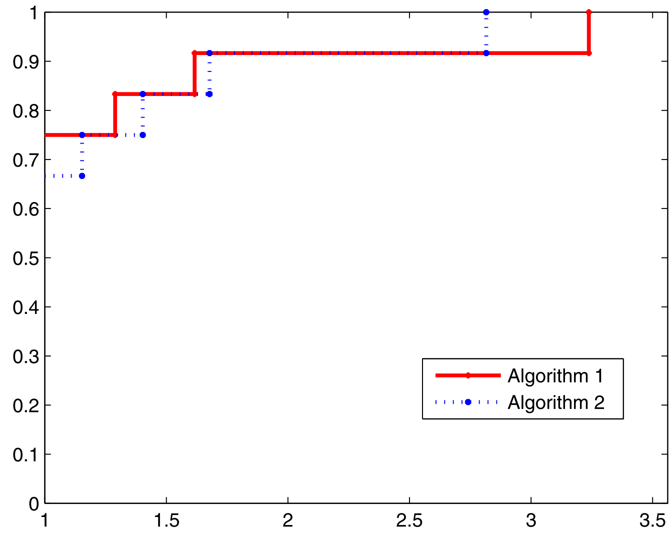

Figure 3 Performance profiles of the total number of iterations of two algorithms (*n* = 10,000).

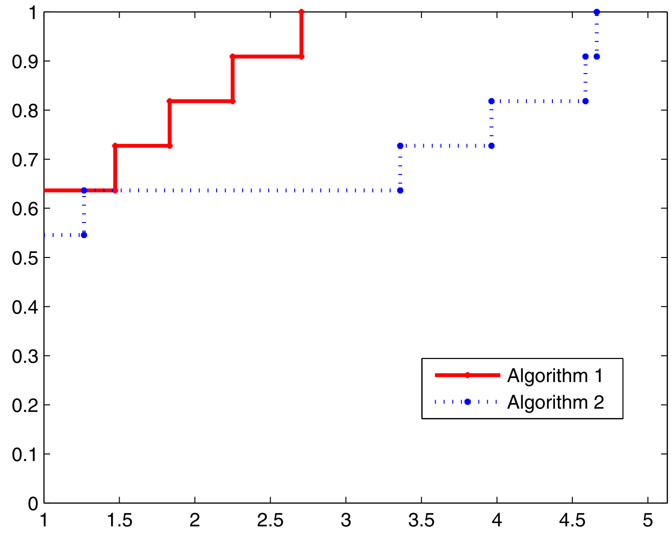

Figure 4 Performance profiles of the CPU time of two algorithms (*n* = 100).

To show the performance of two algorithms, we use the performance profile proposed by Dolan and Moré [19]. The dimensions of 14 test functions are 100, 1,000, 10,000. According to the numerical results, we plot two figures based on the total number of iterations and the CPU time, respectively.

Figure 1 shows that our algorithm is slightly better than Algorithm 2 on the total number of iterations for $n = 100$. Figures 2 and 3 indicate that two algorithms have no large discrepancies on the total number of iterations for $n = 1{,}000, 10{,}000$. From Figures 4-6, it is easy to see that our algorithm performs better than Algorithm 2 does on the CPU time for 14 test problems. Preliminary numerical results show that the performance of our algorithm is notable.

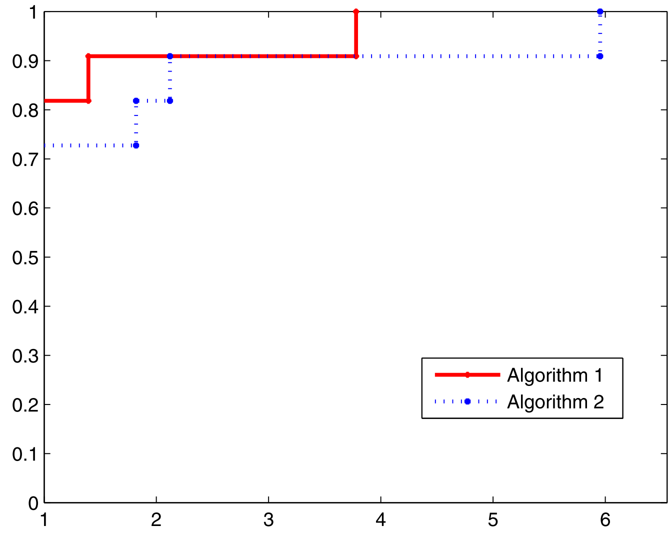

Figure 5 **Performance profiles of the CPU time of two algorithms (*n* = 1,000).**

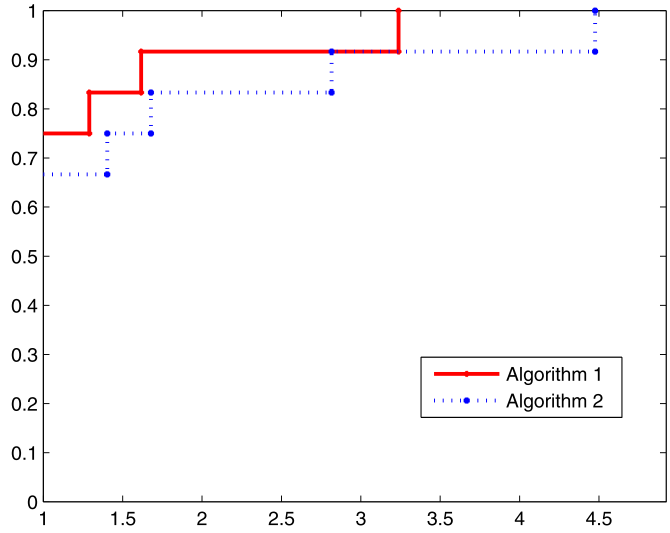

Figure 6 **Performance profiles of the CPU time of two algorithms (*n* = 10,000).**

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

The two authors contributed equally to the writing of this paper. All authors read and approved the final manuscript.

Author details

[1]College of Mathematics and Statistics, Hubei Engineering University, Xiaogan, 432000, China. [2]School of Mathematical Sciences/Information College, Huaibei Normal University, Huaibei, 235000, China.

Acknowledgements

We thank the reviewers and the editors for their valuable suggestions and comments which improve this paper greatly. This work is supported by the Science and Technology Foundation of the Department of Education of Hubei Province (D20152701) and the Foundations of Education Department of Anhui Province (KJ2016A651; 2014jyxm161).

Received: 27 February 2016 Accepted: 14 June 2016 Published online: 07 July 2016

References

1. La Cruz, W, Jose, MM, Marcos, R: Spectral residual method without gradient information for solving large-scale nonlinear systems of equations. Math. Comput. **75**(255), 1429-1448 (2006)
2. Bouaricha, A, Schnabel, RB: Tensor methods for large sparse systems of nonlinear equations. Math. Program. **82**(3), 377-400 (1998)
3. Bergamaschi, L, Moret, I, Zilli, G: Inexact quasi-Newton methods for sparse systems of nonlinear equations. Future Gener. Comput. Syst. **18**(1), 41-53 (2001)
4. La Cruz, W, Raydan, M: Nonmonotone spectral methods for large-scale nonlinear systems. Optim. Methods Softw. **18**(5), 583-599 (2003)
5. Li, Q, Li, D: A class of derivative-free methods for large-scale nonlinear monotone equations. IMA J. Numer. Anal. **31**, 1625-1635 (2011)
6. Yuan, GL, Zhang, MJ: A three-terms Polak-Ribiè-Polyak conjugate gradient algorithm for large-scale nonlinear equations. J. Comput. Appl. Math. **286**, 186-195 (2015)
7. Yuan, GL, Meng, ZH, Li, Y: A modified Hestenes and Stiefel conjugate gradient algorithm for large-scale nonsmooth minimizations and nonlinear equations. J. Optim. Theory Appl. **168**, 129-152 (2016)
8. Barzilai, J, Borwein, JM: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141-148 (1988)
9. Birgin, EG, Martinez, JM, Raydan, M: Inexact spectral projected gradient methods on convex sets. IMA J. Numer. Anal. **23**(4), 539-559 (2003)
10. Dai, YH, Zhang, H: Adaptive two-point stepsize gradient algorithm. Numer. Algorithms **27**(4), 377-385 (2001)
11. Dai, YH: Modified two-point stepsize gradient methods for unconstrained optimization. Comput. Optim. Appl. **22**(1), 103-109 (2002)
12. Raydan, M: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Control Optim. **7**(1), 26-33 (1997)
13. Yuan, GL, Lu, XW, Wei, ZX: BFGS trust-region method for symmetric nonlinear equations. J. Comput. Appl. Math. **230**, 44-58 (2009)
14. Yuan, YX: Trust region algorithm for nonlinear equations. Information **1**, 7-21 (1998)
15. Zhang, JL, Wang, Y: A new trust region method for nonlinear equations. Math. Methods Oper. Res. **58**(2), 283-298 (2003)
16. Yuan, GL, Wei, ZX, Liu, XW: A BFGS trust-region method for nonlinear equations. Computing **92**(4), 317-333 (2011)
17. Moré, JJ, Garbow, BS, Hillstrom, KE: Testing unconstrained optimization software. ACM Trans. Math. Softw. **7**(1), 17-41 (1981)
18. Wang, YJ, Xiu, NH: Theory and Algorithm for Nonlinear Programming. Shanxi Science and Technology Press, Xi'an (2004)
19. Dolan, ED, Moré, JJ: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201-213 (2002)