Bull. Aust. Math. Soc. **95** (2017), 500–511
doi:10.1017/S0004972717000168

A NEW DERIVATIVE-FREE CONJUGATE GRADIENT METHOD FOR LARGE-SCALE NONLINEAR SYSTEMS OF EQUATIONS

XIAOWEI FANG✉ and QIN NI

(Received 21 September 2016; accepted 29 December 2016; first published online 22 March 2017)

Abstract

We propose a new derivative-free conjugate gradient method for large-scale nonlinear systems of equations. The method combines the Rivaie–Mustafa–Ismail–Leong conjugate gradient method for unconstrained optimisation problems and a new nonmonotone line-search method. The global convergence of the proposed method is established under some mild assumptions. Numerical results using 104 test problems from the CUTEst test problem library show that the proposed method is promising.

2010 *Mathematics subject classification*: primary 90C56; secondary 90C30.

Keywords and phrases: large-scale nonlinear systems of equations, conjugate gradient method, nonmonotone line search.

1. Introduction

In this paper, we consider the nonlinear system of equations

$$F(x) = 0 \quad x \in \mathbb{R}^n, \tag{1.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable. Newton and quasi-Newton methods (see the survey [1]) are the most widely used methods to solve such problems because they have very attractive theoretical and practical properties. However, they are not usually suitable for large-scale nonlinear systems of equations because they require the solution of a large system of linear equations using a Jacobian matrix, or an approximation to it, at every iteration.

La Cruz and Raydan [6] presented the spectral algorithm for nonlinear equations (SANE). At each iteration, SANE systematically uses the residual $\pm F(x_k)$ as search direction. La Cruz *et al.* [5] proposed a new nonmonotone line-search technique and developed the derivative-free spectral residual method for solving large-scale nonlinear

This work was supported by the National Natural Science Foundation of China (11071117, 11274109), the Natural Science Foundation of Jiangsu Province (BK20141409), and the Natural Science Foundation of Huzhou University (KX21072).

systems of equations (DF-SANE). An extensive set of numerical experiments indicated that DF-SANE was competitive and frequently better than SANE and well-known Newton–Krylov methods for large-scale problems. Cheng and Li [3] extended the nonmonotone line search proposed by Zhang and Hager [15] to the spectral residual method for solving nonlinear systems of equations (N-DF-SANE).

Conjugate gradient methods are successful methods for solving unconstrained optimisation problems. They are particularly efficient for solving large-scale problems due to their simplicity and low storage (see [11]). Among these is the Polak–Ribière–Polyak method (PRP), which has many variants. Some recent examples of PRP conjugate gradient methods for which the numerical results are very encouraging are studied in [9, 13, 14].

The PRP conjugate gradient methods have been adapted to solve problem (1.1). Li [10] developed a derivative-free PRP method to solve large-scale nonlinear systems of equations with a nonmonotone line search (DF-PRP). Cheng *et al.* [4] proposed a family of derivative-free conjugate gradient methods for large-scale nonlinear systems of equations, based on a steepest descent algorithm (DF-SDCG). These methods come from two modified conjugate gradient methods (TTPRP [16] and TMPRP [2]).

Quite recently, Rivaie *et al.* [12] proposed a new class of nonlinear conjugate gradient methods for unconstrained optimisation problems (RMIL, named after its developers Rivaie, Mustafa, Ismail and Leong). Numerical results show that the RMIL conjugate gradient method is superior and more efficient when compared with other conjugate gradient methods for unconstrained optimisation problems (see [12, pages 11329–11331]).

Motivated by the efficiency of the RMIL conjugate gradient method [12], the structure of the spectral residual method for solving large-scale nonlinear systems of equations [5] and the ideas of the nonmonotone line search [3, 15], we propose here a derivative-free nonmonotone RMIL conjugate gradient method for solving large-scale nonlinear systems of equations (DF-RMIL).

This paper is organised as follows. In Section 2, we present the algorithm. In Section 3, we prove its convergence. In Section 4, we report some numerical results comparing the methods DF-SANE [5], N-DF-SANE [3], DF-SDCG [4] and DF-PRP [10] with the new method DF-RMIL.

The default norm used in this paper is the Euclidean norm.

2. Algorithm

We first consider the nonlinear conjugate gradient method for the unconstrained optimisation problem

$$\min_{x\in\mathbb{R}^n} f(x), \tag{2.1}$$

where the function f is assumed to be continuously differentiable from $\mathbb{R}^n$ into $\mathbb{R}$, and the gradient $\nabla f(x_k)$ is available. The nonlinear conjugate gradient method generates a sequence $\{x_k\}$ by the recursive relation

$$x_{k+1} = x_k + \alpha_k d_k \quad k = 0, 1, \ldots, \tag{2.2}$$

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

where α_k is the steplength and the search direction d_k is updated by

$$d_k = \begin{cases} -\nabla f(x_k) & \text{if } k = 0, \\ -\nabla f(x_k) + \beta_k d_{k-1} & \text{if } k \geq 1, \end{cases}$$

where β_k is a scalar.

In 2012, Rivaie *et al.* [12] proposed a new class of nonlinear conjugate gradient methods (RMIL), in which the parameter β_k is defined by

$$\beta_k = \beta_k^{\text{RMIL}} = \frac{\nabla f(x_k)^T(\nabla f(x_k) - \nabla f(x_{k-1}))}{\|d_{k-1}\|^2}.$$

The RMIL conjugate gradient method is generally believed to be an efficient method because it possesses the restart property. The numerical results in [12] show that this method is superior when compared with other nonlinear conjugate gradient methods for unconstrained optimisation problems.

The steplength α_k of (2.2) is obtained by a line-search rule. In 2004, Zhang and Hager [15] proposed a new nonmonotone line search method, using the weighted average function value C_k to replace the largest function value: that is,

$$C_{k+1} = \frac{\lambda_k Q_k C_k + f(x_{k+1})}{Q_{k+1}}, \tag{2.3}$$

where $Q_{k+1} = \lambda_k Q_k + 1, Q_0 = 1, C_0 = f(x_0), 0 \leq \lambda_k \leq 1$. Cheng and Li [3] extended (2.3) to a derivative-free nonmonotone line-search method, where C_k is given by

$$C_{k+1} = \frac{\lambda_k Q_k (C_k + \eta_k) + f(x_{k+1})}{Q_{k+1}}, \tag{2.4}$$

with $Q_{k+1} = \lambda_k Q_k + 1, Q_0 = 1, C_0 = f(x_0), 0 \leq \lambda_k \leq 1$, and the positive sequence $\{\eta_k\}$ satisfies $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$. In this paper, we propose a new nonmonotone line-search method which extends (2.3) and (2.4). The C_k are updated by the rule

$$C_{k+1} = \lambda_k (C_k + \eta_k) + (1 - \lambda_k) f(x_{k+1}), \tag{2.5}$$

with $C_0 = f(x_0), 0 \leq \lambda_k < 1$, and the positive sequence $\{\eta_k\}$ satisfies $\sum_{k=0}^{\infty} \eta_k = \eta < \infty$.

We now describe a derivative-free RMIL conjugate gradient method (DF-RMIL) for solving large-scale nonlinear systems of equations (1.1). The new method is based on the RMIL nonlinear conjugate gradient method for unconstrained optimisation [12], the derivative-free spectral residual method for solving large-scale nonlinear systems of equations [5] and the new nonmonotone line search method (2.5).

In the remainder of this paper, we let

$$f(x) = \tfrac{1}{2}\|F(x)\|^2. \tag{2.6}$$

Note that if, instead, we start with the problem (2.1), we can view the problem (1.1) as the first-order optimality condition of the problem (2.1), where $F(x)$ is the gradient of $f : \mathbb{R}^n \to \mathbb{R}$.

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

ALGORITHM 2.1 (DF-RMIL).
Step 0. Initialisations. Choose the initial point $x_0 \in \mathbb{R}^n$ and set $C_0 = f(x_0)$ (where f is defined by (2.6)). Suppose that $0 < \rho < 1$, $0 \le \lambda_{\min} < \lambda_{\max} < 1$ and $0 < \alpha_{\min} < \alpha_{\max}$. Select a positive sequence $\{\eta_k\}$ such that

$$\sum_{k=0}^{\infty} \eta_k = \eta < \infty. \tag{2.7}$$

Step 1. Determination of the search direction. Set $F_k = F(x_k)$. Compute d_k by

$$d_k = \begin{cases} -F_k & \text{if } k = 0, \\ -F_k + \beta_k^{\mathrm{ERMIL}} d_{k-1} & \text{if } k \ge 1, \end{cases} \tag{2.8}$$

where

$$\beta_k^{\mathrm{ERMIL}} = \frac{F_k^T (F_k - F_{k-1})}{\|d_{k-1}\|^2}.$$

Step 2. Determination of the steplength. Choose the steplength $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$.
Step 3. Nonmonotone line search. If

$$f(x_k + \alpha_k d_k) \le C_k + \eta_k - \gamma \alpha_k^2 f(x_k), \tag{2.9}$$

then set $x_{k+1} = x_k + \alpha_k d_k$ and go to step 4. Else if

$$f(x_k - \alpha_k d_k) \le C_k + \eta_k - \gamma \alpha_k^2 f(x_k), \tag{2.10}$$

then set $x_{k+1} = x_k - \alpha_k d_k$ and go to step 4. Otherwise set $\alpha_k = \rho \alpha_k$ and go to step 3.
Step 4. Check the stopping condition. If the stopping condition is not met, then go to step 5. Otherwise output x_{k+1}, $F(x_{k+1})$ and stop.
Step 5. Update the parameters. Choose $\lambda_k \in [\lambda_{\min}, \lambda_{\max}]$ and compute

$$C_{k+1} = \lambda_k (C_k + \eta_k) + (1 - \lambda_k) f(x_{k+1}). \tag{2.11}$$

Set $k = k + 1$ and go to step 1.

REMARK 2.2. In step 5, if $\lambda_k = 0$ for all k, then $C_{k+1} = f(x_{k+1})$. If $\lambda_k \to 1$ for all k, then $C_{k+1} \to f(x_0) + \sum_{i=0}^{k} \eta_i$.

The main differences between Algorithm 2.1 and the algorithms DF-SDCG [4] and DF-PRP [10] are as follows.

(i) The search direction of Algorithm 2.1 follows the approach used in the RMIL conjugate gradient method.

(ii) The nonmonotone line search used in the algorithms DF-SDCG and DF-PRP is $f(x_k \pm \alpha_k d_k) \le C_k + \eta_k - t_1 \alpha_k^2 \|d_k\|^2 - t_2 \alpha_k^2 f(x_k)$. In Algorithm 2.1, we use a simpler line-search technique ((2.9) or (2.10)) to benefit from the special structure of the RMIL conjugate gradient method and to assure the global convergence. This simpler line-search technique may make Algorithm 2.1 more efficient.

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

The following lemma shows that the line-search process of Algorithm 2.1 is well defined.

LEMMA 2.3. *If the sequences* $\{C_k\}, \{f(x_k)\}$ *are generated by Algorithm 2.1, then*

$$C_k \le C_{k-1} + \eta_{k-1} \tag{2.12}$$

and

$$f(x_k) \le C_k \le \max_{0\le i\le k} f(x_i) + \sum_{i=0}^{k-1} \eta_i, \tag{2.13}$$

for all $k > 0$.

PROOF. First, from step 3 of Algorithm 2.1,

$$f(x_k) \le C_{k-1} + \eta_{k-1}. \tag{2.14}$$

Using this and (2.11),

$$C_k \le \lambda_{k-1}(C_{k-1} + \eta_{k-1}) + (1 - \lambda_{k-1})(C_{k-1} + \eta_{k-1}) = C_{k-1} + \eta_{k-1}.$$

Second, from (2.11) and (2.14),

$$f(x_k) = \lambda_{k-1} f(x_k) + (1 - \lambda_{k-1}) f(x_k) \le \lambda_{k-1}(C_{k-1} + \eta_{k-1}) + (1 - \lambda_{k-1}) f(x_k) = C_k. \tag{2.15}$$

By successive applications of (2.11) and induction,

$$\begin{aligned} C_k &= \lambda_{k-1}(C_{k-1} + \eta_{k-1}) + (1 - \lambda_{k-1}) f(x_k) \\ &= \sum_{j=0}^{k-2} \Big(\prod_{i=j+1}^{k-1} \lambda_i \Big)(1 - \lambda_j) f(x_{j+1}) + \prod_{i=0}^{k-1} \lambda_i f(x_0) + (1 - \lambda_{k-1}) f(x_k) + \sum_{j=0}^{k-1} \Big(\prod_{i=j}^{k-1} \lambda_i \Big) \eta_j \\ &\le \max_{0\le i\le k} f(x_i) + \sum_{i=0}^{k-1} \eta_i, \end{aligned} \tag{2.16}$$

where we use $C_0 = f(x_0)$. From (2.15) and (2.16), we have (2.13). □

3. Convergence analysis

In order to prove the convergence of Algorithm 2.1, we make the following assumption.

ASSUMPTION 3.1.

(1) The level set $\Omega = \{x \mid f(x) \le f(x_0) + \eta\}$ is bounded, where η is a positive constant such that $\sum_{k=0}^{\infty} \eta_k \le \eta$.

(2) The function $F(x)$ is Lipschitz continuous in some neighbourhood Γ of Ω: namely, there exists a positive constant L such that

$$\|F(x) - F(y)\| \le L\|x - y\| \quad \text{for all } x, y \in \Gamma.$$

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

Assumption 3.1 implies that there is a positive constant δ such that

$$\|F(x)\| \le \delta \quad \text{for all } x \in \Omega.$$

LEMMA 3.2. *The sequence $\{x_k\}$ generated by Algorithm 2.1 is contained in Ω.*

PROOF. By step 3 of Algorithm 2.1, (2.12), (2.7) and induction,

$$\begin{aligned} f(x_{k+1}) &\le C_k + \eta_k \le C_{k-1} + \eta_{k-1} + \eta_k \le \cdots \\ &\le f(x_0) + \sum_{i=0}^{k} \eta_k \le f(x_0) + \eta. \end{aligned} \tag{3.1}$$

Thus the assertion follows from the definition of Ω. □

LEMMA 3.3. *Let $\{x_k\}$ be the sequence generated by Algorithm 2.1. If Assumption 3.1 holds, then*

$$\lim_{k\to\infty} \alpha_k^2 f(x_k) = 0. \tag{3.2}$$

PROOF. By step 3 of Algorithm 2.1,

$$f(x_{k+1}) \le C_k + \eta_k - \gamma\alpha_k^2 f(x_k) \quad \text{for all } k \ge 0.$$

Using this and (2.11),

$$C_{k+1} = \lambda_k(C_k + \eta_k) + (1 - \lambda_k) f(x_{k+1}) \le C_k + \eta_k - (1 - \lambda_k)\gamma\alpha_k^2 f(x_k).$$

By (2.6), (2.7) and Assumption 3.1,

$$\sum_{i=0}^{\infty} (1 - \lambda_k)\gamma\alpha_k^2 f(x_k) < \infty.$$

Because $0 \le \lambda_{\min} \le \lambda_k \le \lambda_{\max} < 1$ and $\gamma > 0$, this implies that $\lim_{k\to\infty} \alpha_k^2 f(x_k) = 0$. □

LEMMA 3.4. *Suppose that Assumption 3.1 holds and let $\{x_k\}$ and $\{d_k\}$ be the sequences generated by Algorithm 2.1. Then*

$$|\beta_k^{\mathrm{ERMIL}}| \, \|d_{k-1}\| \le \delta L \alpha_{k-1} \quad \textit{for all } k \ge 1$$

and

$$\|d_k\| \le \delta(1 + L\alpha_{k-1}) \quad \textit{for all } k \ge 1. \tag{3.3}$$

Moreover, if there is a constant $\varepsilon > 0$ such that

$$\|F_k\| \ge \varepsilon \quad \textit{for all } k \ge 0, \tag{3.4}$$

then

$$\lim_{k\to\infty} |\beta_k^{\mathrm{ERMIL}}| \, \|d_{k-1}\| = 0. \tag{3.5}$$

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

PROOF. First, from step 1 of Algorithm 2.1 and Assumption 3.1,

$$|\beta_k^{\mathrm{ERMIL}}|\,\|d_{k-1}\| = \frac{|F_k^T(F_k - F_{k-1})|}{\|d_{k-1}\|^2}\|d_{k-1}\| \le \frac{L\alpha_{k-1}\|F_k\|\,\|d_{k-1}\|}{\|d_{k-1}\|} \le \delta L\alpha_{k-1}. \tag{3.6}$$

Second, from step 1 of Algorithm 2.1, Assumption 3.1 and (3.6),

$$\|d_k\| \le \|F_k\| + |\beta^{\mathrm{ERMIL}}|\,\|d_{k-1}\| \le \delta(1 + L\alpha_{k-1}).$$

Third, using (2.6), (3.2) and (3.4) and by taking limits in (3.6),

$$\lim_{k\to\infty} |\beta_k^{\mathrm{ERMIL}}|\,\|d_{k-1}\| \le \lim_{k\to\infty} \frac{\delta L\alpha_{k-1}\|F_{k-1}\|}{\|F_{k-1}\|} = \lim_{k\to\infty} \frac{\delta L\sqrt{2\alpha_{k-1}^2 f(x_{k-1})}}{\|F_{k-1}\|} = 0. \qquad \square$$

THEOREM 3.5. *Suppose Assumption 3.1 holds and let $\{x_k\}$ and $\{d_k\}$ be the sequences generated by the Algorithm 2.1. Then either*

$$\liminf_{k\to\infty} \|F_k\| = 0, \tag{3.7}$$

or every limit point x^* of the sequence $\{x_k\}$ satisfies

$$F(x^*)^T J(x^*)F(x^*) = 0, \tag{3.8}$$

where $J(x^*)$ denotes the Jacobian of F at x^*. In particular, if F or $-F$ is strictly monotone, then every bounded subsequence of $\{x_k\}$ converges to the solution of (1.1).

PROOF. Let x^* be an arbitrary limit point of $\{x_k\}$ and let K_1 be an infinite sequence of indices such that

$$\lim_{k\in K_1, k\to\infty} x_k = x^*. \tag{3.9}$$

From (2.6) and (3.2),

$$\lim_{k\in K_1, k\to\infty} \alpha_k^2 f(x_k) = \lim_{k\in K_1, k\to\infty} \tfrac{1}{2}\alpha_k^2\|F_k\|^2 = 0. \tag{3.10}$$

Then, we have two cases.

Case 1. If

$$\limsup_{k\in K_1, k\to\infty} \alpha_k \ne 0,$$

then there is an infinite sequence of indices $K_2 \in K_1$ such that $\{\alpha_k\}_{K_2}$ is bounded away from zero. From (3.10),

$$\lim_{k\in K_2, k\to\infty} \|F_k\| = 0.$$

Since F is continuous and $\lim_{k\in K_2, k\to\infty} x_k = x^*$, we have (3.7).

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

Case 2. Suppose that

$$\lim_{k\in K_1, k\to\infty} \alpha_k = 0. \tag{3.11}$$

From step 3 of Algorithm 2.1, $\rho^{-1}\alpha_k$ satisfies neither (2.9) nor (2.10) when $k \in K_1$ is sufficiently large. Thus

$$f(x_k + \rho^{-1}\alpha_k d_k) > C_k + \eta_k - \gamma\rho^{-2}\alpha_k^2 f(x_k) \tag{3.12}$$

and

$$f(x_k - \rho^{-1}\alpha_k d_k) > C_k + \eta_k - \gamma\rho^{-2}\alpha_k^2 f(x_k). \tag{3.13}$$

By (3.12), (2.13) and (3.1),

$$\frac{f(x_k + \rho^{-1}\alpha_k d_k) - f(x_k)}{\alpha_k} > -\gamma\rho^{-2}\alpha_k(f(x_0) + \eta).$$

Using the mean-value theorem and (2.6), there exists $\xi_k \in (0, 1)$ such that

$$\rho^{-1}F(x_k + \xi_k\rho^{-1}\alpha_k d_k)^T J(x_k + \xi_k\rho^{-1}\alpha_k d_k)d_k > -\gamma\rho^{-2}\alpha_k(f(x_0) + \eta). \tag{3.14}$$

Substituting (2.8) into (3.14) gives

$$\rho^{-1}F(x_k + \xi_k\rho^{-1}\alpha_k d_k)^T J(x_k + \xi_k\rho^{-1}\alpha_k d_k)(-F_k + \beta_k^{\mathrm{ERMIL}} d_{k-1}) > -\gamma\rho^{-2}\alpha_k(f(x_0) + \eta). \tag{3.15}$$

By (3.3) and (3.11) and since $\alpha_k \le \alpha_{\max}$,

$$\lim_{k\in K_1, k\to\infty} \|\xi_k\rho^{-1}\alpha_k d_k\| \le \lim_{k\in K_1, k\to\infty} \xi_k\rho^{-1}\alpha_k\delta(1 + L\alpha_{k-1}) = 0. \tag{3.16}$$

Using (3.9), (3.16) and (3.5) and by taking limits in (3.15),

$$F(x^*)^T J(x^*)F(x^*) \le 0. \tag{3.17}$$

Using (3.13) and proceeding in the same way, yields

$$F(x^*)^T J(x^*)F(x^*) \ge 0. \tag{3.18}$$

The inequalities (3.17) and (3.18) imply (3.8). □

4. Numerical experiments

In this section, we discuss numerical test results for Algorithm 2.1 (DF-RMIL) and compare these with DF-SANE [5], N-DF-SANE [3], DF-SDCG [4] and DF-PRP [10]. The benchmark problems in our experiments are the 104 problems in the CUTEst library [8]. The dimensions of the benchmark problems vary from 50 to 100 000. Our tests were performed on a PC (Intel Core i3 CPU M 380 @2.53GHz 2.53GHz, 2 GB RAM) with Ubuntu Kylin 14.04 using MATLAB R2012a(7.14) 64-bit(glnxa64). All numerical results are listed on the website http://url.cn/43DkkxW.

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

To compare our method with the other methods, we use performance profiles as described in Dolan and Moré [7]. We give three separate comparisons, made on the basis of the number of iterations, the number of function evaluations and the CPU time required to solve the problem. Let P be the set of benchmark problems and let S be the set of algorithms. We define $t_{p,s}$ to be the number of iterations (or the number of function evaluations, or the CPU time in seconds) required to solve the problem $p \in P$ by algorithm $s \in S$. Comparison of each of the three measures is based on the performance ratio defined by

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}}.$$

The performance profile is

$$\rho_s(\tau) = \frac{|\{p \in P : \log_2(r_{p,s}) \le \tau\}|}{|P|},$$

where $|P|$ is the number of benchmark problems. We use the termination condition

$$\frac{\|F(x_k)\|}{\sqrt{n}} \le e_a + e_r \frac{\|F(x_0)\|}{\sqrt{n}},$$

where $e_a = 10^{-5}, e_r = 10^{-4}$ and n is the number of variables of the test problem. This termination condition comes from [5].

The parameters used for each of the five methods are as follows.

DF-SANE. nexp $= 2$, $\sigma_{\min} = 10^{-10}$, $\sigma_{\max} = 10^{10}$, $\sigma_0 = 1, \tau_{\min} = 0.1$, $\tau_{\max} = 0.5$, $\gamma = 10^{-4}, M = 10, \eta_k = \|F(x_0)\|/(1+k)^2$.

N-DF-SANE. $\eta_k = 0.85$, $\sigma_{\min} = 10^{-10}$, $\sigma_{\max} = 10^{10}$, $\rho_{\min} = 0.1$, $\rho_{\max} = 0.5$, $\gamma = 10^{-4}$, $\epsilon_k = \|F(x_0)\|/(1+k)^2$.

DF-SDCG. $\rho_{\min} = 0.1, \rho_{\max} = 0.5, \sigma_{\min} = 10^{-10}, \sigma_{\max} = 10^{10}, \gamma_1 = \gamma_2 = 10^{-4}, \lambda_k = 0.5$, $M = 10, \epsilon_k = \|F(x_0)\|/(1+k)^2$.

DF-PRP. $\rho = 0.5$, $\lambda_k = 0.6$, $\alpha_{\min} = 10^{-10}$, $\alpha_{\max} = 10^{10}$, $t_1 = t_2 = 10^{-4}$, $\eta_k = \|F(x_0)\|/(1+k)^2$.

DF-RMIL. $\rho = 0.5$, $\lambda_k = 0.5$, $\alpha_{\min} = 10^{-10}$, $\alpha_{\max} = 10^{10}$, $\gamma_k = 10^{-4}$ and $\eta_k = \|F(x_0)\|/(1+k)^2$.

We choose the steplength α_k in step 2 of DF-RMIL as in [5] and [4]: namely,

$$\alpha_k = \begin{cases} \sigma & \text{if } \sigma \in [\alpha_{\min}, \alpha_{\max}], \\ 1 & \text{if } \sigma \notin [\alpha_{\min}, \alpha_{\max}] \text{ and } \|d_k\| > 1, \\ \|d_k\|^{-1} & \text{if } \sigma \notin [\alpha_{\min}, \alpha_{\max}] \text{ and } 10^{-5} \le \|d_k\| \le 1, \\ 10^5 & \text{if } \sigma \notin [\alpha_{\min}, \alpha_{\max}] \text{ and } \|d_k\| < 10^{-5}, \end{cases}$$

where $\sigma = -F_k^T d_k / d_k^T z_k, z_k = (F(x_k + \epsilon d_k) - F(x_k))\epsilon$ and $\epsilon = 10^{-8}$.

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

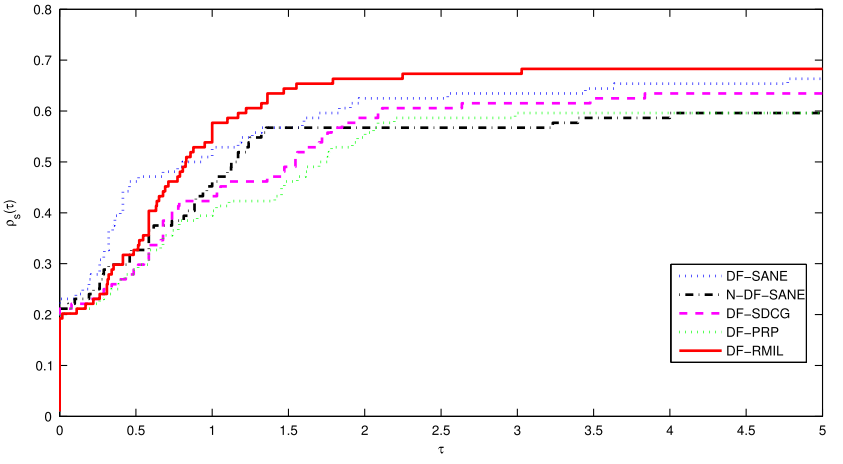

FIGURE 1. Iteration performance profiles on a $\log_2$ scale.

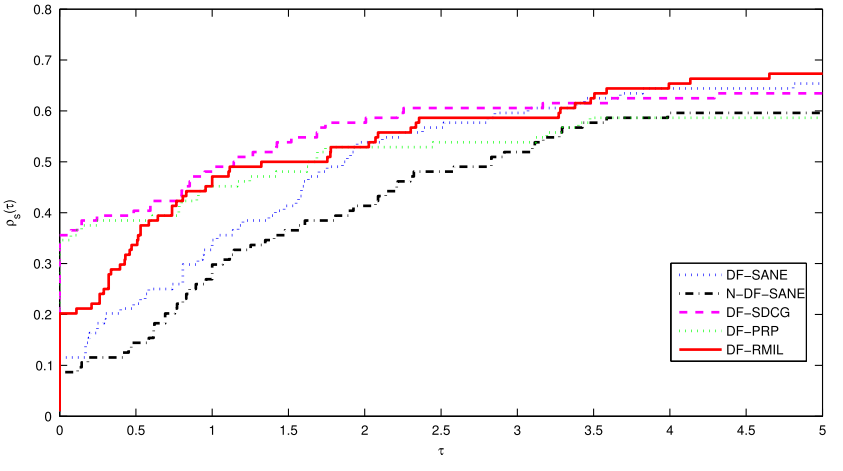

FIGURE 2. Function evaluation performance profiles on a $\log_2$ scale.

Figure 1 shows the performance profiles for the number of iterations for the five methods. DF-RMIL outperforms the other methods when $\tau \geq 0.8$. Figure 2 shows the performance profiles for the number of function evaluations. When $\tau < 3.4$, DF-SDCG uses the smallest number of function evaluations, but DF-RMIL gives the best results when $\tau \geq 3.4$. Figure 3 shows the performance profiles for CPU time. Here, DF-RMIL gives better results than the other methods for $\tau \geq 0.8$ as it solves a higher percentage of problems for a given CPU time. On the whole, DF-RMIL may be a competitive method.

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

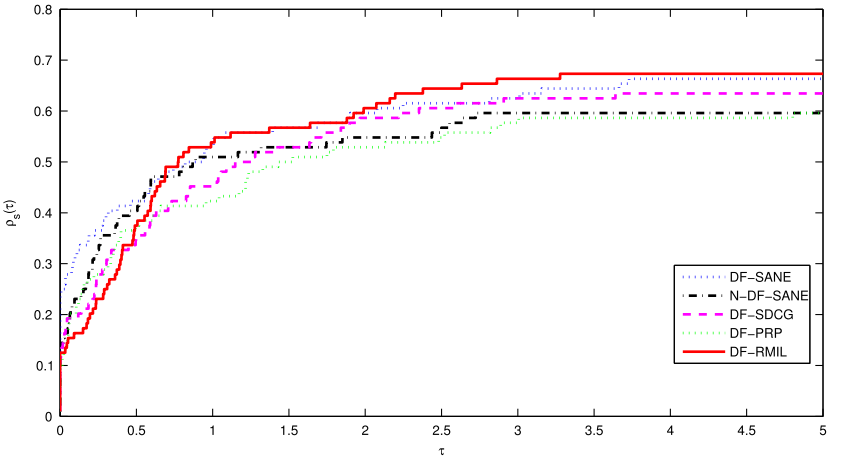

FIGURE 3. CPU time performance profiles on a $\log_2$ scale.

Acknowledgement

We thank the editor and the anonymous referee for their constructive suggestions which improved the paper greatly.

References

[1] M. Al-Baali, E. Spedicato and F. Maggioni, 'Broyden's quasi-Newton methods for a nonlinear system of equations and unconstrained optimization: a review and open problems', *Optim. Methods Softw.* **29** (2014), 937–954.

[2] W. Cheng, 'A two-term PRP-based descent method', *Numer. Funct. Anal. Optim.* **28** (2007), 1217–1230.

[3] W. Cheng and D. H. Li, 'A derivative-free nonmonotone line search and its application to the spectral residual method', *IMA J. Numer. Anal.* **29** (2009), 814–825.

[4] W. Cheng, Y. Xiao and Q. J. Hu, 'A family of derivative-free conjugate gradient methods for large-scale nonlinear systems of equations', *J. Comput. Appl. Math.* **224** (2009), 11–19.

[5] W. L. Cruz, J. Martínez and M. Raydan, 'Spectral residual method without gradient information for solving large-scale nonlinear systems of equations', *Math. Comp.* **75** (2006), 1429–1448.

[6] W. L. Cruz and M. Raydan, 'Nonmonotone spectral methods for large-scale nonlinear systems', *Optim. Methods Softw.* **18** (2003), 583–599.

[7] E. D. Dolan and J. J. Moré, 'Benchmarking optimization software with performance profiles', *Math. Program.* **91** (2001), 201–213.

[8] N. I. M. Gould, D. Orban and Ph. L. Toint, 'CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization', *Comput. Optim. Appl.* **60** (2015), 545–557.

[9] G. Li, C. M. Tang and Z. X. Wei, 'New conjugacy condition and related new conjugate gradient methods for unconstrained optimization', *J. Comput. Appl. Math.* **202**(2) (2007), 523–539.

[10] L. Min, 'A derivative-free PRP method for solving large-scale nonlinear systems of equations and its global convergence', *Optim. Methods Softw.* **29** (2014), 503–514.

[11] E. Polak, *Optimization: Algorithms and Consistent Approximations* (Springer, New York, 2012).

[12] M. Rivaie, M. Mamat, L. W. June and I. Mohd, 'A new class of nonlinear conjugate gradient coefficients with global convergence properties', *Appl. Math. Comput.* **218** (2012), 11323–11332.

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press

[13] Z. X. Wei, G. Li and L. Q. Qi, 'Global convergence of the Polak–Ribière–Polyak conjugate gradient method with inexact line searches for nonconvex unconstrained optimization problems', *Math. Comp.* **77** (2008), 2173–2193.
[14] G. Yu, L. Guan and G. Li, 'Global convergence of modified Polak–Ribière–Polyak conjugate gradient methods with sufficient descent property', *J. Ind. Manag. Optim.* **4**(3) (2008), 565–579.
[15] H. Zhang and W. W. Hager, 'A nonmonotone line search technique and its application to unconstrained optimization', *SIAM J. Optim.* **14** (2004), 1043–1056.
[16] L. Zhang, W. Zhou and D. H. Li, 'A descent modified Polak–Ribière–Polyak conjugate gradient method and its global convergence', *IMA J. Numer. Anal.* **26** (2006), 629–640.

XIAOWEI FANG, College of Sciences,
Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China
and
Department of Mathematics, Huzhou University,
Huzhou 313000, China
e-mail: fangxiaowei@163.com

QIN NI, College of Sciences,
Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China
e-mail: niqfs@nuaa.edu.cn

https://doi.org/10.1017/S0004972717000168 Published online by Cambridge University Press