Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 1425857, 11 pages
https://doi.org/10.1155/2017/1425857

Research Article

A Modified Nonlinear Conjugate Gradient Method for Engineering Computation

Tiefeng Zhu,[1,2] Zaizai Yan,[1] and Xiuyun Peng[1]

[1]*Science College, Inner Mongolia University of Technology, Hohhot 010051, China*
[2]*Department of Information Engineering, College of Youth Politics, Inner Mongolia Normal University, Hohhot 010051, China*

Correspondence should be addressed to Zaizai Yan; zz.yan@163.com

Received 6 July 2016; Revised 6 November 2016; Accepted 8 December 2016; Published 11 January 2017

Academic Editor: Yakov Strelniker

A general criterion for the global convergence of the nonlinear conjugate gradient method is established, based on which the global convergence of a new modified three-parameter nonlinear conjugate gradient method is proved under some mild conditions. A large amount of numerical experiments is executed and reported, which show that the proposed method is competitive and alternative. Finally, one engineering example has been analyzed for illustrative purposes.

1. Introduction

Unconstrained optimization methods are widely used in the fields of nonlinear dynamic systems and engineering computation to obtain the numerical solution of the optimal control problem [1–4]. In this paper, we consider the unconstrained optimization problem:

$$\min \quad f(x), \quad x \in R^n, \tag{1}$$

where $f : R^n \to R$ is a continuously differentiable function. The nonlinear conjugate gradient (CG) method is highly useful for solving this kind of problems because of its simplicity and its very low memory requirement [1]. The iterative formula of the CG methods is given by

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

where $\alpha_k > 0$ is step length which is obtained by carrying out some linear search, such as exact or inexact line search. In practical computation, exact line search is consumption time and the workload is very large, so we usually take the following inexact line search (see [5–7]). Usually, a major inexact line search is the strong Wolfe-Powell line search. The strong Wolfe-Powell line search is to find the step-length α_k in (2) satisfying

$$f(x_k + \alpha_k d_k) - f(x_k) \le \delta \alpha_k g_k^T d_k, \tag{3}$$

$$\sigma g_k^T d_k \le g(x_k + \alpha_k d_k)^T d_k \le -\sigma g_k^T d_k, \tag{4}$$

where $0 < \delta < 1/2$ and $\delta < \sigma < 1$. In this paper, the following modified Wolfe-Powell line search is to find the step-length α_k in (2) satisfying (3) and the following:

$$\sigma g_k^T d_k \le g(x_k + \alpha_k d_k)^T d_k \le 0, \tag{5}$$

and d_k is the search direction defined by

$$d_k = \begin{cases} -g_1 & k = 1 \\ -g_k + \beta_k d_{k-1} & k \ge 2, \end{cases} \tag{6}$$

where g_k denotes the gradient $\nabla f(x_k)$, β_k is scalar, and β_k is chosen so that d_k becomes the kth conjugate direction. There have been many well-known formulae for the scalar β_k, for example,

$$\beta_k^{\mathrm{FR}} = \frac{\|g_k\|^2}{\|g_{k-1}\|^2} \tag{7}$$

(Fletcher-Reeves [8], 1964),

$$\beta_k^{\mathrm{PRP}} = \frac{g_k^T y_{k-1}}{\|g_{k-1}\|^2} \tag{8}$$

(Polak-Ribiere-Polyak [9], 1969),

$$\beta_k^{\mathrm{DY}} = \frac{\|g_k\|^2}{d_{k-1}^T y_{k-1}} \tag{9}$$

(Dai-Yuan [10], 1999), and other formulae (e.g., [11–13]), where $\|\cdot\|$ is the Euclidean norm of vectors, $y_{k-1} = g_k - g_{k-1}$, and "T" stand for the transpose. These methods are generally regarded as very efficient conjugate gradient methods in practical computation.

In recent decades, in order to obtain the CG method which has not only good convergence property but also excellent computation, many researchers have studied the CG method extensively and obtained some improved methods with good properties [14–20]. Li and Feng [21] gave the modified CG method which generates a sufficient descent direction and showed its global convergence property under the strong Wolfe-Powell conditions. Dai and Wen [22] gave a scaled conjugate gradient method. They proved its global convergence property under the strong Wolfe-Powell conditions. Al-Baali [23] proved that the FR method satisfies the sufficient descent condition and converges globally for general objective functions if the strong Wolfe-Powell line search is used. Dai and Yuan [24] also introduced a formula for β_k:

$$\beta_k = \frac{(1-\lambda_k)\|g_k\|^2 + \lambda_k g_k^T y_{k-1}}{(1-\mu_k-\omega_k)\|g_{k-1}\|^2 + \mu_k d_{k-1}^T y_{k-1} - \omega_k d_{k-1}^T g_{k-1}}, \tag{10}$$

where $0 \le \lambda_k \le 1$, $0 \le \mu_k \le 1$, and $0 \le \omega_k \le 1-\mu_k$. Because

$$\begin{aligned} g_k^T y_{k-1} &= \|g_k\|^2 - g_k^T g_{k-1}, \\ g_{k-1}^T d_{k-1} &= -\|g_{k-1}\|^2 + \beta_{k-1} g_{k-1}^T d_{k-2}, \end{aligned} \tag{11}$$

we can rewrite (10) as

$$\beta_k^* = \frac{\|g_k\|^2 - \lambda_k g_k^T g_{k-1}}{\|g_{k-1}\|^2 + \mu_k d_{k-1}^T g_k - \omega_k \beta_{k-1} g_{k-1}^T d_{k-2}}. \tag{12}$$

This formula includes the above three classes of CG method as an extreme case, and global convergence of three parameters of CG method was proved under strong Wolfe-Powell line search. If $\omega_k = 0$, then the family reduces to the two-parameter family of conjugate gradient methods in [25]. Further, if $\lambda_k = 0$, $\mu_k = \mu$, and $\omega_k = 0$, then the family reduces to the one-parameter family in [26]. Therefore, the three-parameter family has the one-parameter family in [26] and the two-parameter family in [25] as its subfamilies. In addition, some hybrid methods can also be regarded as special cases of the three-parameter family [24]. Above many modified CG methods, global convergence was obtained under strong Wolfe-Powell line search; however, in this paper, we further study the CG method, and our main aim is to improve the numerical performance of the CG method while keeping its global convergence with modified Wolfe-Powell line search.

This paper is organized as follows. We first present a criterion for the global convergence of CG method in the next section. In Section 3, we propose a new modified three-parameter conjugate gradient method and establish global convergence results for relative algorithm under modified Wolfe-Powell line search. The preliminary numerical results are contained in Section 4. One engineering example is analyzed for illustration in Section 5. Finally, conclusions appear in Section 6.

2. A Criterion for the Global Convergence of CG Method

In this section, first, we adopt the following assumption used commonly in the research literatures.

Assumption 1. The function f is LC in a neighborhood N of the level set $\Omega := \{x \in R^n \mid f(x) \le f(x_1)\}$ and Ω is bounded. Here, by LC, we mean that the gradient $\nabla f(x_k)$ is Lipschitz continuous with modulus L; that is, there exists $L > 0$ such that

$$\|g(x) - g(y)\| \le L\|x - y\| \quad \text{for any } x, y \in N. \tag{13}$$

Lemma 2 (Zoutendijk condition [27]). *Suppose that Assumption 1 holds, x_k is given by (2) and (6), and α_k is obtained by the modified Wolfe-Powell line search ((3), (5)), while the direction d_k satisfies $g_k^T d_k < 0$. Then,*

$$\sum_{k=1}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < +\infty. \tag{14}$$

Lemma 3 (see [28]). *Suppose that $\lambda_k(>0)$ and C are constants; if $\{a_i\}$ satisfy $\sum_{i=1}^k a_i \ge \lambda_k + C$, then $\sum_{i\ge1}(a_i^2/i) = +\infty$ and $\sum_{k\ge1}(a_k^2/\sum_{i=1}^k a_i) = +\infty$.*

Theorem 4. *Suppose that the objective function satisfies Assumption 1 and that x_k is given by (2) and (6), where α_k satisfies the modified Wolfe-Powell (3) and (5), and $|\beta_k| \le \|g_k\|^2/\|g_{k-1}\|^2$; then, either $g_k = 0$ holds for certain k or*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{15}$$

Proof. Suppose, by contradiction, that the stated conclusion is not true. Then, in view of $\|g_k\| > 0$, there exits a constant $\varepsilon > 0$, such that

$$\|g_k\|^2 \ge \varepsilon, \quad k = 1, 2, \ldots. \tag{16}$$

From (6), we have

$$d_k + g_k = \beta_k d_{k-1}, \quad k = 2, 3, \ldots. \tag{17}$$

By multiplying $d_k + g_k$ on both sides of (17), then we have

$$\|d_k\|^2 = -\|g_k\|^2 - 2g_k^T d_k + (\beta_k)^2 \|d_{k-1}\|^2. \tag{18}$$

Let $t_k = \|d_k\|^2/\|g_k\|^4$ and $r_k = -g_k^T d_k/\|g_k\|^2$; then,

$$t_k = -\frac{1}{\|g_k\|^2} - \frac{2g_k^T d_k}{\|g_k\|^4} + (\beta_k)^2 \frac{\|d_{k-1}\|^2}{\|g_k\|^4}. \tag{19}$$

Thus, from (19) and $|\beta_k| \le \|g_k\|^2/\|g_{k-1}\|^2$, we get

$$\begin{aligned} t_k &\le -\frac{1}{\|g_k\|^2} + \frac{2r_k}{\|g_k\|^2} + \frac{\|g_k\|^4}{\|g_{k-1}\|^4} \cdot \frac{\|d_{k-1}\|^2}{\|g_k\|^4} \\ &= t_{k-1} - \frac{1}{\|g_k\|^2} + \frac{2r_k}{\|g_k\|^2}. \end{aligned} \tag{20}$$

Note that $t_1 = 1/\|g_1\|^2$ and $r_1 = 1$; then, it follows from (20) that

$$t_k \le -\sum_{i=1}^{k} \frac{1}{\|g_i\|^2} + 2\sum_{i=1}^{k} \frac{r_i}{\|g_i\|^2}. \tag{21}$$

From Assumption 1, it follows that there exists constant $M(>0)$, such that

$$\|g(x)\|^2 \le M, \quad \forall x \in \Omega, \tag{22}$$

and from (16), (21), and (22), we get

$$t_k \le -\frac{k}{M} + \frac{2}{\varepsilon}\sum_{i=1}^{k} |r_i|. \tag{23}$$

From the above, it is obvious that

$$t_k \le \frac{2}{\varepsilon}\sum_{i=1}^{k} |r_i|. \tag{24}$$

From the other side, for $t_k \ge 0$ from (23), it follows that

$$\sum_{i=1}^{k} |r_i| \ge \frac{k\varepsilon}{2M}. \tag{25}$$

From (24) and (25) and Lemma 3, we have

$$\sum_{k\ge1} \frac{(g_k^T d_k)^2}{\|d_k\|^2} = \sum_{k\ge1} \frac{r_k^2}{t_k} \ge \frac{\varepsilon}{2}\sum_{k\ge1} \frac{r_k^2}{\sum_{k\ge1} r_i} = +\infty, \tag{26}$$

what contradicts Lemma 2. Therefore, the global convergence is proved. □

3. The Global Convergence for the New Formula and Algorithm Frame

3.1. The New Formula and the Corresponding Properties. Based on formula (10), we put forward a new formula of β_k:

$$\beta_k^{\text{new}} = \frac{\max\{0, \min\{(1-\lambda_k)\|g_k\|^2, \lambda_k g_k^T (g_{k-1} - d_{k-1})\}\}}{(1-\mu_k-\omega_k)\|g_{k-1}\|^2 + \mu_k g_k^T d_{k-1} - (1-\lambda_k+\mu_k+\omega_k) g_{k-1}^T d_{k-1}}, \tag{27}$$

where $1/2 < \lambda_k \le 1, 0 \le \mu_k \le 1, 0 \le \omega_k \le 1$, and $\lambda_k \ge \mu_k + \omega_k$. Because of possible negative values of

$$\min\{(1-\lambda_k)\|g_k\|^2, \lambda_k g_k^T (g_{k-1} - d_{k-1})\}, \tag{28}$$

we use the maximum function to truncate zero and

$$\min\{(1-\lambda_k)\|g_k\|^2, \lambda_k g_k^T (g_{k-1} - d_{k-1})\}. \tag{29}$$

Using the equality

$$g_{k-1}^T d_{k-1} = -\|g_{k-1}\|^2 + \beta_{k-1} g_{k-1}^T d_{k-2}, \tag{30}$$

we can rewrite the denominator of (27) as

$$\begin{aligned} &\|g_{k-1}\|^2 + \mu_k g_k^T d_{k-1} - \omega_k \beta_{k-1} g_{k-1}^T d_{k-2} \\ &\quad + (1-\lambda_k)\|g_{k-1}\|^2 - (1-\mu_k+\omega_k)\beta_{k-1} g_{k-1}^T d_{k-2}. \end{aligned} \tag{31}$$

When

$$(1-\lambda_k)\|g_{k-1}\|^2 = (1-\mu_k+\omega_k)\beta_{k-1} g_{k-1}^T d_{k-2}, \tag{32}$$

then the denominator of β_k^{new} given by (27) reduces to the denominator of β_k^*. On the other hand, when

$$0 < \lambda_k g_k^T (g_{k-1} - d_{k-1}) < (1-\lambda_k)\|g_k\|^2, \tag{33}$$

the numerator of (27) reduces to

$$\lambda_k g_k^T (g_{k-1} - d_{k-1}). \tag{34}$$

When

$$d_{k-1} = \frac{1}{\lambda_k} g_k, \tag{35}$$

then

$$\lambda_k g_k^T (g_{k-1} - d_{k-1}) = \|g_k\|^2 - \lambda_k g_k^T g_{k-1}. \tag{36}$$

Now, the numerator of β_k^{new} (27) reduces to the numerator of β_k^*. From the above analysis, we can see that (27) indeed is an extension of (10). Due to the existence of the parameters λ_k, μ_k, and ω_k, it would be more flexible to call methods (2), (6), and (27) by this paper of conjugate gradient methods. Numerical experiments results in Section 4 demonstrate the influence of these parameters versus formula (27).

Lemma 5. *Suppose that Assumption 1 holds and that x_k is given by (2) and (6), where α_k satisfies the modified Wolfe-Powell conditions (3) and (5), while β_k is computed by (27). Then, one has*

$$\frac{g_k^T d_k}{\|g_k\|^2} < 0 \quad \forall k \geq 1. \tag{37}$$

Proof. When $k = 1$, we have

$$g_1^T d_1 = -\|g_1\|^2, \qquad \frac{g_1^T d_1}{\|g_1\|^2} = -1 < 0. \tag{38}$$

Suppose $g_{k-1}^T d_{k-1} < 0$ hold, $\beta_k = \beta_k^{\text{new}}$ in formula (27), and the conclusion holds.

If $\beta_k = \beta_k^{\text{new}}$, we have

$$g_k^T d_k = -\|g_k\|^2 + \beta_k g_k^T d_{k-1},$$
$$\frac{g_k^T d_k}{\|g_k\|^2} = -1 + \frac{l_k}{(1-\mu_k-\omega_k)\|g_{k-1}\|^2 + \mu_k g_k^T d_{k-1} - (1-\lambda_k+\mu_k+\omega_k) g_{k-1}^T d_{k-1}} \cdot \frac{g_k^T d_{k-1}}{\|g_k\|^2}, \tag{39}$$

where $l_k = \max\{0, \min\{(1-\lambda_k)\|g_k\|^2, \lambda_k g_k^T(g_{k-1}-d_{k-1})\}\}$; by formulas (3) and (5), we have $g_k^T d_{k-1} < 0$. Hence,

$$\frac{g_k^T d_{k-1}}{\|g_k\|^2} < 0,$$
$$\mu_k g_k^T d_{k-1} - (1-\lambda_k+\mu_k+\omega_k) g_{k-1}^T d_{k-1} \geq \mu_k \sigma g_{k-1}^T d_{k-1} - (1-\lambda_k+\mu_k+\omega_k) g_{k-1}^T d_{k-1} = (\mu_k\sigma - 1 + \lambda_k - \mu_k - \omega_k) g_{k-1}^T d_{k-1}. \tag{40}$$

When $\mu_k\sigma - \mu_k < 0, -1+\lambda_k \leq 0$, and $g_{k-1}^T d_{k-1} < 0$, we obtain

$$\mu_k g_k^T d_{k-1} - (1-\lambda_k+\mu_k+\omega_k) g_{k-1}^T d_{k-1} > 0. \tag{41}$$

Due to $(1-\mu_k-\omega_k)\|g_{k-1}\|^2 > 0$ and $l_k \geq 0$, through the above analysis, we have

$$\frac{l_k}{(1-\mu_k-\omega_k)\|g_{k-1}\|^2 + \mu_k g_k^T d_{k-1} - (1-\lambda_k+\mu_k+\omega_k) g_{k-1}^T d_{k-1}} \cdot \frac{g_k^T d_{k-1}}{\|g_k\|^2} < 0. \tag{42}$$

Hence, $g_k^T d_k / \|g_k\|^2 < 0$. □

The result shows that the search direction satisfies descent condition ($g_k^T d_k < 0$); this condition may be crucial for convergence analysis of any conjugate gradient method.

Lemma 6. *Suppose that Assumption 1 holds and that $\{x_k\}$ is given by (2) and (6), where α_k satisfies the modified Wolfe-Powell conditions (3) and (5), while β_k is computed by (27). Then, one has*

$$|\beta_k| \leq \frac{\|g_k\|^2}{\|g_{k-1}\|^2}. \tag{43}$$

Proof. Let $\beta_k = \beta_k^{\text{new}}$; when $(1-\lambda_k)\|g_k\|^2 \leq \lambda_k g_k^T(g_{k-1}-d_{k-1})$, then $l_k = (1-\lambda_k)\|g_k\|^2$; when $(1-\lambda_k)\|g_k\|^2 > \lambda_k g_k^T(g_{k-1} - d_{k-1})$, if $\lambda_k g_k^T(g_{k-1} - d_{k-1}) < 0$, then $l_k = 0$; if $\lambda_k g_k^T(g_{k-1} - d_{k-1}) > 0$, $l_k = \lambda_k g_k^T(g_{k-1} - d_{k-1})$.

By Lemma 5, then $\mu_k g_k^T d_{k-1} - (1-\lambda_k+\mu_k+\omega_k) g_{k-1}^T d_{k-1} > 0$.

To sum up, $|\beta_k| \leq (1-\lambda_k)\|g_k\|^2/(1-\mu_k-\omega_k)\|g_{k-1}\|^2$, and by $\lambda_k > \mu_k + \omega_k$, we have $(1-\lambda_k)/(1-\mu_k-\omega_k) < 1$, and hence $|\beta_k| \leq \|g_k\|^2/\|g_{k-1}\|^2$. □

Theorem 7. *Suppose the objective function $f(x)$ satisfies Assumption 1; consider methods (2) and (6), where β_k is given by (27) and α_k satisfies the modified Wolfe-Powell line search condition. Then, either $g_k = 0$ holds for certain k or*

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{44}$$

Proof. By Theorem 4 and Lemma 6, Theorem 7 is proved. □

The result shows that the proposed algorithm with the modified Wolfe-Powell line search possesses global convergence.

3.2. Algorithm A.

Based on the discussed above, now we can describe the algorithm frame for solving the unconstrained optimization problems (1) as follows.

Step 0. Choose an initial point $x_0 \in R^n$, given constants $\varepsilon_0 > 0$, $\delta \in (0,1)$, $\sigma \in (\delta, 1)$ and $\lambda_k \in (1/2, 1]$, $\mu_k \in [0,1]$, and $\omega_k \in [0,1]$, subject to $\lambda_k \geq \mu_k + \omega_k$, set $d_0 = g_0$, and let $k := 0$.

Step 1. If a stopping criterion $\|g_k\| < \varepsilon_0$ is satisfied, then stop; otherwise, go to Step 2.

Step 2. Determine a step size α_k by line searches (3) and (5).

Step 3. Let $x_{k+1} = x_k + \alpha_k d_k$; compute β_k and d_k by (27) and (6).

Step 4. Set $k := k + 1$ and go to Step 1.

4. Numerical Experiments and Results

In this section, in order to show the performance of the given algorithm, we test our proposed algorithm (algorithm A) and DY algorithm (given by formula (10)) via unconstrained optimization problems from Andrei [29] as follow. These testing functions are often used in engineering field:

(1) Sphere function $f_{\text{Sph}}(x) = \sum_{i=1}^{100} x_i^2$, $x_i \in [-5.12, 5.12]$, $x^* = (0, 0, \ldots, 0)$, $f_{\text{Sph}}(x^*) = 0$.

(2) Rastrigin function $f_{\text{Ras}}(x) = 100 + \sum_{i=1}^{10}(x_i^2 - 10\cos(2\pi x_i))$, $x_i \in [-5.12, 5.12]$, $x^* = (0, 0, \ldots, 0)$, $f_{\text{Ras}}(x^*) = 0$.

(3) Freudenstein and Roth function (Froth) $f_{\text{Froth}}(x) = (-13 + x_1 + ((5 - x_2)x_2 - 2)x_2)^2 + (-29 + x_1 + ((x_2 + 1)x_2 - 14)x_2)^2$, $x^* = (5, 4)$, $f_{\text{Froth}}(x^*) = 0$.

(4) Perturbed Quadratic diagonal function (Pqd) $f_{\text{Pqd}}(x) = (\sum_{i=1}^{5} x_i)^2 + \sum_{i=1}^{5}(i/100)x_i^2$, $x^* = (0, 0, \ldots, 0)$, $f_{\text{Pqd}}(x^*) = 0$.

(5) Extended White & Holst function (Ewh) $f_{\text{Ewh}}(x) = \sum_{i=1}^{5}[100(x_{2i} - x_{2i-1}^3)^2 + (1 - x_{2i-1})^2]$, $x^* = (1, 1, \ldots, 1)$, $f_{\text{Ewh}}(x^*) = 0$.

(6) Raydan 1 function $f_{\text{Ray1}}(x) = \sum_{i=1}^{2}(i/10)(\exp(x_i) - x_i)$, $x^* = (0, 0)$, $f_{\text{Ray1}}(x^*) = 0.3$.

(7) Raydan 2 function $f_{\text{Ray2}}(x) = \sum_{i=1}^{500}(\exp(x_i) - x_i)$, $x^* = (0, 0, \ldots, 0)$, $f_{\text{Ray2}}(x^*) = 500$.

(8) Extended Trigonometric function (Etri) $f_{\text{Etri}}(x) = \sum_{i=1}^{10}[(n - \sum_{j=1}^{10}\cos x_j) + i(1 - \cos x_i) - \sin x_i]^2$, $x^* = (0, 0, \ldots, 0)$, $f_{\text{Etri}}(x^*) = 0$.

(9) Extended Powell function (Epow) $f_{\text{Epow}}(x) = (x_3 + 10x_2)^2 + 5(x_3 - x_2)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$, $x^* = (0, 0, \ldots, 0)$, $f_{\text{Epow}}(x^*) = 0$.

(10) Wood function $f_{\text{Wood}}(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + (1 - x_3)^2 + 90(x_4 - x_3^2)^2 + 10(x_2 + x_4 - 2)^2 + 0.1(x_2 - x_4^2)^2$, $x^* = (1, 1, \ldots, 1)$, $f_{\text{Wood}}(x^*) = 0$.

(11) Extended Wood function (Ewood) $f_{\text{Ewood}}(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (1 - x_3)^2 + 90(x_3^2 - x_4)^2 + 10.1\{(x_2 - 1)^2 + (x_4 - 1)^2\} + 19.8(x_2 - 1)(x_4 - 1)$, $x^* = (1, 1, \ldots, 1)$, $f_{\text{Ewood}}(x^*) = 0$.

(12) Perturbed Quadratic function (Perq) $f_{\text{Perq}}(x) = \sum_{i=1}^{3} ix_i^2 + (1/100)(\sum_{i=1}^{n} x_i)^2$, $x^* = (0, 0, \ldots, 0)$, $f_{\text{Perq}}(x^*) = 0$.

(13) Extended Tridiagonal 1 function (Etri1) $f_{\text{Etri1}}(x) = \sum_{i=1}^{500}[(x_{2i-1} + x_{2i} - 3)^2 + (x_{2i-1} - x_{2i} + 1)^4]$, $x^* = (1, 2, 1, 2, \ldots, 1, 2)$, $f_{\text{Etri1}}(x^*) = 0$.

(14) Extended Miele & Cantrell function (Emic) $f_{\text{Emic}}(x) = \sum_{i=1}^{2}[(\exp(x_{4i-3}) - x_{4i-2})^2 + 100(x_{4i-2} - x_{4i-1})^6 + \{\tan(x_{4i-1} - x_{4i})\}^4 + x_{4i-3}^8]$, $x^* = (0, 1, 1, 1, 0, 1, 1, 1, \ldots, 0, 1, 1, 1)$, $f_{\text{Emic}}(x^*) = 0$.

(15) Extended Rosenbrock function (Erosen) $f_{\text{Erosen}}(x) = \sum_{i=1}^{10}[100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2]$, $x^* = (1, 1, \ldots, 1, 1)$, $f_{\text{Erosen}}(x^*) = 0$.

(16) Generalized Rosenbrock function (Grosen) $f_{\text{Grosen}}(x) = \sum_{i=1}^{1999}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$, $x^* = (1, 1, \ldots, 1, 1)$, $f_{\text{Grosen}}(x^*) = 0$.

(17) QUARTC function $f_{\text{QUAR}}(x) = \sum_{i=1}^{20}(x_i - 1)^4$, $x^* = (1, 1, \ldots, 1, 1)$, $f_{\text{QUAR}}(x^*) = 0$.

(18) LIARWHD function $f_{\text{LIAR}}(x) = \sum_{i=1}^{100} 4(x_i^2 - x_1)^2 + \sum_{i=1}^{100}(x_i - 1)^2$, $x^* = (1, 1, \ldots, 1, 1)$, $f_{\text{LIAR}}(x^*) = 0$.

(19) Staircase 1 function $f_{\text{Stai1}}(x) = \sum_{i=1}^{4}(\sum_{j=1}^{i} x_j)^2$, $x^* = (0, 0, \ldots, 0, 0)$, $f_{\text{Stai1}}(x^*) = 0$.

(20) Staircase 2 function $f_{\text{Stai2}}(x) = \sum_{i=1}^{300}[(\sum_{j=1}^{i} x_j) - i]^2$, $x^* = (1, 1, \ldots, 1, 1)$, $f_{\text{Stai2}}(x^*) = 0$.

(21) POWER function $f_{\text{POWER}}(x) = \sum_{i=1}^{1000}(ix_i)^2$, $x^* = (0, 0, \ldots, 0, 0)$, $f_{\text{POWER}}(x^*) = 0$.

(22) Diagonal 4 function $f_{\text{Dia4}}(x) = \sum_{i=1}^{2}(1/2)(x_{2i-1}^2 + 100x_{2i}^2)$, $x^* = (0, 0, \ldots, 0, 0)$, $f_{\text{Dia4}}(x^*) = 0$.

(23) Extended BD1 function (EBD1) $f_{\text{EBD1}}(x) = \sum_{i=1}^{5000}[(\exp(x_{2i-1}) - x_{2i})^2 + (x_{2i-1}^2 + x_{2i}^2 - 2)^2$, $x^* = (1, 1, \ldots, 1, 1)$, $f_{\text{EBD1}}(x^*) = 0$.

(24) CUBE function $f_{\text{CUBE}}(x) = (x_1 - 1)^2 + \sum_{i=2}^{300} 100(x_i - x_{i-1}^3)^2$, $x^* = (1, 1, \ldots, 1, 1)$, $f_{\text{CUBE}}(x^*) = 0$.

Here, x^* and $f(x^*)$ are the optimal solution and the function value at the optimal solution, respectively. For each algorithm, the parameters are chosen as $\delta = 0.04$ and $\sigma = 0.5$. All codes were written in MATLAB 7.5 and run on Lenovo with 1.90 GHz CPU processor, 2.43 GB RAM memory, and Windows XP operating system. The stop criterion of the iteration is one of the following conditions: (1) $\|g_k\| \le \varepsilon_0 = 10^{-4}$ and (2) the number of iterations Itr > 5000. If condition (2) occurs, the method is deemed to fail for solving the corresponding test problem, and denote it by "F." For the first three test problems, we present experimental results to observe the behavior of the proposed and DY (given by formula (10)) conjugate gradient algorithm for different λ_k, different μ_k, and different ω_k. Details of the schemes for parameters set are given in Table 1. Numerical results of test problems are listed in Tables 2, 3, 4, 5, 6, 7, and 8, respectively. Table 9 shows numerical results of other test problems. Here, x_0 denotes the initial point of the test problems and $\overline{x}$ and $f(\overline{x})$ are iteration value and the function value at the final iteration, respectively.

Based on Table 1's sixteen kinds of scheme (different parameters set), we compared algorithm A with DY (given by formula (10)) conjugate gradient algorithm based on different initial point for three test problems. It is easy to see that the two algorithms based on different scheme (different parameters set) are successful for the first and the second test problems listed in Tables 2, 3, and 4. From Tables 5, 6, 7, 8, and 9, we can see that algorithm A is more successful than DY (given by formula (10)) conjugate gradient algorithm. For example, for the first three test problems based on different scheme (different parameters set), algorithm A based on different initial point all achieved satisfied iteration value and the function value at the final

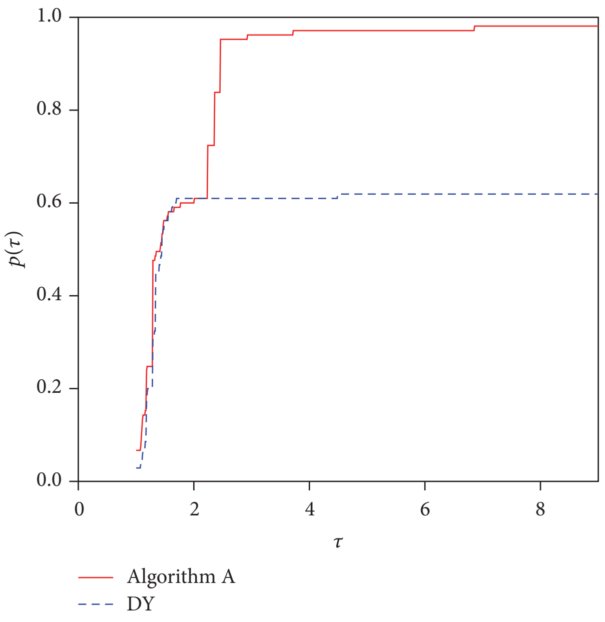

FIGURE 1: Performance profile on the absolute errors of $f(\overline{x})$ versus $f(x^*)$ (algorithm A versus DY).

TABLE 1: Several schemes for the parameters set.

Parameters set	Scheme number
$\lambda_k = 1.0, \mu_k = 0.1, \omega_k = 0.1$	[1]
$\lambda_k = 0.8, \mu_k = 0.1, \omega_k = 0.1$	[2]
$\lambda_k = 0.7, \mu_k = 0.1, \omega_k = 0.1$	[3]
$\lambda_k = 0.6, \mu_k = 0.1, \omega_k = 0.1$	[4]
$\lambda_k = 1.0, \mu_k = 0.1, \omega_k = 0.8$	[5]
$\lambda_k = 1.0, \mu_k = 0.1, \omega_k = 0.6$	[6]
$\lambda_k = 1.0, \mu_k = 0.1, \omega_k = 0.4$	[7]
$\lambda_k = 1.0, \mu_k = 0.1, \omega_k = 0.2$	[8]
$\lambda_k = 0.9, \mu_k = 0.3, \omega_k = 0.5$	[9]
$\lambda_k = 0.9, \mu_k = 0.3, \omega_k = 0.4$	[10]
$\lambda_k = 0.9, \mu_k = 0.3, \omega_k = 0.3$	[11]
$\lambda_k = 0.9, \mu_k = 0.3, \omega_k = 0.1$	[12]
$\lambda_k = 0.7, \mu_k = 0.2, \omega_k = 0.1$	[13]
$\lambda_k = 0.7, \mu_k = 0.3, \omega_k = 0.1$	[14]
$\lambda_k = 0.7, \mu_k = 0.4, \omega_k = 0.1$	[15]
$\lambda_k = 0.7, \mu_k = 0.5, \omega_k = 0.1$	[16]

iteration. Nevertheless, under some scheme (parameters set), DY (given by formula (10)) conjugate gradient algorithm cannot search satisfied iteration solution and the function value at the final iteration. From Tables 5–9, we can also see that DY (given by formula (10)) conjugate gradient algorithm sometimes is failed based on some scheme (parameters set); however, our algorithm is failed only one time. These indicate that the influence of parameters value's changing in formula (27) on the algorithm is not big. We presented the Dolan and Moré [30] performance profiles for the algorithm A and DY method. Note that the performance ratio $p(\tau)$ is the probability for a solver s for the tested problems with the factor τ of the smallest cost. As we can see from Figure 1, algorithm A is superior to DY method for the absolute errors of $f(\overline{x})$ versus $f(x^*)$. Hence, compared with the DY (given by formula (10)) conjugate gradient algorithm, algorithm A has higher stability and adaptability. Therefore, algorithm A yields a better numerical performance than the DY (given by formula (10)) conjugate gradient algorithm. From the above analysis, we can conclude that algorithm A is competitive for solving unconstrained optimization problems.

5. Application to Engineering

In this section, we present a real example to illustrate application of the algorithm proposed in this article. The example is the results of tests on endurance of deep groove ball bearings. For illustrating the purposes, we applied the real dataset of 23 observed failure times that was initially reported in Lieblein and Zelen [31] and later by a number of authors including Abouammoh and Alshingiti [32] and Krishna and Kumar [33]. The following dataset represents the number of millions of revolutions before failure for each of the 23 ball bearings in a life test: 17.88, 28.92, 33.0, 41.52, 42.12, 45.60, 48.40, 51.84, 51.96, 54.12, 55.56, 67.80, 68.64, 68.64, 68.88, 84.12, 93.12, 98.64, 105.12, 105.84, 127.92, 128.04, and 173.4. Dey and Pradhan [34] indicated that Weibull distribution fits this dataset better than the exponential, inverted exponential, and gamma distribution. A random variable X follows the Weibull distribution with probability density function (pdf) being that

$$f(x;\alpha,\lambda) = \alpha\lambda e^{-\lambda x^{\alpha}} x^{\alpha-1}, \quad x > 0, \tag{45}$$

where $\alpha > 0$ and $\lambda > 0$ are the shape and scale parameters, respectively. Let n denote the number of observed failures

Table 2: The numerical results of sphere function for different scheme.

		Algorithm A	DY (given by formula (10)) algorithm
		$\bar{x}/f(\bar{x})$	$\bar{x}/f(\bar{x})$
Sphere	x_0	$(-1,-2,-3,-4,-5,1,2,3,4,5,-1,-2,\dots,1,2,3,4,5)$	$(-1,-2,-3,-4,-5,1,2,3,4,5,-1,-2,\dots,1,2,3,4,5)$
	1	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	2	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	3	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	4	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	5	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	6	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
Scheme	7	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	8	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	9	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	10	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	11	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$
	12	$\bar{x}/2.9098e-11$	$\bar{x}/2.9098e-11$

Note: $\bar{x} = 1.0e-5 * (0.0512, 0.1024, 0.1536, 0.2048, 0.2560, -0.0512, -0.1024, -0.1536, -0.2048, -0.2560, 0.0512, 0.1024, 0.1536, 0.2048, 0.2560, \dots, -0.0512, -0.1024, -0.1536, -0.2048, -0.2560)$.

Table 3: The numerical results of sphere function for different scheme.

		Algorithm A	DY (given by formula (10)) algorithm
		$\bar{x}/f(\bar{x})$	$\bar{x}/f(\bar{x})$
Sphere	x_0	$(3,3,\dots,3)$	$(3,3,\dots,3)$
	1	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	2	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	3	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	4	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	5	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	6	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
Scheme	7	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	8	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	9	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	10	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	11	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$
	12	$1.0e-5*(-0.1536,-0.1536,\dots,-0.1536)/2.5952e-11$	$1.0e-5*(0.2556,0.2556,\dots,0.2556)/7.1845e-11$

Table 4: The numerical results of Rastrigin function for different scheme.

		Algorithm A	DY (given by formula (10)) algorithm
		$\bar{x}/f(\bar{x})$	$\bar{x}/f(\bar{x})$
Rastrigin	x_0	$(0.5,0.5,\dots,0.5)$	$(0.5,0.5,\dots,0.5)$
	1	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$
	2	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$
	3	$1.0e-7*(0.3504,0.3504,\dots,0.3504)/7.3008e-13$	$1.0e-7*(0.3504,0.3504,\dots,0.3504)/7.3008e-13$
	4	$-1.0e-7*(0.6856,0.6856,\dots,0.6856)/2.7978e-12$	$-1.0e-7*(0.6856,0.6856,\dots,0.6856)/2.7978e-12$
	5	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$
	6	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$
Scheme	7	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$
	8	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$	$1.0e-7*(0.6520,0.6520,\dots,0.6520)/2.5260e-12$
	13	$-1.0e-7*(0.7804,0.7804,\dots,0.7804)/3.6238e-12$	$-1.0e-7*(0.7804,0.7804,\dots,0.7804)/3.6238e-12$
	14	$1.0e-7*(0.6697,0.6697,\dots,0.6697)/2.6699e-12$	$1.0e-7*(0.6697,0.6697,\dots,0.6697)/2.6699e-12$
	15	$-1.0e-7*(0.7222,0.7222,\dots,0.7222)/3.1015e-12$	$-1.0e-7*(0.7222,0.7222,\dots,0.7222)/3.1015e-12$
	16	$1.0e-7*(0.5194,0.5194,\dots,0.5194)/1.6094e-12$	$1.0e-7*(0.5194,0.5194,\dots,0.5194)/1.6094e-12$

Table 5: The numerical results of Rastrigin function for different scheme.

		Algorithm A	DY (given by formula (10)) algorithm
		$\overline{x}/f(\overline{x})$	$\overline{x}/f(\overline{x})$
Rastrigin	x_0	(0.5, 0.5, 0.5, 0.5, 0.5, 0.1, 0.1, 0.1, 0.1, 0.1)	(0.5, 0.5, 0.5, 0.5, 0.5, 0.1, 0.1, 0.1, 0.1, 0.1)
	1	$1.0e-7*(0.0658,\ldots,0.0658,0.1700,\ldots,0.1700)/2.6645e-14$	$1.0e-7*(0.0658,\ldots,0.0658,0.1700,\ldots,0.1700)/2.6645e-14$
	2	$1.0e-7*(0.2044,\ldots,0.2044,0.5798,\ldots,0.5798)/2.4514e-13$	$1.0e-7*(0.2044,\ldots,0.2044,0.5798,\ldots,0.5798)/2.4514e-13$
	3	$1.0e-7*(0.2786,\ldots,0.2786,0.8842,\ldots,0.8842)/4.5830e-13$	$1.0e-7*(0.2786,\ldots,0.2786,0.8842,\ldots,0.8842)/4.5830e-13$
	4	$1.0e-6*(0.0288,\ldots,0.0288,0.1034,\ldots,0.1034)/4.9027e-13$	$1.0e-6*(0.0288,\ldots,0.0288,0.1034,\ldots,0.1034)/4.9027e-13$
	5	$1.0e-7*(0.0658,\ldots,0.0658,0.1700,\ldots,0.1700)/2.6645e-14$	$(-0.9950,\ldots,-0.9950,13.9269,\ldots,13.9269)/2.9849$
	6	$1.0e-7*(0.0658,\ldots,0.0658,0.1700,\ldots,0.1700)/2.6645e-14$	$(0.0000,\ldots,0.0000,-8.9540,\ldots,-8.9540)/4.6096e-12$
Scheme	7	$1.0e-7*(0.0658,\ldots,0.0658,0.1700,\ldots,0.1700)/2.6645e-14$	$(0.0000,\ldots,0.0000,19.8909,\ldots,19.8909)/1.6094e-12$
	8	$1.0e-7*(0.0658,\ldots,0.0658,0.1700,\ldots,0.1700)/2.6645e-14$	$(0.9550,\ldots,0.9550,-20.8843,\ldots,-20.8843)/2.9849$
	13	$1.0e-7*(0.2687,\ldots,0.2687,0.8076,\ldots,0.8076)/4.2633e-13$	F/F
	14	$1.0e-7*(0.2558,\ldots,0.2558,0.7716,\ldots,0.7716)/3.8369e-13$	F/F
	15	$1.0e-7*(0.2430,\ldots,0.2430,0.7357,\ldots,0.7357)/3.5172e-13$	F/F
	16	$1.0e-7*(0.2303,\ldots,0.2303,0.7000,\ldots,0.7000)/3.1442e-13$	F/F

Table 6: The numerical results of Freudenstein and Roth function for different scheme.

		Algorithm A	DY (given by formula (10)) algorithm
		$\overline{x}/f(\overline{x})$	$\overline{x}/f(\overline{x})$
Froth	x_0	(0.5, −2)	(0.5, −2)
	1	$(4.9992, 4)/8.6900e-7$	$(5.0000, 4.0000)/4.0440e-10$
	2	$(4.9992, 4)/8.6900e-7$	F/F
	3	$(4.9992, 4)/8.6900e-7$	$(5.0000, 4.0000)/4.0440e-10$
	4	$(4.9992, 4)/8.6900e-7$	$(11.4128, -0.8968)/48.9843$
	5	$(4.9992, 4)/8.6900e-7$	$(5.0000, 4.0000)/4.0440e-10$
	6	$(4.9992, 4)/8.6900e-7$	$(5.0000, 4.0000)/4.0440e-10$
Scheme	7	$(4.9992, 4)/8.6900e-7$	F/F
	8	$(4.9992, 4)/8.6900e-7$	$(11.4128, -0.8968)/48.9843$
	9	$(4.9992, 4)/8.6900e-7$	$(11.4128, -0.8968)/48.9843$
	10	$(4.9992, 4)/8.6900e-7$	$(11.4128, -0.8968)/48.9843$
	11	$(4.9992, 4)/8.6900e-7$	$(11.4128, -0.8968)/48.9843$
	12	$(4.9992, 4)/8.6900e-7$	F/F

and $t_1,\ldots,t_n$ denote the complete sample; the logarithm likelihood function is

$$\log L(\alpha,\lambda;\text{data}) = n\log(\alpha\lambda) - \lambda\sum_{i=1}^{n} t_i^{\alpha} + (\alpha-1)\sum_{i=1}^{n}\log(t_i). \tag{46}$$

The corresponding differential equations are

$$\frac{\partial \log L}{\partial \alpha} = \frac{n}{\alpha} - \alpha\lambda\sum_{i=1}^{n} t_i^{\alpha-1} + \sum_{i=1}^{n}\log(t_i) = 0,$$
$$\frac{\partial \log L}{\partial \lambda} = \frac{n}{\alpha} - \sum_{i=1}^{n} t_i^{\alpha} = 0. \tag{47}$$

Table 7: The numerical results of Freudenstein and Roth function for different scheme.

		Algorithm A	DY (given by formula (10)) algorithm
		$\overline{x}/f(\overline{x})$	$\overline{x}/f(\overline{x})$
	x_0	$(-0.5, 2)$	$(-0.5, 2)$
Froth	1	$(4.9986, 4)/3.0289e-6$	F/F
	2	$(4.9986, 4)/3.0289e-6$	F/F
	3	$(4.9986, 4)/3.0289e-6$	$(11.4128, -0.8968)/48.9843$
	4	$(4.9986, 4)/3.0289e-6$	$(4.9999, -4.0000)/1.0003e-8$
	5	$(4.9986, 4)/3.0289e-6$	F/F
	6	$(4.9986, 4)/3.0289e-6$	F/F
Scheme	7	$(4.9986, 4)/3.0289e-6$	F/F
	8	$(4.9986, 4)/3.0289e-6$	F/F
	9	$(4.9986, 4)/3.0289e-6$	F/F
	10	$(4.9986, 4)/3.0289e-6$	F/F
	11	$(4.9986, 4)/3.0289e-6$	F/F
	12	$(4.9986, 4)/3.0289e-6$	F/F

Table 8: The numerical results of Freudenstein and Roth function for different scheme.

		Algorithm A	DY (given by formula (10)) algorithm
		$\overline{x}/f(\overline{x})$	$\overline{x}/f(\overline{x})$
	x_0	$(-0.5, -2)$	$(-0.5, -2)$
Froth	1	$(4.9989, 4.0000)/1.7488e-6$	$(5.0000, 4.0000)/1.7660e-10$
	2	$(4.9989, 4.0000)/1.7488e-6$	$(11.4128, -0.8968)/48.9843$
	3	$(4.9989, 4.0000)/1.7488e-6$	F/F
	4	$(4.9989, 4.0000)/1.7488e-6$	F/F
	5	$(4.9989, 4.0000)/1.7488e-6$	F/F
	6	$(4.9989, 4.0000)/1.7488e-6$	$(5.0000, 4.0000)/1.7404e-10$
Scheme	7	$(4.9989, 4.0000)/1.7488e-6$	$(11.4128, -0.8968)/48.9843$
	8	$(4.9989, 4.0000)/1.7488e-6$	F/F
	13	$(4.9989, 4.0000)/1.7488e-6$	F/F
	14	$(4.9989, 4.0000)/1.7488e-6$	$(5.0000, 4.0000)/2.6700e-10$
	15	$(4.9989, 4.0000)/1.7488e-6$	$(11.4128, -0.8968)/48.9843$
	16	$(4.9989, 4.0000)/1.7488e-6$	$(11.4128, -0.8968)/48.9843$

From (47), a closed-form solution of α and λ does not exist, so a numerical technique (minimization $-\log L$) must be used to find the maximum likelihood estimation (MLE) of α and λ for any given dataset. By using algorithm A, We obtain $\widehat{\alpha} = 3.183499$ and $\widehat{\lambda} = 7.0363e-7$. Dey and Pradhan [34] obtained the MLE of the parameters as follows: $(\widehat{\alpha}, \widehat{\lambda}) = (3.1835, 1.4329e-6)$. From the numerical results, we can see that our algorithm is alternative for the above real unconstrained optimization problem.

6. Conclusion

In this article, by modifying the scalar β_k, we have proposed a three-parameter family of conjugate gradient method for solving large-scale unconstrained optimization problems. Global convergence of the proposed methods under modified Wolfe-Powell line search and general criterion are established, respectively. Numerical results show that our algorithm is competitive for solving unconstrained optimization problems. So, the proposed method is an alternative method used in the reliability data.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

The research work are supported by National Natural Science Foundation of China (11361036; 11461051), the Joint Specialized Research Fund for the Doctoral Program of Higher Education, Inner Mongolia Education Department (20131514110005), Higher School Science Research

Table 9: The numerical results of different function for $\lambda_k = 0.9$, $\mu_k = 0.3$, and $\omega_k = 0.1$.

Functions	x_0	Algorithm A	DY (given by formula (10)) algorithm
		$\overline{x}/f(\overline{x})$	$\overline{x}/f(\overline{x})$
Pqd	$(0.5, -0.5, 0.5, 0.8, 0.9)$	$(0.0081, 0.0081, \ldots, -0.0182)/2.2226e-4$	$(-0.0345, -0.0345, -0.0346, -0.0347, -0.0612)/0.0401$
Ewh	$(1, 2, 1, 2, \ldots, 1, 2, 1, 2)$	$(1.0028, 1.0083, \ldots, 1.0028, 1.0083)/2.3050e-5$	$(1, 1, \ldots, 1, 1)/5.1174e-11$
Raydan1	$(1, 1)$	$1.0e-3 * (0.9370, 0.0005)/0.3000$	$1.0e-3 * (0.9917, 0.0008)/0.3000$
Raydan2	$(1, 1, \ldots, 1, 1)$	$(0, 0, \ldots, 0, 0)/500$	$(0, 0, \ldots, 0, 0)/500$
Etri	$(1, 1, \ldots, 1, 1)$	a	b
Epow	$(3, -1, 0, 1)$	$(2.0000, 0, 0, 2.0000)/5.8875e-11$	F/F
Wood	$(-3, -1, -3, -1)$	$(1.0002, 1.0004, 0.9998, 0.9996)/1.6386e-7$	F/F
Ewood	$(-3, 1.2, -3, 1.2)$	$(1.0001, 1.0002, 0.9999, 0.9998)/2.0820e-8$	F/F
Perq	$(1, 2, 3)$	$1.0e-4 * (0.0973, -0.0020, -0.1014)/4.0320e-10$	$1.0e-4 * (0.0480, 0.0011, 0.1524)/7.2434e-10$
Etri1	$(2, 2, \ldots, 2)$	F/F	$(1.2, 1.2, \ldots, 1.2)/6.5680e-11$
Emic	$(2, 2, \ldots, 2)$	c	F/F
Erosen	$(-1.2, 1, \ldots, -1.2, 1)$	$(0.9999, 0.9999, \ldots, 0.9999)/6.3141e-9$	$(1, 1, \ldots, 1)/3.2274e-12$
Grosen	$(2, 2, \ldots, 2)$	$(1, 1, \ldots, 1)/5.8136e-10$	$(1, 1, \ldots, 1)/5.8136e-10$
QUARTC	$(2, 2, \ldots, 2)$	$(0.6338, 0.6338, \ldots, 0.6338)/0.0899$	$(0.5278, 0.5278, \ldots, 0.5278)/0.2487$
LIARWHD	$(4, 4, \ldots, 4)$	$(1, 1, \ldots, 1)/4.2649e-10$	$(1, 1, \ldots, 1)/7.5304e-10$
Staircase1	$(2, 2, \ldots, 2)$	$1.0e-4 * (0.0448, -0.1380, -0.0297, -0.1576)/2.7012e-10$	$1.0e-4 * (-0.2478, 0.7435, -0.8469, 0.5991)/4.9189e-9$
Staircase2	$(0, 0, \ldots, 0)$	$(1, 1, \ldots, 1)/5.1016e-10$	$(1, 0.9999, \ldots, 1, 0.9999)/4.1476e-9$
POWER	$(1, 1, \ldots, 1, 1)$	$-1.0e-4 * (0.1081, 0, 0, \ldots, 0, 0.0214)/1.9507e-9$	$-1.0e-4 * (0.1081, 0, 0, \ldots, 0, 0.0214)/1.9507e-9$
Diagonal4	$(2, 2, \ldots, 2)$	$1.0e-4 * (-0.2394, 0.0062, -0.2394, 0.0062)/6.1191e-10$	$1.0e-4 * (0.4352, 0.0055, 0.4352, 0.0055)/1.9247e-9$
EBD1	$(0, 1, \ldots, 0, 1)$	$(1, 1, \ldots, 1)/3.1321e-10$	$(1, 1, \ldots, 1)/3.1321e-10$
CUBE	$(-1.2, 1, \ldots, -1.2, 1)$	$(1, 1.0001, \ldots, 1, 1.0001)/1.4082e-9$	$(0.9693, 0.9108, \ldots, 0.9693, 0.9108)/9.3983e-4$

$a = 1.0e3 * (0.0009, 0.0258, -0.0314, 0.0067, 0.0817, 1.0242, 0.1885, -0.5213, -0.9109, -0.4524)/8.0025e-11$.
$b = 1.0e4 * (0.0522, -0.1413, 0.0641, 0.0603, -0.2218, -0.5115, 0.2249, 0.7358, -1.3182, 0.3192)/4.0124e-10$.
$c = (0.2140, 1.2386, 1.2810, 1.3032, -0.5016, 0.6837, 0.8063, 0.7930)/0.0105$.

Project of Inner Mongolia (NJZY16394), Natural Science Foundation of Inner Mongolia (2014MS0112), and Science Research Project of Inner Mongolia University of Technology (ZD201409).

References

[1] B. Balaram, M. D. Narayanan, and P. K. Rajendrakumar, "Optimal design of multi-parametric nonlinear systems using a parametric continuation based genetic algorithm approach," *Nonlinear Dynamics*, vol. 67, no. 4, pp. 2759–2777, 2012.

[2] L. Zhang, H. Gao, Z. Chen, Q. Sun, and X. Zhang, "Multi-objective global optimal parafoil homing trajectory optimization via Gauss pseudospectral method," *Nonlinear Dynamics*, vol. 72, no. 1-2, pp. 1–8, 2013.

[3] X.-z. Jiang and J.-b. Jian, "A sufficient descent Dai-Yuan type nonlinear conjugate gradient method for unconstrained optimization problems," *Nonlinear Dynamics*, vol. 72, no. 1-2, pp. 101–112, 2013.

[4] L. Zhu, G. Wang, and H. Chen, "Estimating steady multi-variables inverse heat conduction problem by using conjugate gradient method," *Proceedings of the Chinese Society of Electrical Engineering*, vol. 31, no. 8, pp. 58–61, 2011.

[5] Z. Wei, G. Li, and L. Qi, "New nonlinear conjugate gradient formulas for large-scale unconstrained optimization problems," *Applied Mathematics and Computation*, vol. 179, no. 2, pp. 407–430, 2006.

[6] Z. X. Wei, G. Y. Li, and L. Q. Qi, "Global convergence of the Polak-Ribière-Polyak conjugate gradient method with an Armijo-type inexact line search for nonconvex unconstrained optimization problems," *Mathematics of Computation*, vol. 77, no. 264, pp. 2173–2193, 2008.

[7] G. Yuan, Z. Wei, and Q. Zhao, "A modified Polak-Ribière-Polyak conjugate gradient algorithm for large-scale optimization problems," *IIE Transactions*, vol. 46, no. 4, pp. 397–413, 2014.

[8] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, 1964.

[9] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.

[10] Y. H. Dai and Y. Yuan, "A nonlinear conjugate gradient method with a strong global convergence property," *SIAM Journal on Optimization*, vol. 10, no. 1, pp. 177–182, 1999.

[11] M. R. Hestenes and E. Stiefel, "Method of conjugate gradient for solving linear equations," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, 1952.

[12] R. Fletcher, *Practical Methods of Optimization. Vol. I*, A Wiley-Interscience Publication, John Wiley & Sons, Chichester, UK, 2nd edition, 1987.

[13] Y. Liu and C. Storey, "Efficient generalized conjugate gradient algorithms, part 1: theory," *Journal of Optimization Theory and Applications*, vol. 69, no. 1, pp. 129–137, 1991.

[14] H. Schramm and J. Zowe, "A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results," *SIAM Journal on Optimization*, vol. 2, no. 1, pp. 121–152, 1992.

[15] L. Lukšan and J. Vlček, "A bundle-Newton method for nonsmooth unconstrained minimization," *Mathematical Programming*, vol. 83, no. 3, pp. 373–391, 1998.

[16] G. Yuan, Z. Wei, and G. Li, "A modified Polak-Ribière-Polyak conjugate gradient algorithm for nonsmooth convex programs," *Journal of Computational and Applied Mathematics*, vol. 255, pp. 86–96, 2014.

[17] Z.-F. Dai, "Two modified HS type conjugate gradient methods for unconstrained optimization problems," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 74, no. 3, pp. 927–936, 2011.

[18] Z. Dai and F. Wen, "Another improved Wei-Yao-Liu nonlinear conjugate gradient method with sufficient descent property," *Applied Mathematics and Computation*, vol. 218, no. 14, pp. 7421–7430, 2012.

[19] X. Z. Jiang, L. Han, and J. B. Jian, "A globally convergent mixed conjugate gradient method with Wolfe-Powell line search," *Mathematica Numerica Sinica*, vol. 34, no. 1, pp. 103–112, 2012.

[20] K. Sugiki, Y. Narushima, and H. Yabe, "Globally convergent three-term conjugate gradient methods that use secant conditions and generate descent search directions for unconstrained optimization," *Journal of Optimization Theory and Applications*, vol. 153, no. 3, pp. 733–757, 2012.

[21] M. Li and H. Feng, "A sufficient descent LS conjugate gradient method for unconstrained optimization problems," *Applied Mathematics and Computation*, vol. 218, no. 5, pp. 1577–1586, 2011.

[22] Z. Dai and F. Wen, "A modified CG-DESCENT method for unconstrained optimization," *Journal of Computational and Applied Mathematics*, vol. 235, no. 11, pp. 3332–3341, 2011.

[23] M. Al-Baali, "Descent property and global convergence of the Fletcher-Reeves method with inexact line search," *IMA Journal of Numerical Analysis*, vol. 5, no. 1, pp. 121–124, 1985.

[24] Y. H. Dai and Y. Yuan, "A three-parameter family of nonlinear conjugate gradient methods," *Mathematics of Computation*, vol. 70, no. 235, pp. 1155–1167, 2001.

[25] L. Nazareth, "Conjugate-gradient methods," in *Encyclopedia of Optimization*, C. Floudas and P. Pardalos, Eds., Kluwer Academic Publishers, Boston, Mass, USA, 1999.

[26] Y. H. Dai and Y. Yuan, "A class of globally convergent conjugate gradient methods," Research Report ICM-98-030, Institute of Computational Mathematics and Scientific/Engineering Computing, Chinese Academy of Sciences, 1998.

[27] G. Zoutendijk, "Nonlinear programming, computational methods," in *Integer and Nonlinear Programming*, pp. 37–86, North-Holland, Amsterdam, The Netherlands, 1970.

[28] Y. H. Dai and Y. X. Yuan, *Nonlinear Conjugate Gradient Method*, Science Press of Shanghai, Shanghai, China, 2000.

[29] N. Andrei, "An unconstrained optimization test functions collection," *Advanced Modeling and Optimization*, vol. 10, no. 1, pp. 147–161, 2008.

[30] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, no. 2, pp. 201–213, 2002.

[31] J. Lieblein and M. Zelen, "Statistical investigation of the fatigue life of deep-groove ball bearings," *Journal of Research of the National Bureau of Standards*, vol. 57, no. 5, pp. 273–316, 1956.

[32] A. M. Abouammoh and A. M. Alshingiti, "Reliability estimation of generalized inverted exponential distribution," *Journal of Statistical Computation and Simulation*, vol. 79, no. 11-12, pp. 1301–1315, 2009.

[33] H. Krishna and K. Kumar, "Reliability estimation in generalized inverted exponential distribution with progressively type-II censored sample," *Journal of Statistical Computation and Simulation*, vol. 83, no. 6, pp. 1007–1019, 2013.

[34] S. Dey and B. Pradhan, "Generalized inverted exponential distribution under hybrid censoring," *Statistical Methodology*, vol. 18, pp. 101–114, 2014.