A Barzilai and Borwein scaling conjugate gradient method for unconstrained optimization problems

Liumei Wang, Wenyu Sun, Raimundo J.B. de Sampaio, Jinyun Yuan

1. Introduction

In this paper we consider the unconstrained optimization problem

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

where $f: \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and bounded below. Conjugate gradient (CG) methods are very important for solving (1.1), especially for large scale optimization problems, this is because, first, it does not need matrix operations and matrix storage; second, its programming is simple; third, it has strong convergence property (see [18,24]). The nonlinear conjugate gradient method generates a sequence $\{x_k\}$ using the iterative scheme

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

where the $k$th steplength factor $\alpha_k$ is obtained by carrying out some line search, and $d_k \in \mathbb{R}^n$ is the $k$th search direction generated by

$$d_k = \begin{cases} -g_k, & \text{if } k = 0; \\ -g_k + \beta_{k-1} d_{k-1}, & \text{if } k > 0. \end{cases}$$

This work was supported by National Natural Science Foundation of China (grant No. 11171159, 11331012), the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) of China, and CAPES and CNPq of Brazil.

E-mail addresses: wangliumei111@sina.com (L. Wang), wysun@njnu.edu.cn (W. Sun), raimundo.sampaio@pucpr.br (R.J.B. de Sampaio), jin@ufpr.br (J. Yuan).

http://dx.doi.org/10.1016/j.amc.2015.04.046

0096-3003/© 2015 Elsevier Inc. All rights reserved.
where $\beta_{k-1}$ is a key CG parameter. The typical choices of $\beta_{k-1}$ are as follows:

$$\beta_{FR}^{k-1} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}},$$  \hspace{1cm} (1.4)$$

$$\beta_{HS}^{k-1} = \frac{g_k^T (g_k - g_{k-1})}{d_{k-1}^T (g_k - g_{k-1})},$$  \hspace{1cm} (1.5)$$

$$\beta_{PRP}^{k-1} = \frac{g_k^T (g_k - g_{k-1})}{g_{k-1}^T (g_k - g_{k-1})},$$  \hspace{1cm} (1.6)$$

$$\beta_{DIXON}^{k-1} = \frac{g_k^T g_k}{d_{k-1}^T (g_k - g_{k-1})},$$  \hspace{1cm} (1.7)$$

$$\beta_{DY}^{k-1} = \frac{g_k^T g_k}{d_{k-1}^T (g_k - g_{k-1})},$$  \hspace{1cm} (1.8)$$

where $\beta_{FR}^{k-1}$, $\beta_{HS}^{k-1}$, $\beta_{PRP}^{k-1}$, $\beta_{DIXON}^{k-1}$, and $\beta_{DY}^{k-1}$ refer respectively to Fletcher–Reeves formula [9], Hestenes–Stiefel formula [16], Polak–Ribiere–Polyak formula [19], Dixon formula [7], and Dai–Yuan formula [5]. The global convergence of the conjugate gradient methods is established and discussed in [1,5,10,11,14,18,20,24,25]. In 2005, Hager and Zhang [13] proposed a new nonlinear conjugate gradient method where $d_k$ is defined as

$$d_{k+1} = -g_{k+1} + \bar{\beta}^N_k d_k, \quad d_0 = -g_0,$$  \hspace{1cm} (1.9)$$

where

$$\bar{\beta}^N_k = \max(\beta^N_k, \eta_k),$$  \hspace{1cm} (1.10)$$

$$\eta_k = -\frac{1}{\|d_k\| \min(\eta, \|g_k\|)},$$  \hspace{1cm} (1.11)$$

$\eta > 0$ is a constant, and

$$\beta^N_k = \frac{1}{d_k^T y_k} \left( y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k} \right)^T g_{k+1},$$

with $y_k = y_{k+1} - y_k$. A good property of the new conjugate gradient method is that the search direction $d_k$ satisfies $g_k^T d_k \leq -\frac{7}{8} \|g_k\|^2$ which is independent of the line search used. In order to constrain the choice of $\alpha_k$ to ensure convergence, Hager and Zhang [13] considered the line search that satisfies the Wolfe conditions:

$$f(x_k + \alpha_k d_k) - f(x_k) \leq \delta \alpha_k g_k^T d_k,$$  \hspace{1cm} (1.12)$$

$$g_{k+1}^T d_k \geq \sigma g_k^T d_k,$$  \hspace{1cm} (1.13)$$

where $0 < \delta \leq \sigma < 1$. They established the global convergence result. Sun et al. [23] gave a Beale’s preconditioning conjugate gradient method with three-term which is efficient in numerical experiments.

The Barzilai and Borwein (BB) gradient method is a two-point step-size gradient method [2,8,24] in which the step-size is derived from a two-point approximation of the secant equation. Its iterative scheme is

$$x_{k+1} = x_k - \frac{1}{\theta_k} g_k = x_k - D_k g_k,$$  \hspace{1cm} (1.14)$$

The choice of $\theta_k$ must meet the conditions that $D_k = \frac{1}{\theta_k} I$ is an approximation of the inverse matrix of the Hessian and has some approximate quasi-Newton property. In order to force the matrix $D_k$ to have certain quasi-Newton properties, it is reasonable to require

$$\min \|D_k s_{k-1} - y_{k-1}\| = \min \left\| \frac{1}{\theta_k} s_{k-1} - y_{k-1} \right\|,$$

where $\| \cdot \|$ refers to the Euclidean norm. With these, Barzilai and Borwein [2] obtained the following choices of $\theta_k$:

$$\theta_k = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}$$
or
\[ \theta_k = \frac{y_{k-1}^T y_{k-1}}{s_{k-1}^T y_{k-1}}, \]
where \( s_{k-1} = x_k - x_{k-1}, y_k = g_k - g_{k-1} \). Raydan [21] proved that, for strictly convex quadratic function, the BB method is globally convergent. Further, Raydan [22] proved the global convergence of the BB method based on the nonmonotone line search technique for solving non-convex quadratic functions. Dai and Liao [3] established R-linear convergence of BB method. The numerical results reported in [4,6,12,22] show that some modified versions of BB algorithm are competitive to several well-known conjugate gradient algorithms for large-scale unconstrained optimization. Later, Zhang et al. [26] presented a nonmonotone filter BB method for optimization. They proved the global convergence and reported the efficient numerical results. He et al. [15] proposed a BB-projection method for solving variational inequalities and obtained efficient numerical results.

The characteristic of the Barzilai and Borwein (BB) gradient method is that it does not need to compute the Hessian matrix but possesses some approximate quasi-Newton property. A lot of numerical experiments show that it only requires less computational efforts but it is superior to general gradient methods (see [2,8,22,24,26]). It gives us an inspiration: BB method is a gradient-like method but possesses certain approximate quasi-Newton properties. A lot of numerical experiments show that it only requires less computational efforts but it is superior to general gradient methods (see [2,8,22,24,26]). It gives us an inspiration: BB method is a gradient-like method but possesses certain approximate quasi-Newton properties. At the same time it employs the previous two-point information in the iterative procedure. Also, it is well-known that CG method is a kind of efficient methods for large scale optimization. How to inherit the idea and the property of BB method in the conjugate gradient method and give a powerful BB scaling CG method for solving unconstrained optimization problems? This is our motivation.

Therefore, the new method proposed in this paper is interesting. The difference between the proposed method and the BB method lies in that, in new method, the new direction is a combination of \( -\lambda_k g_{k+1} \) and \( \bar{\beta}^N_k d_k \), but BB method only employs \( -\bar{g}_{k+1} \) to generate new direction. In addition, the difference between the proposed method and the CG method lies in that, for \( -\bar{g}_{k+1} \) part, new method uses parameter \( \lambda_k \) to scale the negative gradient direction, i.e., we put the BB parameter \( \lambda_k \) with two-point step-size information into CG framework. So, we can expect that the new approach will be superior to both BB method and CG method. The initial numerical results verify our expectation.

The organization of the paper is as follows. In Section 2 the Barzilai–Borwein scaling conjugate gradient algorithm is presented. In Section 3 we establish the descent property and the global convergence of our algorithm. In Section 4 we report some numerical results which show our algorithm is competitive. Finally, we give some concluding remarks in Section 5.

2. A Barzilai–Borwein scaling conjugate gradient method

Simply, our Barzilai–Borwein scaling conjugate gradient method is as follows,
\[ d_{k+1} = -\lambda_k g_{k+1} + \bar{\beta}^N_k d_k, \quad d_0 = -g_0, \] (2.1)
where \( \bar{\beta}^N_k \) is defined by (1.10) and \( \lambda_k \) is computed by the Barzilai and Borwein gradient method, i.e.,
\[ \lambda_k = \frac{1}{\theta_k} \quad \text{and} \quad \theta_k = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}. \]

Now we describe our algorithm (short for BBSCG).

**Algorithm 2.1.** (BB scaling conjugate gradient algorithm (BBSCG))

Step 0. Given \( x_0 \in \mathbb{R}^n, \epsilon > 0, 0 < \delta \leq \sigma < 1, \eta > 0, \theta \in (0, 1), \gamma \in (0, 1), \theta_0, 0.5 < \xi < 1, 0 < \tau < 8. \)

Step 1. Compute \( g_0 = g(x_0) \). Set \( d_0 = -g_0 \) and set \( k = 0 \).

Step 2. If \( \|g_k\| \leq \epsilon \), stop.

Step 3. Compute \( \alpha_k \) by line search Wolfe rule and set \( x_{k+1} = x_k + \alpha_k d_k \).

Step 4. Set \( y_k = g_{k+1} - g_k \cdot \bar{\beta}^N_k d_k = \frac{1}{d_k^T y_k} \left( y_k - 2d_k \frac{\|y_k\|^2}{d_k^T y_k} \right) g_{k+1} \).
\[ \eta_k = \frac{1}{\|g_k\|} \min\{\eta_k, \|g_k\|\}, \quad \bar{\beta}^N_k = \max\{\beta^N_k, \eta_k\}. \]

Step 5. S1: If \( \theta_k \leq \xi \) or \( \theta_k \geq \frac{1}{\xi} \), then set \( \theta_k = \tau \).

S2: Set \( \lambda_k = \frac{1}{\theta_k} \).

S3: Set \( d_{k+1} = -\lambda_k g_{k+1} + \bar{\beta}^N_k d_k, \theta_{k+1} = \frac{s_k^T y_k}{s_k^T s_k} \).

Step 6. Set \( k = k + 1 \) and go to Step 2.

**Remarks.**

(1) The aim of S1 in Step 5 is to keep the sequence \( (\lambda_k) \) uniformly bounded. In fact, for all \( k, 0 < \min(\xi, 1) \leq \lambda_k \leq \max(\frac{1}{\xi}, 1) \).

(2) When the line search in the algorithm satisfies the Wolfe conditions, the condition \( \frac{1}{\xi} y_k \neq 0 \) holds.
3. Convergence analysis

Our convergence analysis is based on Gilbert and Nocedal [10] for the PRP+ scheme and Hager and Zhang [13] for the scheme (1.9). Following the framework in [10,13], we give Lemma 3.1–3.3 which differ in the treatment of our scheme (2.1). In the following, we first give assumptions:

A0. $f: \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and bounded below.
A1. The level set $L = \{ x \in \mathbb{R}^n \mid f(x) \leq f(x_0) \}$ is bounded.
A2. $\nabla f$ satisfies the Lipschitz condition

$$
\| \nabla f(x) - \nabla f(x_k) \| \leq L \| x - x_k \|
$$

for all $x$ on the line segment connecting $x_k$ and $x_{k+1}$, where $L$ is the Lipschitz constant.

In the following lemma, we show the descent property of $d_k$.

**Lemma 3.1.** If $d^Ty_k \neq 0$ and $d_k$ is generated by (2.1):

$$
d_{k+1} = -\lambda_k g_{k+1} + \hat{g}^N_k d_k, \quad d_0 = -g_0,
$$

then, when $\frac{1}{2} \leq \xi < 1$ and $0 < \tau < 1$, we have

$$
g^T_k d_{k+1} < 0.
$$

**Proof.** (i) Since $d_0 = -g_0$, we have $g^T_0 d_0 = -\|g_0\|^2 < 0$ which satisfies (3.1). (ii) Multiplying (2.1) by $g^T_{k+1}$, we have

$$
g^T_{k+1} d_{k+1} = -\lambda_k \| g_{k+1} \|^2 + \hat{g}^N_k g^T_{k+1} d_k
$$

$$
= -(\lambda_k - 1) \| g_{k+1} \|^2 - \| g_{k+1} \|^2 + \hat{g}^N_k g^T_{k+1} d_k.
$$

It follows from Theorem 1.1 of [13] that

$$
-\| g_{k+1} \|^2 + \hat{g}^N_k g^T_{k+1} d_k \leq -\frac{7}{8} \| g_{k+1} \|^2.
$$

so we have

$$
g^T_{k+1} d_{k+1} \leq - (\lambda_k - 1) \| g_{k+1} \|^2 - \frac{7}{8} \| g_{k+1} \|^2
$$

$$
= - (\lambda_k - \frac{1}{8}) \| g_{k+1} \|^2.
$$

(3.2)

From Step 5 of Algorithm 2.1, we know

$$
\lambda_k \geq \min \left( \xi, \frac{1}{\tau} \right).
$$

If $\xi > \frac{1}{\tau}$, then $\lambda_k \geq \frac{1}{2} > \frac{1}{8}$. If $\frac{1}{\tau} > \xi$, then $\lambda_k \geq \xi > \frac{1}{8}$. Thus we always have $\lambda_k > \frac{1}{8}$. Hence, $g^T_{k+1} d_{k+1} < 0$ follows. □

From Lemma 3.1 we can see that all directions generated by (2.1) are descent directions. Now we establish the global convergence of Algorithm 2.1.

**Lemma 3.2.** Suppose that the conditions of Lemma 3.1 are satisfied and the Assumption A2 holds. If the line search satisfies the Wolfe conditions (1.12) and (1.13), then

$$
\alpha_k \geq \frac{1 - \sigma}{L \| d_k \|^2} \| g^T_k d_k \|.
$$

(3.3)

**Proof.** Subtracting $g^T_k d_k$ from both sides of (1.13), we obtain

$$
g^T_{k+1} d_k - g^T_k d_k \geq \sigma g^T_k d_k - g^T_k d_k.
$$

By the Lipschitz condition, we have

$$
(\sigma - 1) g^T_k d_k \leq (g_{k+1} - g_k)^T d_k \leq \alpha_k \cdot L \| d_k \|^2.
$$

Since $d_k$ is a descent direction and $\sigma < 1$, (3.3) follows. □

**Lemma 3.3.** If Assumptions A0–A2 are satisfied, then for the schemes (2.1), (1.10) and (1.11), and a line search satisfying the Wolfe conditions (1.12) and (1.13), we have

$$
d_k \neq 0 \quad \text{for each } k \quad \text{and} \quad \sum_{k=0}^{\infty} \| u_{k+1} - u_k \|^2 < \infty
$$

whenever $\inf \{ \| g_k \| \mid k \geq 0 \} > 0$ and here $u_k$ is defined as $u_k = \frac{d_k}{\| d_k \|^2}$. 
Proof. Define $\gamma = \inf \{ \| g_k \| \mid k \geq 0 \}$.
Since $\gamma > 0$, then from (3.2), we have $d_k \neq 0$ for each $k$.

Note that the level set $L$ is bounded and that $f$ is bounded below, it follows from (1.12) and (3.3) that the Zoutendijk condition (see [24]) holds:

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\| d_k \|^2} < \infty.$$  

Also, from (3.2) we have

$$\gamma^4 \sum_{k=0}^{\infty} \frac{1}{\| d_k \|^2} \leq \sum_{k=0}^{\infty} \frac{\| g_k \|^4}{\| d_k \|^2} \leq \sum_{k=0}^{\infty} \frac{1}{(\lambda_k - \beta_k^+)^2} \| d_k \|^2 = \sum_{k=0}^{\infty} \frac{1}{(\lambda_k - \beta_k^+)^2} \| d_k \|^2 < \infty,$$

(3.4)where for all $k$,

$$0 < \frac{1}{(\lambda_k - \beta_k^+)^2} < \frac{1}{(\lambda - \beta)^2}.$$  

Define

$$\beta_k^+ = \max \{ \tilde{\beta}_k^N, 0 \}, \quad \beta_k^- = \min \{ \tilde{\beta}_k^N, 0 \}, \quad r_k = \frac{-\lambda_{k-1} g_k + \beta_k^- d_{k-1}}{\| d_k \|}, \quad \delta_k = \beta_k^+ \frac{\| d_{k-1} \|}{\| d_k \|}.$$  

By (2.1), (1.10) and (1.11), we have

$$u_k = \frac{d_k}{\| d_k \|} = \frac{-\lambda_{k-1} g_k + (\beta_k^+ + \beta_k^-) d_{k-1}}{\| d_k \|} = r_k + \frac{\beta_k^+ d_{k-1}}{\| d_k \|},$$

$$= r_k + \frac{\beta_k^+ d_{k-1}}{\| d_k \|} \cdot \frac{d_{k-1}}{\| d_{k-1} \|} = r_k + \delta_k \cdot u_{k-1}.$$  

Since $u_k$ is a unit vector,

$$\| r_k \| = \| u_k - \delta_k u_{k-1} \| = \delta_k \| u_k - u_{k-1} \|.$$  

Also since $\delta_k > 0$, we have

$$\| u_k - u_{k-1} \| \leq (1 + \delta_k) \| u_k - u_{k-1} \| = \| u_k - u_{k-1} \| + \| \delta_k u_k - u_{k-1} \| = 2 \| r_k \|.$$  

(3.5)

By the definition of $\beta_k$ and the fact $\tilde{\beta}_k^N \geq \eta_k$, we have the following bound for the numerator of $r_k$:

$$\| -\lambda_{k-1} g_k + \beta_k^- d_{k-1} \| \leq \| \lambda_{k-1} g_k \| - \min \{ \tilde{\beta}_k^N, 0 \} \| d_{k-1} \| \leq \| \lambda_{k-1} g_k \| - \eta_{k-1} \| d_{k-1} \| \leq \| \lambda_{k-1} g_k \| + \| d_{k-1} \| \min \{ \eta, \gamma \} \| d_{k-1} \| \leq \max \left\{ \frac{1}{\lambda}, \frac{1}{\tau}, \frac{1}{\min \{ \eta, \gamma \}} \right\} \Gamma,$$

(3.6)

where $\Gamma = \max_{k \in \mathbb{Z}} \| \nabla f(x) \|$. Let $c = \max \left\{ \frac{1}{\lambda}, \frac{1}{\tau}, \frac{1}{\min \{ \eta, \gamma \}} \right\}$, then we have

$$\| u_k - u_{k-1} \| \leq \frac{2c}{\| d_k \|}.$$  

(3.7)
Table 1
Testing functions.

<table>
<thead>
<tr>
<th>P</th>
<th>Function</th>
<th>Dimensions</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Helical valley function</td>
<td>( n = 3 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>2</td>
<td>Biggs exp6 function</td>
<td>( n = 6 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>3</td>
<td>Gaussian function</td>
<td>( n = 3 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>4</td>
<td>Powell badly scaled function</td>
<td>( n = 2 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>5</td>
<td>Box 3-dimensional function</td>
<td>( n = 3 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>6</td>
<td>Brown and Dennis function</td>
<td>( n = 4 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>7</td>
<td>Gulf research and development function</td>
<td>( n = 3 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>8</td>
<td>Beale function</td>
<td>( n = 2 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>9</td>
<td>Wood function</td>
<td>( n = 4 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>10</td>
<td>Scaled Cube function ((c = 100))</td>
<td>( n = 2 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>11</td>
<td>Scaled Rosenbrock function</td>
<td>( n = 2 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>12</td>
<td>Watson function</td>
<td>( n = 5000 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>13</td>
<td>Penalty function</td>
<td>( n = 5000 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>14</td>
<td>Trigonometric function</td>
<td>( n = 5000 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>15</td>
<td>Extended Rosenbrock function</td>
<td>( n = 5000 )</td>
<td>Moré et al. [17]</td>
</tr>
<tr>
<td>16</td>
<td>Extended Powell function</td>
<td>( n = 5000 )</td>
<td>Moré et al. [17]</td>
</tr>
</tbody>
</table>

\[
\sum_{k=0}^{\infty} \| u_k - u_{k-1} \|^2 \leq \sum_{k=0}^{\infty} 4c^2 \| d_k \|^2 < \infty.
\]

**Theorem 3.4.** If Assumptions A0–A2 hold, then for the scheme (2.1) with (1.10)–(1.11) and a line search satisfying the Wolfe conditions (1.12)-(1.13), either \( g_k = 0 \) for some \( k \), or

\[
\lim \inf_{k \to \infty} \| g_k \| = 0.
\]

**Proof.** This theorem is a result from Lemma 3.1–3.3. Since the proof is similar to Theorem 3.2 in [13], we omit it. \(\square\)

4. Numerical experiments

In this section, we report the numerical experiments of BBSCG algorithm, and compare its performance with that of the CG-DESCENT algorithm [13]. All testing functions are classical and standard, and come from Moré [17]. All codes are written in MATLAB and run on PC with XP system.

The values of the parameters are as follows:

\[
\epsilon = 10^{-6}, \quad \delta = 0.1, \quad \sigma = 0.9, \quad \eta = 0.01, \quad \theta = 0.5,
\]

\[
\gamma = 0.66, \quad \Theta_0 = 0.3, \quad 1/8 < \xi < 1, \quad 0 < \tau < 8.
\]

The stop criteria are given below:

(1) \( \| g_k \| \leq \epsilon \) is satisfied.

(2) The iteration number exceeds 5000.

Table 1 indicates the testing functions, and Table 2, Figs. 1–4 give the comparisons of the two methods. The symbols in Tables 1 and 2 are explained as follows:

- \( n \): the dimension of testing functions.
- \( n_f \): the number of evaluating testing functions.
- \( n_g \): the number of evaluating gradients of testing functions.
- \( \text{iter} \): the number of iterations.
- \( \cdots \cdots \): the algorithm is failed or the times of iteration exceed 5000.

Table 2 gives the comparisons of numerical experiments of BBSCG and CG-DESCENT. We compare BBSCG to CG-DESCENT from number of function evaluation, number of gradient evaluation, number of iterations, and CPU time. From Table 2, we can see that CG-DESCENT algorithm is failed in function No. 9 and BBSCG algorithm cannot solve function No. 12. For functions No. 1, 4, 6, 10, 15, the performance of two algorithms is almost the same. In addition, the BBSCG is superior to CG-DESCENT in 8 functions: No. 2, 3, 5, 8, 9, 13, 14, 16 and BBSCG is inferior in 3 functions: No. 7, 11, 12. So, the limited numerical experiments indicate that the algorithm BBSCG is potentially efficient.
Table 2
Comparisons of the numerical experiments of BBSCG and CG-DESCENT.

<table>
<thead>
<tr>
<th>P</th>
<th>n</th>
<th>BBSCG</th>
<th></th>
<th></th>
<th></th>
<th>CG-DESCENT</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>n_f</td>
<td>n_x</td>
<td>iter</td>
<td>Time (s)</td>
<td>n_f</td>
<td>n_x</td>
<td>iter</td>
<td>Time (s)</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>----------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>----------</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>557</td>
<td>358</td>
<td>40</td>
<td>7.6656250e−2</td>
<td>535</td>
<td>359</td>
<td>38</td>
<td>8.593750e−2</td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>281</td>
<td>321</td>
<td>41</td>
<td>8.9062500e−2</td>
<td>344</td>
<td>432</td>
<td>53</td>
<td>1.250000e−1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>23</td>
<td>24</td>
<td>4</td>
<td>4.6875000e−3</td>
<td>22</td>
<td>24</td>
<td>4</td>
<td>9.375000e−3</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>122</td>
<td>28</td>
<td>3</td>
<td>1.5625000e−2</td>
<td>122</td>
<td>29</td>
<td>3</td>
<td>1.875000e−2</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>42</td>
<td>41</td>
<td>5</td>
<td>1.7187500e−2</td>
<td>44</td>
<td>47</td>
<td>5</td>
<td>2.031250e−2</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>800</td>
<td>806</td>
<td>53</td>
<td>1.3750000e−1</td>
<td>791</td>
<td>699</td>
<td>47</td>
<td>1.312500e−1</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>2</td>
<td>1.5625000e−2</td>
<td>3</td>
<td>8</td>
<td>2</td>
<td>6.250000e−3</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>170</td>
<td>146</td>
<td>15</td>
<td>3.9062500e−2</td>
<td>188</td>
<td>169</td>
<td>16</td>
<td>4.531250e−2</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>1380</td>
<td>831</td>
<td>93</td>
<td>1.6933750e−1</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>273</td>
<td>161</td>
<td>19</td>
<td>3.7500000e−2</td>
<td>287</td>
<td>175</td>
<td>18</td>
<td>5.312500e−2</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>383</td>
<td>260</td>
<td>26</td>
<td>5.7812500e−2</td>
<td>347</td>
<td>232</td>
<td>23</td>
<td>5.312500e−2</td>
</tr>
<tr>
<td>12</td>
<td>5000</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>45205</td>
<td>31156</td>
<td>4003</td>
<td>1.473438e+1</td>
</tr>
<tr>
<td>13</td>
<td>5000</td>
<td>61</td>
<td>61</td>
<td>30</td>
<td>1.248008e−1</td>
<td>101</td>
<td>101</td>
<td>50</td>
<td>1.404009e+1</td>
</tr>
<tr>
<td>14</td>
<td>5000</td>
<td>24</td>
<td>29</td>
<td>4</td>
<td>1.665625e+0</td>
<td>30</td>
<td>36</td>
<td>5</td>
<td>2.043750e+0</td>
</tr>
<tr>
<td>15</td>
<td>5000</td>
<td>377</td>
<td>249</td>
<td>27</td>
<td>5.131250e+0</td>
<td>317</td>
<td>213</td>
<td>24</td>
<td>4.404888e+0</td>
</tr>
<tr>
<td>16</td>
<td>5000</td>
<td>1057</td>
<td>889</td>
<td>76</td>
<td>3.762188e+1</td>
<td>2659</td>
<td>2127</td>
<td>236</td>
<td>9.036875e+1</td>
</tr>
</tbody>
</table>

Fig. 1. Function performance profiles.

Fig. 2. Gradient performance profiles.
5. Conclusion

In this paper, we propose a Barzilai–Borwein scaling conjugate gradient method for nonlinear unconstrained optimization problems. In this method we use BB parameter with two-point information to scale $-g_k + 1$ in the framework of the conjugate gradient method. We prove the global convergence of the new method. Limited numerical experiments and comparisons show that our new algorithm is potentially efficient.

Acknowledgments

The authors are grateful to Dr. Hongchao Zhang at Louisiana State University for his careful reading and helpful comments. The authors are also grateful to two anonymous referees for their kind and valuable comments. Their comments essentially improve our paper. Finally, the authors thank the Editor–in-Chief and the Associate Editor for their kind and constructive comments.

References