Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 6, No. 3, pp. 591 – 613, September 2010

EXTENDED BARZILAI-BORWEIN METHOD FOR UNCONSTRAINED MINIMIZATION PROBLEMS

Yasushi Narushima, Takahiko Wakamatsu and Hiroshi Yabe*

Abstract: In 1988, Barzilai and Borwein presented a new choice of step size for the gradient method for solving unconstrained minimization problems. Their method aimed to accelerate the convergence of the steepest descent method. The Barzilai-Borwein method has a low storage requirement and inexpensive computations. Therefore, many authors have paid attention to the Barzilai-Borwein method and have proposed some variants to solve large-scale unconstrained minimization problems. In this paper, we extend the Barzilai-Borwein-type methods of Friedlander et al. to more general class and establish global and Q-superlinear convergence properties of the proposed method for minimizing a strictly convex quadratic function. Furthermore, we apply our method to general objective functions. Finally, some numerical experiments are given.

Key words: *large-scale unconstrained minimization problem, Barzilai-Borwein method, global convergence, Q-superlinear convergence*

Mathematics Subject Classification: *90C06, 90C30*

1 Introduction

We consider the following large-scale unconstrained minimization problems:

$$\min \quad f(x), \tag{1.1}$$

where $n \in \boldsymbol{N}$ is very large, $f : \boldsymbol{R}^n \to \boldsymbol{R}$ is sufficiently smooth and its gradient $g \equiv \nabla f$ is available. Although the Newton method and quasi-Newton methods are effective for solving unconstrained minimization problems, these methods cannot apply directly to large-scale unconstrained minimization problems. Therefore, numerical methods which are based on the steepest descent direction are paid attention to, because they avoid the storage of matrices. In this paper, we consider the gradient method defined by

$$x_{k+1} = x_k - \frac{1}{\alpha_k} g_k, \tag{1.2}$$

where x_k is the k-th approximation to the optimal solution x_* of (1.1), g_k is the gradient vector of f at x_k and $1/\alpha_k$ is a step size.

*The authors would like to thank the associate editor and the referees for valuable comments. The first and third authors are supported in part by the Grant-in-Aid for Scientific Research (C) 21510164 of Japan Society for the Promotion of Science.

The steepest descent method is the simplest gradient method for unconstrained minimization problems. In the steepest descent method, which can be traced back to Cauchy (1847), the following exact step size

$$\frac{1}{\alpha_k} = \underset{\alpha>0}{\operatorname{argmin}} \ f(x_k - \frac{1}{\alpha} g_k)$$

is used. Unfortunately, it has been widely known that it converges rather slowly in most cases. In order to overcome this defect, Barzilai and Borwein [1] proposed another step size. Specifically, they approximated the Hessian $\nabla^2 f(x_k)$ by $\alpha_k I$ and based on the secant condition, they considered the following minimization problem:

$$\alpha_k = \arg\min_{\alpha\in R} ||\alpha I s_{k-1} - y_{k-1}||$$

where $s_{k-1} = x_k - x_{k-1}$, $y_{k-1} = g_k - g_{k-1}$ and $\|\cdot\|$ denotes the Euclidean norm. This minimum value is defined by

$$\alpha_k = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}. \tag{1.3}$$

The gradient method with (1.3) is called the Barzilai-Borwein method.

Moreover, Dai, Hager, Schittkowski and Zhang [4] presented numerical results by using

$$\alpha_k = \frac{s_{\nu(k)}^T y_{\nu(k)}}{s_{\nu(k)}^T s_{\nu(k)}} \quad \text{with} \quad \nu(k) = M_c \left\lfloor \frac{k-1}{M_c} \right\rfloor, \tag{1.4}$$

where for $r \in \boldsymbol{R}$, $\lfloor r \rfloor$ denotes the largest integer j such that $j \le r$ and M_c is a positive integer. The gradient method with (1.4) is called the cyclic Barzilai-Borwein method. Numerical results in [4] suggested that their method performed better than the Barzilai-Borwein method did. Since the search direction of the Barzilai-Borwein method $(-(1/\alpha_k)g_k)$ is not necessarily a descent direction, Raydan [17] applied the nonmonotone line search by Grippo et al. [10] to the Barzilai-Borwein method, and proved its global convergence property.

Many researchers study the gradient method for minimizing a strictly convex quadratic function, namely,

$$\min \quad f(x) = \frac{1}{2} x^T A x - b^T x, \tag{1.5}$$

where $A \in \boldsymbol{R}^{n\times n}$ is a symmetric positive definite matrix and $b \in \boldsymbol{R}^n$ is a given vector. For an application of the Barzilai-Borwein method to problem (1.5), Raydan [16] established its global convergence and Dai and Liao [5] proved R-linear rate of convergence. Yuan [19] proposed a choice of α_k such that the solution of (1.5) with $n = 2$ can be found within four iterations, and proved that its related method converges linearly for a general case with $n \ge 2$. Friedlander, Martinez, Molina and Raydan [9] proposed a new gradient method with retards, in which α_k is defined by

$$\alpha_k = \frac{g_{\nu(k)}^T A^{\rho(k)+1} g_{\nu(k)}}{g_{\nu(k)}^T A^{\rho(k)} g_{\nu(k)}}, \quad \nu(k) \in \{k, k-1, ..., \max\{0, k-m\}\} \tag{1.6}$$

and $\rho(k) \in \{q_1, ..., q_m\}$, where m is a positive integer, and $q_1, ..., q_m$ (≥ -2) are integers. They established its global convergence for problem (1.5) and proved the Q-superlinear rate of convergence in the special case. Within the framework of the gradient method with retards, some researchers proposed new choices of α_k. Raydan and Svaiter [18] proposed the Cauchy-Barzilai-Borwein method which chooses α_k of the steepest descent method and the Barzilai-Borwein method alternately. Zhou et al. [21] proposed a method which chooses α_k of the steepest descent method and the minimal gradient method alternately. Yuan [20] proposed a method which chooses α_k of the Barzilai-Borwein method and other types of Barzilai-Borwein method alternately.

The Barzilai-Borwein method and its related methods are reviewed by Dai and Yuan [6] and Fletcher [8].

In this paper, we propose a new step size by extending (1.6). This paper is organized as follows. In Section 2, we propose a new step size and present the algorithm of our method for strictly convex quadratic functions. We show the global convergence property of our method following Friedlander et al. [9]. Moreover using the Dennis-Moré condition, we discuss Q-superlinear convergence. In Section 3, we apply a restricted class of the proposed method to general objective functions by using nonmonotone line search. We establish its global and Q-superlinear convergence properties. Finally, some numerical results are given in Section 4.

2 Extended Barzilai-Borwein Method for Quadratic Functions

In this section, we consider an extension of the Barzilai-Borwein method for minimizing strictly convex quadratic function (1.5). It is desirable that $\alpha_k I$ approximates A (or $1/\alpha_k I$ approximates A^{-1}), and hence Friedlander et al. used a Rayleigh quotient of A. Since a convex combination of Rayleigh quotients possesses more curvature information than (1.6) does, it is significant to construct a method based on such a combination. Accordingly, following Friedlander et al. [9], we propose a new step size for (1.2) as follows:

$$\begin{aligned} \alpha_k &= \sum_{i=1}^{\ell} \phi_i \frac{g_{\nu_i(k)}^T A^{\rho_i(k)+1} g_{\nu_i(k)}}{g_{\nu_i(k)}^T A^{\rho_i(k)} g_{\nu_i(k)}}, \\ \phi_i &\geq 0, \quad \sum_{i=1}^{\ell} \phi_i = 1, \quad \nu_i(k) \in \{k, k-1, ..., \max\{0, k-m\}\} \end{aligned} \tag{2.1}$$

and $\rho_i(k) \in \{q_1, ..., q_m\}$, where ℓ and m are positive integers, and $q_1, ..., q_m$ are integers. We call this gradient method the *extended Barzilai-Borwein* (EBB) method.

Now we describe the algorithm of our method as follows.

Algorithm EBB.

Step 0 . Give $x_0 \in \boldsymbol{R}^n$ and $\ell,\ m \in \boldsymbol{N}$, and set $k = 0$. If $g_0 = 0$, then stop. Otherwise go to Step 1.

Step 1 . Compute α_k by (2.1).

Step 2 . Let $x_{k+1} = x_k - \dfrac{1}{\alpha_k} g_k$. If $g_{k+1} = 0$, then stop.

Step 3 . Let $k := k + 1$ and go to Step 1.

Using (1.2) and $g_k = Ax_k - b$, we have

$$s_k = -\frac{1}{\alpha_k} g_k \quad \text{and} \quad y_k = As_k. \tag{2.2}$$

If $\nu_i(k) \neq k$ for all k, expression (2.2) gives

$$\alpha_k = \sum_{i=1}^{\ell} \phi_i \frac{s_{\nu_i(k)}^T A^{\rho_i(k)+1} s_{\nu_i(k)}}{s_{\nu_i(k)}^T A^{\rho_i(k)} s_{\nu_i(k)}} = \sum_{i=1}^{\ell} \phi_i \frac{y_{\nu_i(k)}^T A^{\rho_i(k)-1} y_{\nu_i(k)}}{y_{\nu_i(k)}^T A^{\rho_i(k)-2} y_{\nu_i(k)}}. \tag{2.3}$$

We note that if $\ell = 1$, $\nu_1(k) = k$ and $\rho_1(k) = 0$ for all k, (2.1) becomes $\alpha_k = g_k^T A g_k / g_k^T g_k$, which implies the steepest descent method. On the other hand, if $\ell = 1$, $\nu_1(k) = \max\{0, k-1\}$ and $\rho_1(k) = 0$ for all k, using (2.2) and (2.3) yields $\alpha_k = s_{k-1}^T y_{k-1} / s_{k-1}^T s_{k-1}$, which is the Barzilai-Borwein method (1.3). Moreover, if $\ell = 1$ and $q_j \geq -2$, then by (2.1), we see that $\alpha_k = g_{\nu_1(k)}^T A^{\rho_1(k)+1} g_{\nu_1(k)} / g_{\nu_1(k)}^T A^{\rho_1(k)} g_{\nu_1(k)}$, which is the gradient method with retards (1.6). Therefore, (2.1) is the extension of (1.3) and (1.6).

Let $\{\lambda_1, \lambda_2, ..., \lambda_n\}$ $(\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n)$ be eigenvalues of A and let $\{v_1, v_2, ..., v_n\}$ be orthonormal eigenvectors of A associated with the eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_n\}$. Since α_k is the Rayleigh quotient of A, the following relation holds

$$0 < \lambda_1 \leq \alpha_k \leq \lambda_n \qquad \text{for all } k. \tag{2.4}$$

In the following subsections, we consider convergence properties of Algorithm EBB.

2.1 Global Convergence

In this subsection, we establish global convergence of the extended Barzilai-Borwein method for problem (1.5) following Friedlander et al. [9]. Let $\{x_k\}$ be the sequence generated by Algorithm EBB. Letting $e_k = x_* - x_k$, we get

$$g_k = Ax_k - b = -Ae_k. \tag{2.5}$$

By (2.1) and (2.5), α_k can be written by

$$\alpha_k = \sum_{i=1}^{\ell} \phi_i \frac{e_{\nu_i(k)}^T A^{\rho_i(k)+3} e_{\nu_i(k)}}{e_{\nu_i(k)}^T A^{\rho_i(k)+2} e_{\nu_i(k)}}. \tag{2.6}$$

For the initial error e_0, there exist constants $d_1^0, d_2^0, ..., d_n^0$ such that

$$e_0 = \sum_{j=1}^{n} d_j^0 v_j. \tag{2.7}$$

It follows from (2.5) that

$$e_{k+1} = e_k + \frac{1}{\alpha_k} g_k = \frac{1}{\alpha_k}(\alpha_k I - A) e_k. \tag{2.8}$$

Thus, using (2.7) and (2.8) yields

$$e_{k+1} = \left\{ \prod_{i=0}^{k} \frac{1}{\alpha_i} (\alpha_i I - A) \right\} \left(\sum_{j=1}^{n} d_j^0 v_j \right) = \sum_{j=1}^{n} d_j^0 \left\{ \prod_{i=0}^{k} \frac{1}{\alpha_i} (\alpha_i - \lambda_j) \right\} v_j.$$

Therefore, defining

$$d_j^{k+1} = \prod_{i=0}^{k} \left(\frac{\alpha_i - \lambda_j}{\alpha_i} \right) d_j^0 \qquad \text{for } j = 1, ..., n,$$

we have

$$e_{k+1} = \sum_{j=1}^{n} d_j^{k+1} v_j \qquad \text{for all} \quad k, \tag{2.9}$$

which implies the relation

$$d_j^{k+1} = \left(\frac{\alpha_k - \lambda_j}{\alpha_k} \right) d_j^k \qquad \text{for} \quad j = 1, ..., n. \tag{2.10}$$

Moreover, by (2.4), the following relations hold for any k

$$\left| 1 - \frac{\lambda_i}{\alpha_k} \right| \leq \frac{\lambda_n - \lambda_1}{\lambda_1} \quad (i = 1, ..., n). \tag{2.11}$$

In order to establish global convergence of Algorithm EBB, we give some lemmas. The following lemma corresponds to Lemma 2.1 in Friedlander et al. [9] and the proof is exactly the same as that of Lemma 2.1 in [9], so we omit it.

Lemma 2.1. *The sequence $\{d_1^k\}$ converges to zero Q-linearly with convergence factor $\hat{c}_1 = 1 - (\lambda_1/\lambda_n)$.*

The following lemma corresponds to Lemma 2.2 in Friedlander et al. [9].

Lemma 2.2. *If the sequences $\{d_1^k\}, \{d_2^k\}, ..., \{d_{p-1}^k\}$ converge to zero for a fixed integer p $(2 \leq p \leq n)$, then*

$$\liminf_{k \to \infty} |d_p^k| = 0$$

holds.

Proof. In order to prove this lemma by contradiction, we suppose that there exists a positive constant ε such that

$$(d_p^k)^2 \min_{1 \leq u \leq m} \lambda_p^{q_u+2} \geq \varepsilon \qquad \text{for all} \quad k. \tag{2.12}$$

Then, by (2.6), (2.9) and the orthonormality of the eigenvectors $\{v_1, v_2, ..., v_n\}$, we obtain

$$\alpha_k = \sum_{i=1}^{\ell} \phi_i \frac{(\sum_{j=1}^n d_j^{\nu_i(k)} v_j)^T A^{\rho_i(k)+3} (\sum_{j=1}^n d_j^{\nu_i(k)} v_j)}{(\sum_{j=1}^n d_j^{\nu_i(k)} v_j)^T A^{\rho_i(k)+2} (\sum_{j=1}^n d_j^{\nu_i(k)} v_j)} = \sum_{i=1}^{\ell} \phi_i \frac{\sum_{j=1}^n (d_j^{\nu_i(k)})^2 \lambda_j^{\rho_i(k)+3}}{\sum_{j=1}^n (d_j^{\nu_i(k)})^2 \lambda_j^{\rho_i(k)+2}}. \tag{2.13}$$

Since the sequences $\{d_1^k\}, \{d_2^k\}, ..., \{d_{p-1}^k\}$ converge to zero, there exists a sufficiently large $\hat{k}$ such that

$$\sum_{j=1}^{p-1} (d_j^k)^2 \max_{1 \leq u \leq m} \lambda_j^{q_u+2} \leq \frac{1}{2}\varepsilon \qquad \text{for all} \quad k \geq \hat{k}. \tag{2.14}$$

By (2.13) and (2.14), we have for all $k \geq \hat{k} + m$

$$\begin{aligned} \alpha_k &\geq \sum_{i=1}^{\ell} \phi_i \frac{\sum_{j=p}^n (d_j^{\nu_i(k)})^2 \lambda_j^{\rho_i(k)+2} \lambda_j}{\sum_{j=1}^{p-1} (d_j^{\nu_i(k)})^2 \lambda_j^{\rho_i(k)+2} + \sum_{j=p}^n (d_j^{\nu_i(k)})^2 \lambda_j^{\rho_i(k)+2}} \\ &\geq \sum_{i=1}^{\ell} \phi_i \frac{\lambda_p \sum_{j=p}^n (d_j^{\nu_i(k)})^2 \lambda_j^{\rho_i(k)+2}}{\frac{1}{2}\varepsilon + \sum_{j=p}^n (d_j^{\nu_i(k)})^2 \lambda_j^{\rho_i(k)+2}}. \end{aligned} \tag{2.15}$$

Since from (2.12) we get

$$\sum_{j=p}^{n}(d_j^{\nu_i(k)})^2\lambda_j^{\rho_i(k)+2} \geq (d_p^{\nu_i(k)})^2\lambda_p^{\rho_i(k)+2} \geq (d_p^{\nu_i(k)})^2 \min_{1\leq u\leq m}\lambda_p^{q_u+2} \geq \varepsilon \qquad \text{for all} \quad k \geq \hat{k}+m,$$

(2.4) and (2.15) yield for all $k \geq \hat{k}+m$

$$\lambda_n \geq \alpha_k \geq \sum_{i=1}^{\ell}\phi_i \frac{\lambda_p}{\frac{1}{2}\varepsilon\left(1/\sum_{j=p}^{n}(d_j^{\nu_i(k)})^2\lambda_j^{\rho_i(k)+2}\right)+1} \geq \frac{2}{3}\lambda_p,$$

which implies

$$\left|1-\frac{\lambda_p}{\alpha_k}\right| \leq \max\left(\frac{1}{2}, 1-\frac{\lambda_p}{\lambda_n}\right) \leq \max\left(\frac{1}{2}, 1-\frac{\lambda_1}{\lambda_n}\right) < 1 \qquad \text{for all} \quad k \geq \hat{k}+m. \tag{2.16}$$

Using (2.10) and (2.16) yields

$$|d_p^{k+1}| = \left|1-\frac{\lambda_p}{\alpha_k}\right||d_p^k| \leq \hat{c}_2|d_p^k| \quad \text{for all } k \geq \hat{k}+m$$

with $\hat{c}_2 = \max(1/2, 1-\lambda_1/\lambda_n) < 1$. Because this conclusion contradicts the hypothesis (2.12), we find that the lemma is true. □

By using Lemmas 2.1 and 2.2, we can prove the next theorem.

Theorem 2.3. *Let $\{x_k\}$ be the sequence generated by Algorithm EBB for problem* (1.5) *and let x_* be the unique minimizer of f. Then, either $x_j = x_*$ for some finite j, or the sequence $\{x_k\}$ converges to x_*.*

Proof. If there exists a finite integer j such that $x_j = x_*$, then this theorem is true. Hence we only consider the case $x_k \neq x_*$ for all k. From (2.9) and orthonormality of v_i $(i = 1, \ldots, n)$, we have

$$\|e_k\|^2 = \sum_{i=1}^{n}(d_i^k)^2, \tag{2.17}$$

and hence if all $\{d_i^k\}$ $(i = 1, ..., n)$ converge to zero, then the sequence $\{x_k\}$ converges to the solution. Now we prove that all $\{d_i^k\}$ $(i = 1, ..., n)$ converge to zero by the induction. Lemma 2.1 shows that $\{d_1^k\}$ converges to zero. Let assume that $\{d_1^k\}, ..., \{d_{p-1}^k\}$ all tend to zero. There exists a sufficiently large $\hat{k}$ such that

$$\sum_{j=1}^{p-1}(d_j^k)^2 \max_{1\leq u\leq m}\lambda_j^{q_u+2} \leq \frac{1}{2}\varepsilon \qquad \text{for all} \quad k \geq \hat{k}$$

for any given $\varepsilon > 0$. By Lemma 2.2, there exists a $k'(\geq \hat{k}+m)$ such that

$$\min_{0\leq t\leq m}(d_p^{k'-t})^2 \min_{1\leq u\leq m}\lambda_p^{q_u+2} < \varepsilon.$$

Let $\{\bar{k}_r\}(\geq k')$ be a sequence such that the following inequalities hold

$$\min_{0\leq t\leq m}(d_p^{\bar{k}_r-1-t})^2 \min_{1\leq u\leq m}\lambda_p^{q_u+2} < \varepsilon \quad \text{and} \quad \min_{0\leq t\leq m}(d_p^{\bar{k}_r-t})^2 \min_{1\leq u\leq m}\lambda_p^{q_u+2} \geq \varepsilon,$$

and let $\varphi(\bar{k}_r)$ be the first integer greater than $\bar{k}_r$ for which the following inequality holds

$$\min_{0\le t\le m}(d_p^{\varphi(\bar{k}_r)-t})^2 \min_{1\le u\le m}\lambda_p^{q_u+2} < \varepsilon.$$

By taking Lemma 2.2 into account, it suffices to consider the following two cases (i) and (ii).

Case (i). If the sequence $\{\bar{k}_r\}$ is a finite sequence, then there exists a sufficiently large $k''(\ge k')$ such that

$$\min_{0\le t\le m}(d_p^{k-t})^2 \min_{1\le u\le m}\lambda_p^{q_u+2} = (d_p^{k-t'})^2 \min_{1\le u\le m}\lambda_p^{q_u+2} < \varepsilon \qquad \text{for any } k \ge k'', \tag{2.18}$$

where t' is an integer which depends on k. By (2.10), (2.11) and (2.18), we have

$$\begin{aligned}
(d_p^k)^2 &= \left(\prod_{i=k-t'}^{k-1}\frac{\alpha_i-\lambda_p}{\alpha_i}\right)^2 (d_p^{k-t'})^2 \\
&\le \left(\prod_{i=k-t'}^{k-1}\frac{\lambda_n-\lambda_1}{\lambda_1}\right)^2 (d_p^{k-t'})^2 \\
&\le \max\left(\left(\frac{\lambda_n-\lambda_1}{\lambda_1}\right)^{2m}, 1\right)(d_p^{k-t'})^2 \\
&\le \max\left(\left(\frac{\lambda_n-\lambda_1}{\lambda_1}\right)^{2m}, 1\right)\frac{\varepsilon}{\min_{1\le u\le m}\lambda_p^{q_u+2}},
\end{aligned} \tag{2.19}$$

which implies that for all $k \ge k''$, the following holds

$$(d_p^k)^2 \le \hat{c}_3\varepsilon \quad \text{with} \quad \hat{c}_3 = \max\left(\left(\frac{\lambda_n-\lambda_1}{\lambda_1}\right)^{2m}, 1\right)\frac{1}{\min_{1\le u\le m}\lambda_p^{q_u+2}}. \tag{2.20}$$

Case (ii). If the sequence $\{\bar{k}_r\}$ is an infinite sequence, by the definitions of $\{\bar{k}_r\}$ and $\{\varphi(\bar{k}_r)\}$, we get

$$\min_{0\le t\le m}(d_p^{k-t})^2 \min_{1\le u\le m}\lambda_p^{q_u+2} \ge \varepsilon \qquad \text{for } k\ (\bar{k}_r \le k \le \varphi(\bar{k}_r)-1) \tag{2.21}$$

$$\min_{0\le t\le m}(d_p^{k-t})^2 \min_{1\le u\le m}\lambda_p^{q_u+2} < \varepsilon \qquad \text{for } k\ (\varphi(\bar{k}_r) \le k \le \bar{k}_{r+1}-1). \tag{2.22}$$

As shown in (2.18), (2.19) and (2.20), inequality (2.22) yields

$$(d_p^k)^2 \le \hat{c}_3\varepsilon \quad \text{for} \quad k\ (\varphi(\bar{k}_r) \le k \le \bar{k}_{r+1}-1). \tag{2.23}$$

Since (2.15) holds for all $k \ge \hat{k}+m$, we have from (2.21)

$$\begin{aligned}
\lambda_n \ge \alpha_k &\ge \sum_{i=1}^{\ell}\phi_i\frac{\lambda_p}{\frac{1}{2}\varepsilon\left(1/\sum_{j=p}^n(d_j^{\nu_i(k)})^2\lambda_j^{\rho_i(k)+2}\right)+1} \\
&\ge \sum_{i=1}^{\ell}\phi_i\frac{\lambda_p}{\frac{1}{2}\varepsilon\left(1/(d_p^{\nu_i(k)})^2\lambda_p^{\rho_i(k)+2}\right)+1} \\
&\ge \frac{2}{3}\lambda_p
\end{aligned} \tag{2.24}$$

for all k such that $\bar{k}_r \leq k \leq \varphi(\bar{k}_r) - 1$. As shown in (2.16), inequality (2.24) implies $|1 - \lambda_p/\alpha_k| < 1$, so (2.10) yields

$$|d_p^{k+1}| = \left|1 - \frac{\lambda_p}{\alpha_k}\right| |d_p^k| \leq |d_p^k| \qquad \text{for } \ k \ (\bar{k}_r \leq k \leq \varphi(\bar{k}_r) - 1). \tag{2.25}$$

Thus, by (2.25), (2.10) and (2.11), we have

$$(d_p^k)^2 \leq (d_p^{\bar{k}_r})^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_1}\right)^2 (d_p^{\bar{k}_r - 1})^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_1}\right)^2 \hat{c}_3 \varepsilon = \hat{c}_4 \varepsilon$$
$$\text{for } \ k \ (\bar{k}_r \leq k \leq \varphi(\bar{k}_r))$$

with $\hat{c}_4 = \hat{c}_3 \left\{(\lambda_n - \lambda_1)/\lambda_1\right\}^2$. The last inequality can be obtained by using (2.23).

By summarizing the cases (i) and (ii), we obtain for all $k(\geq k'')$ $(d_p^k)^2 \leq \hat{c}_5 \varepsilon$ with $\hat{c}_5 = \max(\hat{c}_3, \hat{c}_4)$. Since $\varepsilon > 0$ can be chosen arbitrarily small, we deduce $\lim\limits_{k\to\infty} |d_p^k| = 0$ as required. Therefore, by induction on p, we have $\lim\limits_{k\to\infty} |d_i^k| = 0$ for $i = 1, ..., n$ and then $\lim\limits_{k\to\infty} ||e_k|| = 0$ holds by (2.17). This completes the proof. $\square$

Note that Theorem 2.3 is the extension of Theorem 2.1 in Friedlander et al. [9]. More recently, Yuan [20] independently proved Theorem 2.3. We does not omit the proof of Theorem 2.3 because the proof in [20] is different from the proof of this theorem.

2.2 Q-superlinear Convergence

In this subsection, we analyze the local behavior of Algorithm EBB. To this end, we deal with the case where $\nu_i(k) \neq k$ and $\rho_i(k)$ does not depend on k in (2.1), say $\rho_i(k) = r_i$ for a given integer r_i $(i = 1, \ldots, \ell)$. Then (2.3) implies

$$\alpha_k = \sum_{i=1}^{\ell} \phi_i \frac{s_{\nu_i(k)}^T A^{r_i+1} s_{\nu_i(k)}}{s_{\nu_i(k)}^T A^{r_i} s_{\nu_i(k)}}, \tag{2.26}$$

where $\nu_i(k) \in \{k-1, ..., \max\{0, k-m\}\}$ for $i = 1, \ldots, \ell$.

The following theorem is the extension of Theorem 3.1 in Friedlander et al. [9].

Theorem 2.4. *Let $\{x_k\}$ be the sequence generated by Algorithm EBB with* (2.26) *for problem* (1.5). *Assume that the sequence $\{s_k/||s_k||\}$ is convergent, that is, there exists $s \in \boldsymbol{R}^n$ such that $\lim_{k\to\infty} s_k/||s_k|| = s$ and $||s|| = 1$. Then s becomes an eigenvector of A with the eigenvalue $s^T As$ and $\lim_{k\to\infty} \alpha_k = s^T As$. Moreover, the sequence $\{x_k\}$ converges Q-superlinearly to x_*.*

Proof. It follows immediately from Theorem 2.3 that $\{x_k\}$ converges to x_*. Thus, we need only show that $\{x_k\}$ converges Q-superlinearly to x_*.

Letting $A^{r_i/2} = \sum_{j=1}^n \lambda_j^{r_i/2} v_j v_j^T$, we have $(A^{r_i/2})^2 = A^{r_i}$ and $(A^{r_i/2})^T = A^{r_i/2}$ for $i = 1, ..., \ell$. Then, equation (2.26) can be written by

$$\alpha_k = \sum_{i=1}^{\ell} \phi_i \left(\frac{A^{r_i/2} s_{\nu_i(k)}}{||A^{r_i/2} s_{\nu_i(k)}||}\right)^T A \left(\frac{A^{r_i/2} s_{\nu_i(k)}}{||A^{r_i/2} s_{\nu_i(k)}||}\right). \tag{2.27}$$

For simplicity, we define

$$\hat{s}^{(i)} = \frac{A^{r_i/2}s}{||A^{r_i/2}s||} \quad \text{for} \quad i = 1, ..., \ell \quad \text{and} \quad \alpha = \sum_{i=1}^{\ell} \phi_i \hat{s}^{(i)T} A \hat{s}^{(i)}.$$

From the fact that $\nu_i(k) \geq k - m$ $(i = 1, ..., \ell)$, we get

$$\lim_{k\to\infty} \frac{A^{r_i/2}s_{\nu_i(k)}}{||A^{r_i/2}s_{\nu_i(k)}||} = \hat{s}^{(i)} \quad \text{for} \quad i = 1, ..., \ell. \tag{2.28}$$

Therefore, by (2.27) and (2.28), we have

$$\lim_{k\to\infty} \alpha_k = \sum_{i=1}^{\ell} \phi_i \hat{s}^{(i)T} A \hat{s}^{(i)} = \alpha.$$

It follows from (2.2) and $x_{k+1} = x_k + s_k$ that

$$s_{k+1} = -\frac{1}{\alpha_{k+1}}(A - \alpha_k I)s_k.$$

Premultiplying this equation by $A^{r_i/2}$ and normalizing it, we have

$$\frac{A^{r_i/2}s_{k+1}}{||A^{r_i/2}s_{k+1}||} \quad = \quad -\frac{(A - \alpha_k I)A^{r_i/2}s_k/||A^{r_i/2}s_k||}{||(A - \alpha_k I)A^{r_i/2}s_k/||A^{r_i/2}s_k||||},$$

which implies

$$\left\|(A - \alpha_k I)\frac{A^{r_i/2}s_k}{||A^{r_i/2}s_k||}\right\| \frac{A^{r_i/2}s_{k+1}}{||A^{r_i/2}s_{k+1}||} = -(A - \alpha_k I)\frac{A^{r_i/2}s_k}{||A^{r_i/2}s_k||} \quad \text{for} \quad i = 1, ..., \ell.$$

Taking limits on both sides of this equation, we have

$$||(A - \alpha I)\hat{s}^{(i)}||\hat{s}^{(i)} \quad = \quad -(A - \alpha I)\hat{s}^{(i)} \quad \text{for} \quad i = 1, ..., \ell.$$

Furthermore, premultiplying this equation by $\hat{s}^{(i)T}$ yields

$$||(A - \alpha I)\hat{s}^{(i)}|| \quad = \quad -\hat{s}^{(i)T}A\hat{s}^{(i)} + \alpha \quad \text{for} \quad i = 1, ..., \ell. \tag{2.29}$$

Thus, by (2.29) and the fact that $\sum_{i=1}^{\ell} \phi_i = 1$, we have

$$\sum_{i=1}^{\ell} \phi_i ||(A - \alpha I)\hat{s}^{(i)}|| = -\sum_{i=1}^{\ell} \phi_i \hat{s}^{(i)T} A \hat{s}^{(i)} + \alpha = 0.$$

Since there exists some j such that $\phi_j > 0$, we have

$$||(A - \alpha I)\hat{s}^{(j)}|| = 0. \tag{2.30}$$

On the other hand, we get

$$\begin{aligned} \frac{||(A - \alpha_k I)s_k||}{||s_k||} \quad &\leq \quad \frac{||A^{-r_j/2}||||(A - \alpha_k I)A^{r_j/2}s_k||}{||A^{r_j/2}s_k||} \frac{||A^{r_j/2}s_k||}{||s_k||} \\ &\leq \quad ||A^{r_j/2}||||A^{-r_j/2}|| \frac{||(A - \alpha_k I)A^{r_j/2}s_k||}{||A^{r_j/2}s_k||}. \end{aligned} \tag{2.31}$$

Therefore, using (2.31) and (2.30), we obtain

$$\lim_{k\to\infty} \frac{||(A-\alpha_k I)s_k||}{||s_k||} = 0. \tag{2.32}$$

Because we can regard $\alpha_k I$ as an approximation matrix of $\nabla^2 f(x_k)(=A)$ in Dennis and Moré condition (see [7], for example), the sequence $\{x_k\}$ converges Q-superlinearly to x_*. In addition, (2.32) yields $(A-\alpha I)s = 0$, which means that s is an eigenvector of A with the eigenvalue $\alpha = s^T As$. Therefore, the proof is complete. □

3 Extended Barzilai-Borwein Method for General Functions

In this section, we consider an application of Algorithm EBB to general unconstrained minimization problems (1.1). In (2.1), we use the positive definite matrix A which is the Hessian of the objective function. On the other hand, calculations of the Hessian of the objective function are very expensive if the objective function is not quadratic. Accordingly, we would like to express (2.26) without using the Hessian A. To this end, we fix $r_i = 0$ or 1 in (2.26) and consider the following:

$$\begin{aligned} \alpha_k &= \sum_{i=1}^{\ell} \left(\phi_i^{(1)} \frac{s_{\nu_i(k)}^T y_{\nu_i(k)}}{s_{\nu_i(k)}^T s_{\nu_i(k)}} + \phi_i^{(2)} \frac{y_{\nu_i(k)}^T y_{\nu_i(k)}}{s_{\nu_i(k)}^T y_{\nu_i(k)}} \right) \\ \phi_i^{(1)} &\geq 0, \quad \phi_i^{(2)} \geq 0, \quad \sum_{i=1}^{\ell} (\phi_i^{(1)} + \phi_i^{(2)}) = 1, \quad \nu_i(k) \in \{k-1, ..., \max\{0, k-m\}\}, \end{aligned} \tag{3.1}$$

where ℓ and m are positive integers. We note that the first and the second term in (3.1) correspond to the cases $r_i = 0$ and $r_i = 1$, respectively. Since (3.1) does not explicitly use the matrix A, it can be applied to general objective functions.

For general unconstrained minimization problems, we should use globalization technique. Since α_k in (3.1) is not necessarily positive, i.e. the direction $-(1/\alpha_k)g_k$ is not necessarily a descent search direction of the objective function, it is appropriate to use a nonmonotone line search, which was originally developed by Grippo et al. [10, 11] for Newton type methods. Recently, several researchers applied the nonmonotone line search to gradient-based methods, and obtained efficient methods for large-scale unconstrained optimization problems. For example, Dai [2] showed the global convergence of the nonmonotone conjugate gradient method, and Raydan [17] proved the global convergence of the nonmonotone Barzilai-Borwein method. Moreover, Grippo and Sciandrone [12] proposed another type of the nonmonotone Barzilai-Borwein method. Dai [3] gives the basic analysis of the nonmonotone line search strategy.

The proposed algorithm with the nonmonotone line search is given by the following:

Algorithm NEBB.

Step 0 . Give $x_0 \in \boldsymbol{R}^n$ and $\ell,\ m \in \boldsymbol{N}$. Set $k = 0$, $0 < \bar{\alpha} \ll 1$, $\delta > 0$, $0 < \eta_1 \leq \eta_2$, $0 < \eta_3 \leq \eta_4 < 1$ and $\xi \in (0,1)$, and let $\bar{M}$ be a positive integer. Go to Step 1.

Step 1 . Compute α_k by (3.1). If $\bar{\alpha} \leq \alpha_k \leq \dfrac{1}{\bar{\alpha}}$, set $p_k = -\dfrac{1}{\alpha_k} g_k$, and otherwise set $p_k = -\delta g_k$.

Step 2 . Give $t_k^{(0)} \in [\eta_1,\ \eta_2]$ and $M(k)$ such that $M(0) = 0$ and $0 \leq M(k) \leq \min\{M(k-1)+1, \bar{M}\}$ if $k \geq 1$. Set $i = 0$ and go to Step 2.1.

Step 2.1 . If $f(x_k + t_k^{(i)} p_k) \leq \max_{0\leq j\leq M(k)}\{f(x_{k-j})\} + \xi t_k^{(i)} g_k^T p_k$ holds, set $t_k \equiv t_k^{(i)}$ and go to Step 3.

Step 2.2 . Choose $\sigma_k^{(i)} \in [\eta_3,\ \eta_4]$ and compute $t_k^{(i+1)}$ such that $t_k^{(i+1)} = t_k^{(i)} \sigma_k^{(i)}$.

Step 2.3 . Set $i := i + 1$ and go to Step 2.1.

Step 3 . Let $x_{k+1} = x_k + t_k p_k$. If the stopping criterion is satisfied, then stop.

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

In Step 2, we usually choose $t_k^{(0)} = 1$. Since we choose a small value as $\bar{\alpha}$, $p_k = -\dfrac{1}{\alpha_k} g_k$ would be chosen in almost all iterations as far as $\alpha_k > 0$. We note that the search direction p_k satisfies

$$g_k^T p_k \leq -c_1 \|g_k\|^2 \quad \text{and} \quad \|p_k\| \leq c_2 \|g_k\| \quad \text{for all } k \tag{3.2}$$

for some positive constants c_1 and c_2. These relations lead to the following theorem.

Theorem 3.1. *Assume that the objective function f is bounded below on $\boldsymbol{R}^n$ and is continuously differentiable in a neighborhood $\mathcal{N}$ of the level set $\mathcal{L} = \{x \in \boldsymbol{R}^n : f(x) \leq f(x_0)\}$. We also assume that the gradient g is Lipschitz continuous in $\mathcal{N}$. Let the sequence $\{x_k\}$ be generated by Algorithm NEBB. Then our method converges in the sense that*

$$\lim_{k\to\infty} \|g_k\| = 0.$$

Proof. From (3.2) and Theorem 2.1 of Dai [3], we have the results immediately. □

In the rest of this section, we consider the local behavior of Algorithm NEBB for general functions. For this purpose, we make the following assumptions. In what follows, we denote $\nabla^2 f$ by H, and $\nabla^2 f(x_*)$ by H_*.

Assumption 3.2.

1. The objective function f is twice continuously differentiable in an open convex neighborhood $\mathcal{N}$ of the local solution x_*. In addition, there exist positive constants m_1 and m_2 such that

$$m_1 \|v\|^2 \leq v^T H(x) v \leq m_2 \|v\|^2 \qquad \text{for all } x \in \mathcal{N} \text{ and } v \in \boldsymbol{R}^n. \tag{3.3}$$

2. In Step 2 of Algorithm NEBB, $t_k = 1$ is chosen for k sufficiently large. The parameter $\bar{\alpha}$ satisfies $\bar{\alpha} \leq m_1$ and $m_2 \leq \dfrac{1}{\bar{\alpha}}$.

3. The sequence $\{x_k\}$ generated by Algorithm NEBB converges to the solution x_*.

Under Assumption 3.2, we obtain the following theorem.

Theorem 3.3. *Let $\{x_k\}$ be the sequence generated by Algorithm NEBB. Suppose that Assumption 3.2 holds, and that the sequence $\{s_k/\|s_k\|\}$ is convergent, that is, there exists $s \in \boldsymbol{R}^n$ such that $\lim_{k\to\infty} s_k/||s_k|| = s$ and $||s|| = 1$. Then s becomes an eigenvector of H_* with the eigenvalue $s^T H_* s$ and $\lim_{k\to\infty} \alpha_k = s^T H_* s$. Moreover, the sequence $\{x_k\}$ converges Q-superlinearly to x_*.*

Proof. We assume that k is sufficiently large. From Assumption 3.2, $x_k \in \mathcal{N}$ for all k. By the mean value theorem, we have $y_k = \int_0^1 H(x_k + ts_k)s_k \; dt$. Since from (3.3) $H(x)$ is symmetric positive definite in $\mathcal{N}$, $H(x)^{1/2}$ is well-defined in $\mathcal{N}$. We define $\tilde{H}_k \equiv \int_0^1 H(x_k + ts_k) \; dt$ and $\tilde{s}_k \equiv \tilde{H}_k^{1/2} s_k$. Then (3.1) yields

$$\begin{aligned}
\alpha_k &= \sum_{i=1}^{\ell} \left\{ \phi_i^{(1)} \frac{s_{\nu_i(k)}^T \tilde{H}_{\nu_i(k)} s_{\nu_i(k)}}{s_{\nu_i(k)}^T s_{\nu_i(k)}} + \phi_i^{(2)} \frac{\tilde{s}_{\nu_i(k)}^T \tilde{H}_{\nu_i(k)} \tilde{s}_{\nu_i(k)}}{\tilde{s}_{\nu_i(k)}^T \tilde{s}_{\nu_i(k)}} \right\} \\
&= \sum_{i=1}^{\ell} \left\{ \phi_i^{(1)} \left(\frac{s_{\nu_i(k)}}{\|s_{\nu_i(k)}\|} \right)^T \tilde{H}_{\nu_i(k)} \left(\frac{s_{\nu_i(k)}}{\|s_{\nu_i(k)}\|} \right) + \phi_i^{(2)} \left(\frac{\tilde{s}_{\nu_i(k)}}{\|\tilde{s}_{\nu_i(k)}\|} \right)^T \tilde{H}_{\nu_i(k)} \left(\frac{\tilde{s}_{\nu_i(k)}}{\|\tilde{s}_{\nu_i(k)}\|} \right) \right\}. \quad (3.4)
\end{aligned}$$

It follows from the definition of $\tilde{s}_{\nu_i(k)}$ that

$$\tilde{s} \equiv \lim_{k\to\infty} \frac{\tilde{s}_{\nu_i(k)}}{\|\tilde{s}_{\nu_i(k)}\|} = \lim_{k\to\infty} \frac{\tilde{H}_{\nu_i(k)}^{1/2} s_{\nu_i(k)}/\|s_{\nu_i(k)}\|}{\|\tilde{H}_{\nu_i(k)}^{1/2} s_{\nu_i(k)}\|/\|s_{\nu_i(k)}\|} = \frac{H_*^{1/2} s}{\|H_*^{1/2} s\|}.$$

Therefore, by taking limit in (3.4), we obtain

$$\alpha \equiv \lim_{k\to\infty} \alpha_k = \sum_{i=1}^{\ell} \left(\phi_i^{(1)} s^T H_* s + \phi_i^{(2)} \tilde{s}^T H_* \tilde{s} \right). \quad (3.5)$$

On the other hand, (3.3), (3.4) and Assumption 3.2 yield $\bar{\alpha} \le m_1 \le \alpha_k \le m_2 \le 1/\bar{\alpha}$. Thus, it follows that

$$p_k = -\frac{1}{\alpha_k} g_k, \quad x_{k+1} = x_k - \frac{1}{\alpha_k} g_k \quad \text{and} \quad s_k = -\frac{1}{\alpha_k} g_k \quad (3.6)$$

hold. By using the mean value theorem, we have

$$g_k = g(x_*) + \int_0^1 H(x_* + t(x_k - x_*))(x_k - x_*) \; dt = -\int_0^1 H(x_* - te_k) \; dt \; e_k, \quad (3.7)$$

where $e_k = x_* - x_k$. Set $\hat{H}_k \equiv \int_0^1 H(x_* - te_k) \; dt$. Since (3.6) and (3.7) yield

$$s_k = -\frac{1}{\alpha_k} g_k = \frac{1}{\alpha_k} \hat{H}_k e_k, \quad (3.8)$$

we have

$$e_{k+1} = e_k - s_k = e_k - \frac{1}{\alpha_k} \hat{H}_k e_k = \left(I - \frac{1}{\alpha_k} \hat{H}_k \right) e_k. \quad (3.9)$$

It follows from (3.8) and (3.9) that

$$\begin{aligned}
s_{k+1} &= \frac{1}{\alpha_{k+1}} \hat{H}_{k+1} \left(I - \frac{1}{\alpha_k} \hat{H}_k \right) e_k \\
&= \frac{1}{\alpha_{k+1}} \hat{H}_{k+1} \left(I - \frac{1}{\alpha_k} \hat{H}_k \right) \alpha_k \hat{H}_k^{-1} s_k \\
&= -\frac{1}{\alpha_{k+1}} \hat{H}_{k+1} \hat{H}_k^{-1} (\hat{H}_k - \alpha_k I) s_k. \quad (3.10)
\end{aligned}$$

We normalize the above equation, and we get

$$\frac{s_{k+1}}{\|s_{k+1}\|} = -\frac{\hat{H}_{k+1}\hat{H}_k^{-1}(\hat{H}_k - \alpha_k I)s_k}{\|\hat{H}_{k+1}\hat{H}_k^{-1}(\hat{H}_k - \alpha_k I)s_k\|},$$

which implies

$$\left\|\hat{H}_{k+1}\hat{H}_k^{-1}(\hat{H}_k - \alpha_k I)\frac{s_k}{\|s_k\|}\right\| \frac{s_{k+1}}{\|s_{k+1}\|} = -\hat{H}_{k+1}\hat{H}_k^{-1}(\hat{H}_k - \alpha_k I)\frac{s_k}{\|s_k\|}.$$

Taking limits on both sides of this equation, we have

$$\|(H_* - \alpha I)s\|s = -(H_* - \alpha I)s,$$

and hence, premultiplying this equation by s^T, we have from $\|s\| = 1$

$$\|(H_* - \alpha I)s\| = -s^T H_* s + \alpha. \tag{3.11}$$

Moreover, since (3.10) yields $H_*^{1/2}s_{k+1} = -\frac{1}{\alpha_{k+1}}H_*^{1/2}\hat{H}_{k+1}\hat{H}_k^{-1}(\hat{H}_k - \alpha_k I)s_k$, we also have, in a similar way,

$$\|(H_* - \alpha I)\tilde{s}\| = -\tilde{s}^T H_* \tilde{s} + \alpha. \tag{3.12}$$

Therefore, from (3.5), (3.11) and (3.12), we get

$$\begin{aligned} \sum_{i=1}^{\ell}(\phi_i^{(1)}\|(H_* - \alpha I)s\| + \phi_i^{(2)}\|(H_* - \alpha I)\tilde{s}\|) &= -\sum_{i=1}^{\ell}(\phi_i^{(1)}s^T H_* s + \phi_i^{(2)}\tilde{s}^T H_* \tilde{s}) + \alpha \\ &= 0, \end{aligned}$$

which implies that either $\|(H_* - \alpha I)s\| = 0$ or $\|(H_* - \alpha I)\tilde{s}\| = 0$ holds. Since conditions $\|(H_* - \alpha I)s\| = 0$ and $\|(H_* - \alpha I)\tilde{s}\| = 0$ are equivalent, we consider only the case $\|(H_* - \alpha I)s\| = 0$. Thus we obtain

$$\lim_{k\to\infty}\frac{\|(H_* - \alpha_k I)s_k\|}{\|s_k\|} = \|(H_* - \alpha I)s\| = 0. \tag{3.13}$$

Because we can regard $\alpha_k I$ as an approximation matrix of $\nabla^2 f(x_k)$ in Dennis and Moré condition (see [7], for example), the sequence $\{x_k\}$ converges Q-superlinearly to x_*. In addition, (3.13) yields $(H_* - \alpha I)s = 0$. This means that s is an eigenvector of H_* with the eigenvalue $\alpha = s^T H_* s$. Therefore, the proof is complete. □

4 Numerical Experiments

In this section, we present some numerical results of Algorithms EBB and NEBB to compare with other methods. Since the steepest descent method converged very slowly, we omit its numerical results. Moreover, we investigate how a choice of the parameters included in our methods affects numerical performance.

In our numerical experiments, we set $\ell = 2$ and $r_1 = r_2 (= r)$ in (2.26). Moreover, we fix $r = 0$ or 1. Thus α_k is rewritten by the forms

- $r = 0$
$$\alpha_k = \phi_1 \frac{s_{\nu_1(k)}^T y_{\nu_1(k)}}{s_{\nu_1(k)}^T s_{\nu_1(k)}} + \phi_2 \frac{s_{\nu_2(k)}^T y_{\nu_2(k)}}{s_{\nu_2(k)}^T s_{\nu_2(k)}}, \quad \phi_1 + \phi_2 = 1, \quad \phi_1 \geq 0,\ \phi_2 \geq 0$$
- $r = 1$
$$\alpha_k = \phi_1 \frac{y_{\nu_1(k)}^T y_{\nu_1(k)}}{s_{\nu_1(k)}^T y_{\nu_1(k)}} + \phi_2 \frac{y_{\nu_2(k)}^T y_{\nu_2(k)}}{s_{\nu_2(k)}^T y_{\nu_2(k)}}, \quad \phi_1 + \phi_2 = 1, \quad \phi_1 \geq 0,\ \phi_2 \geq 0.$$

As mentioned in Section 2, if we choose $\phi_1 = 1,\ \phi_2 = 0,\ r = 0$, and $\nu_1(k) = k - 1$, then it becomes the Barzilai-Borwein method, and if we choose $\phi_1 = 1$ and $\phi_2 = 0$, then it becomes a gradient method with retards.

Following Dai et al. [4], we used the following choice of $\nu_i(k)$:

$$\nu_i(k) \quad = \quad M_c \left\lfloor \frac{k - m_i}{M_c} \right\rfloor, \tag{4.1}$$

where $m_i (i = 1,\ 2)$ are positive integers. In this section, we call Algorithms EBB and NEBB with (4.1) *cyclic EBB* and *cyclic NEBB*, respectively. If $\phi_1 = 1$, $\phi_2 = 0$, $m_1 = 1$ and $r = 0$, we see that

$$\alpha_k = \frac{s_{\nu_1(k)}^T y_{\nu_1(k)}}{s_{\nu_1(k)}^T s_{\nu_1(k)}} \quad \text{and} \quad \nu_1(k) = M_c \left\lfloor \frac{k-1}{M_c} \right\rfloor,$$

which is the cyclic Barzilai-Borwein method. In each experiment, we set $\alpha_0 = 1$. The parameters used in our experiments are described in each table. Note that the values of parameters $\nu_i(k), M_c$ and m_i $(i = 1,\ 2)$ indicate how old information we use. For example, if we choose $\nu_1(k) = k - 5$ and $\nu_2(k) = k - 6$, we use g_{k-5} and g_{k-6} at the k-th iteration, and if we choose $M_c = 5$, $m_1 = 3$ and $m_2 = 4$, we use g_{k-9} according to circumstances.

We used the following stopping condition:

$$\|g_k\| \leq 10^{-5}.$$

4.1 Numerical Results of Algorithm EBB for (1.5)

In this subsection, we give some numerical results of Algorithm EBB. The objective function we used is

$$f(x) = \frac{1}{2} x^T A x, \quad x \in \boldsymbol{R}^n.$$

The following matrices were chosen as the matrix A:

- Diag: the diagonal matrix defined by
$$\mathrm{diag}\left\{1,\ \frac{\lambda_n}{n} 2,\ ...,\ \frac{\lambda_n}{n} i,\ ...,\ \lambda_n\right\}$$
- Hilbert: the Hilbert matrix.

- bcsstm: symmetric positive definite matrices in Matrix Market [13].

We set $x_0 = (1, ..., 1)^T$ as a starting point.

The numerical results of Algorithm EBB are summarized in Tables 1–3. We give the number of iterations in each table, and " Sum " denotes the sum of the number of iterations in each column. In addition, " Failed " means that the number of iterations exceeds 10000. In each column, if " Failed " occurred, then we wrote " $*$ " in " Sum ".

From Table 1, we see the following observations.

- By comparing each "Sum", the method with $(r, \phi_1, \phi_2, \nu_1(k), \nu_2(k)) = (1, 1, 0, k-3, -)$ performed well. In addition, the methods with $(r, \phi_1, \phi_2, \nu_1(k), \nu_2(k)) = (0, 1, 0, k-3, -)$, $(1, 0.25, 0.75, k-3, k-4)$, $(1, 0.75, 0.25, k-3, k-4)$ also performed well.

- For the cases $\nu_1(k) = k-1$ and $\nu_2(k) = k-2$, our methods did not converge to the solution occasionally.

- Choices of $\nu_1(k)$, $\nu_2(k)$ and r affected the numerical results more than choices of ϕ_1 and ϕ_2 did.

From Tables 2 and 3, we see the following observations.

- The cyclic EBB with $(M_c, m_1, m_2) = (3, 3, 4)$ and $(3, 3, -)$ (which means ϕ_1, ϕ_2 and r are any parameters) performed better than other methods.

- For the cases $(M_c, m_1, m_2) = (3, 1, 2)$, our methods did not converge to the solution occasionally.

Summarizing our numerical results, we conclude that the numerical performance of our method was greatly affected by the choice of $\nu_i(k)$ or (M_c, m_i). Taking into account that the steepest descent method is involved in the case $\nu_1(k) = k$ (it means current information), we see that our method with old information performed better than that with current or near current information. However, if we use too old information, then our method becomes unstable. It is important to find proper choices of $\nu_i(k)$ or (M_c, m_i). In our numerical results, EBB with $(\nu_1(k), \nu_2(k)) = (k-3, k-4)$, and the cyclic EBB with $(M_c, m_1, m_2) = (3, 3, 4)$ performed well. On the other hand, the choices of the other parameters also affected the numerical performance of our method, but we cannot observe any remarkable tendency.

4.2 Numerical Results of Algorithm NEBB for (1.1)

In this subsection, we give some numerical results of Algorithm NEBB. The test problems we used are described in Grippo et al. [11] and Moré et al. [14]. In Table 4, the first column, the second column, the third column and the fourth column denote the problem number used in this paper, the problem name, the dimension of the problem and the references, respectively.

The numerical results of Algorithm NEBB are summarized in Tables 5–7. In Algorithm NEBB, we set $\bar{\alpha} = 10^{-16}$, $\delta = 1$, $\xi = 0.0001$, $t_k^{(0)} = 1$, $\bar{M} = 10$, $\sigma_k^{(i)} = 0.5$. The numerical results are given in the form of "the number of iterations / the number of function evaluations", and " Sum I " and " Sum F " denote the sum of the number of iterations and the sum of the number of function evaluations, respectively. We note that the number of gradient evaluations is the same as the number of iterations. In addition, " Failed " means that the number of iterations exceeds 1000.

Table 1: Numerical results of EBB

r		0	1	0	1	0	1	0	1
ϕ_1		1	1	1	1	0.5	0.5	0.5	0.5
ϕ_2		0	0	0	0	0.5	0.5	0.5	0.5
$\nu_1(k)$		$k-1$	$k-1$	$k-3$	$k-3$	$k-1$	$k-1$	$k-3$	$k-3$
$\nu_2(k)$		–	–	–	–	$k-2$	$k-2$	$k-4$	$k-4$
P	n								
Diag ($\lambda_n = 1000$)	1000	242	271	223	299	220	314	251	241
Diag ($\lambda_n = 10000$)	1000	331	351	353	367	315	298	314	315
Hilbert	100	104	95	124	162	276	183	132	85
Hilbert	1000	213	209	223	247	332	368	293	211
bcsstm19	817	9559	7528	6938	6089	Failed	9876	7167	6479
bcsstm20	485	6310	6494	3515	3613	8970	7297	5782	5641
bcsstm21	3600	10	10	12	6	9	10	12	6
bcsstm22	138	64	79	129	71	72	67	95	70
bcsstm26	1922	1509	1594	1743	1344	2228	1502	1259	1163
Sum		18342	16631	13260	12198	$*$	19915	15305	14211

r		0	0	1	1	0	0	1	1
ϕ_1		0.25	0.75	0.25	0.75	0.25	0.75	0.25	0.75
ϕ_2		0.75	0.25	0.75	0.25	0.75	0.25	0.75	0.25
$\nu_1(k)$		$k-1$	$k-1$	$k-1$	$k-1$	$k-3$	$k-3$	$k-3$	$k-3$
$\nu_2(k)$		$k-2$	$k-2$	$k-2$	$k-2$	$k-4$	$k-4$	$k-4$	$k-4$
P	n								
Diag ($\lambda_n = 1000$)	1000	297	259	280	240	265	292	242	262
Diag ($\lambda_n = 10000$)	1000	382	320	301	305	350	315	323	359
Hilbert	100	201	167	217	136	123	125	162	108
Hilbert	1000	394	264	341	311	232	243	249	224
bcsstm19	817	Failed	Failed	Failed	7937	8131	6840	6577	5693
bcsstm20	485	Failed	7919	Failed	8978	5406	5519	5007	5703
bcsstm21	3600	9	10	6	10	12	12	6	6
bcsstm22	138	75	81	63	59	68	104	67	79
bcsstm26	1922	1831	1502	2367	1748	1224	1754	1133	1367
Sum		$*$	$*$	$*$	19724	15811	15204	13766	13801

Table 2: Numerical results of cyclic EBB with $M_c = 3$

r		0	1	0	1	0	1	0	1
ϕ_1		1	1	1	1	0.5	0.5	0.5	0.5
ϕ_2		0	0	0	0	0.5	0.5	0.5	0.5
M_c		3	3	3	3	3	3	3	3
m_1		1	1	3	3	1	1	3	3
m_2		–	–	–	–	2	2	4	4
P	n								
Diag ($\lambda_n = 1000$)	1000	254	359	287	311	263	323	329	265
Diag ($\lambda_n = 10000$)	1000	320	308	351	362	332	350	297	376
Hilbert	100	209	143	116	141	128	194	134	122
Hilbert	1000	380	260	221	275	386	240	236	300
bcsstm19	817	8008	7760	5483	5156	Failed	Failed	5197	6065
bcsstm20	485	6575	5842	3776	3341	6542	7805	3692	3575
bcsstm21	3600	11	11	13	6	11	11	12	6
bcsstm22	138	87	68	142	68	80	72	98	62
bcsstm26	1922	1593	1559	2289	1760	2036	2038	1553	2147
Sum		17437	16310	12678	11420	$*$	$*$	11548	12918

r		0	0	1	1	0	0	1	1
ϕ_1		0.25	0.75	0.25	0.75	0.25	0.75	0.25	0.75
ϕ_2		0.75	0.25	0.75	0.25	0.75	0.25	0.75	0.25
M_c		3	3	3	3	3	3	3	3
m_1		1	1	1	1	3	3	3	3
m_2		2	2	2	2	4	4	4	4
P	n								
Diag ($\lambda_n = 1000$)	1000	255	305	262	236	324	302	272	299
Diag ($\lambda_n = 10000$)	1000	426	363	359	356	344	314	369	357
Hilbert	100	228	144	158	203	116	116	122	125
Hilbert	1000	401	263	404	311	374	317	227	278
bcsstm19	817	Failed	8946	Failed	7073	6200	4983	6344	5723
bcsstm20	485	8132	7993	7976	6893	3807	3809	3647	3539
bcsstm21	3600	11	11	11	11	12	13	6	6
bcsstm22	138	81	71	62	65	140	92	62	65
bcsstm26	1922	1857	1692	1742	1369	1396	1775	1340	1398
Sum		$*$	19788	$*$	16517	12713	11721	12389	11790

Table 3: Numerical results of cyclic EBB with $M_c = 5$

r		0	1	0	1	0	1	0	1
ϕ_1		1	1	1	1	0.5	0.5	0.5	0.5
ϕ_2		0	0	0	0	0.5	0.5	0.5	0.5
M_c		5	5	5	5	5	5	5	5
m_1		1	1	3	3	1	1	3	3
m_2		–	–	–	–	2	2	4	4
P	n								
Diag ($\lambda_n = 1000$)	1000	294	322	303	307	302	277	302	282
Diag ($\lambda_n = 10000$)	1000	412	353	353	354	353	352	334	362
Hilbert	100	162	137	Failed	182	112	117	237	112
Hilbert	1000	302	282	467	Failed	397	232	302	282
bcsstm19	817	6657	6312	6515	6717	7182	7212	7042	7007
bcsstm20	485	3963	3737	4277	4452	5012	4335	5087	4624
bcsstm21	3600	12	12	13	6	12	12	13	6
bcsstm22	138	102	83	72	107	75	87	72	62
bcsstm26	1922	1587	1797	1442	1857	1697	1422	1727	1897
Sum		13491	13035	$*$	$*$	15142	14046	15116	14634

r		0	0	1	1	0	0	1	1
ϕ_1		0.25	0.75	0.25	0.75	0.25	0.75	0.25	0.75
ϕ_2		0.75	0.25	0.75	0.25	0.75	0.25	0.75	0.25
M_c		5	5	5	5	5	5	5	5
m_1		1	1	1	1	3	3	3	3
m_2		2	2	2	2	4	4	4	4
P	n								
Diag ($\lambda_n = 1000$)	1000	298	312	292	306	306	290	287	290
Diag ($\lambda_n = 10000$)	1000	352	405	328	357	377	353	433	338
Hilbert	100	137	102	127	182	147	172	237	142
Hilbert	1000	212	307	227	227	323	Failed	317	347
bcsstm19	817	6442	6332	5622	5882	6608	6382	6577	6221
bcsstm20	485	4987	4862	5202	5612	6208	5591	4687	4487
bcsstm21	3600	12	12	12	12	13	13	6	6
bcsstm22	138	103	102	74	82	127	88	62	67
bcsstm26	1922	1658	1552	1742	1527	1678	2048	1912	1447
Sum		14201	13986	13626	14187	15787	$*$	14518	13345

Table 4: Test problems

P	Name	Dimension	References
1	Extended Rosenbrock Function	$n = 10000$	Moré et al. [14]
2	Extended Powell Singular Function	$n = 10000$	Moré et al. [14]
3	Trigonometric Function	$n = 10000$	Moré et al. [14]
4	Broyden Tridiagonal Function	$n = 10000$	Moré et al. [14]
5	Oren Function	$n = 100$	Grippo et al. [11]
6	Cube Function	$n = 2$	Grippo et al. [11]
7	Wood Function	$n = 4$	Moré et al. [14]
8	Beale Function	$n = 2$	Moré et al. [14]
9	Helical Valley Function	$n = 3$	Moré et al. [14]
10	Jennrich and Sampson Function	$n = 2$	Moré et al. [14]
11	Freudenstein and Roth Function	$n = 2$	Moré et al. [14]

In order to compare our method with conjugate gradient (CG) methods, we examined typical CG methods (Fletcher-Reeves (FR) method, Hestenes-Stiefel (HS) method, Polak-Ribière Plus (PR+) method, and Dai-Yuan (DY) method, see [15] for example). It is reasonable to use a monotone line search for CG methods. Thus we used the Armijo condition and the bisection method in the line search procedure, which means Step 2 of Algorithm NEBB with $\xi = 0.1$, $M(k) = 0$, $t_k^{(0)} = 1$ and $\sigma_k^{(i)} = 0.5$. In each iteration, if CG methods did not generate a descent direction, then we used the steepest descent direction. However such a case rarely occurred. The CG methods, for Problems 4 and 5, did not converge to the solution. So we omit these numerical results. The numerical results of CG methods are given in Table 8.

For Algorithm NEBB, we investigate the frequency of taking $t_k = 1$, namely $x_{k+1} = x_k - 1/\alpha_k g_k$. The frequency of taking $t_k = 1$ depended on problems and the choice of parameters. The ratio (the frequency of taking $t_k = 1$/the number of iterations) are 65% – 100%. In Tables 5–7, the averages of the ratio are 85%, 82% and 79%, respectively. It seems that the older information becomes, the lower the ratio becomes.

From Tables 5–7, we see the following observations.

- NEBB with $(r, \phi_1, \phi_2, \nu_1(k), \nu_2(k)) = (1, 0.5, 0.5, k-1, k-2)$ and $(1, 0.25, 0.75, k-1, k-2)$ performed better than the other variants.
- NEBB with $(\nu_1(k), \nu_2(k)) = (k-1, k-2)$ needed the number of function evaluations less than NEBB with $(\nu_1(k), \nu_2(k)) = (k-3, k-4)$.
- The cyclic NEBB with $(r, M_c, m_1, m_2) = (0, 3, 1, 2), (0, 5, 3, -)$ and $(0, 5, 3, 4)$ performed very poorly for Problem 2.

Summarizing our numerical results, we conclude that the numerical performance of our method was greatly affected by not only the choice of $\nu_i(k)$ or (M_c, m_i) but also r. Especially, we find that the choice $r = 1$ is more appropriate than the choice $r = 0$ for general objective functions. It seems that the older information becomes, the more the number of function evaluations we need. We recommend NEBB with $(r, \phi_1, \phi_2, \nu_1(k), \nu_2(k)) = (1, 0.5, 0.5, k-1, k-2)$ and $(1, 0.25, 0.75, k-1, k-2)$. By comparing NEBB (with $(r, \phi_1, \phi_2, \nu_1(k), \nu_2(k)) = (1, 0.5, 0.5, k-1, k-2)$ and $(1, 0.25, 0.75, k-1, k-2)$) with CG methods, NEBB needed the number of iterations more than CG methods, while NEBB is superior to CG methods from the viewpoint of the number of function evaluations. When the number of variables is very large, the computational effort is sometimes dominated by the cost of evaluating the function value and the cost of evaluating the gradient. Therefore we can regard our methods as efficient methods for large scale problems.

Finally, in order to investigate the local behavior of NEBB, we tested the following convex function:

$$f(x) = \frac{1}{2} x^T A x + \|x\|^4 + e^{\|x\|^2/n^2}, \tag{4.2}$$

where $n = 1000$. Here we set

$$A = A_1 = \begin{pmatrix} c & 1 & & & 0 \\ 1 & c & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & c & 1 \\ 0 & & & 1 & c \end{pmatrix} \quad \text{or} \quad A = A_2 = \begin{pmatrix} c & 1 & \cdots & 1 \\ 1 & c & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & c \end{pmatrix}$$

Table 5: Numerical results of NEBB

r		0	1	0	1	0	1	0	1
ϕ_1		1	1	1	1	0.5	0.5	0.5	0.5
ϕ_2		0	0	0	0	0.5	0.5	0.5	0.5
$\nu_1(k)$		$k-1$	$k-1$	$k-3$	$k-3$	$k-1$	$k-1$	$k-3$	$k-3$
$\nu_2(k)$		–	–	–	–	$k-2$	$k-2$	$k-4$	$k-4$
P	n								
1	10000	108/243	119/190	147/503	120/310	81/129	78/104	85/226	119/281
2	10000	338/819	253/486	490/1782	429/1312	477/799	226/303	466/1432	368/937
3	10000	67/83	67/73	60/96	92/108	68/74	77/78	64/102	78/92
4	10000	219/274	361/475	Failed	116/184	78/89	84/90	120/176	379/654
5	100	114/182	101/143	121/214	116/176	112/148	97/118	122/203	117/174
6	2	70/162	71/130	80/328	93/343	105/148	67/84	88/269	69/201
7	4	286/585	161/242	402/1142	185/444	751/1214	214/248	262/611	173/331
8	2	8/13	8/13	9/14	9/14	7/12	7/12	9/14	9/14
9	3	14/21	17/24	19/27	20/27	15/22	14/21	18/26	14/21
10	2	27/41	21/30	33/71	32/65	18/25	29/36	31/60	32/55
11	2	69/145	53/99	66/200	72/217	70/109	36/51	63/166	60/147
Sum I		1320	1232	$*$	1284	1782	929	1328	1418
Sum F		2568	1905	$*$	3200	2769	1145	3285	2907

r		0	0	1	1	0	0	1	1
ϕ_1		0.25	0.75	0.25	0.75	0.25	0.75	0.25	0.75
ϕ_2		0.75	0.25	0.75	0.25	0.75	0.25	0.75	0.25
$\nu_1(k)$		$k-1$	$k-1$	$k-1$	$k-1$	$k-3$	$k-3$	$k-3$	$k-3$
$\nu_2(k)$		$k-2$	$k-2$	$k-2$	$k-2$	$k-4$	$k-4$	$k-4$	$k-4$
P	n								
1	10000	81/131	58/99	79/97	106/151	107/309	62/151	112/254	111/252
2	10000	Failed	338/559	274/337	253/347	504/1353	445/1225	349/821	424/959
3	10000	52/59	66/74	74/75	76/77	64/92	75/120	76/92	84/99
4	10000	71/78	91/100	106/111	234/259	110/160	217/406	299/490	563/922
5	100	116/157	121/168	83/104	96/121	124/221	137/235	110/162	114/172
6	2	93/140	89/134	48/59	61/77	89/303	84/269	68/187	103/305
7	4	377/621	234/341	151/177	224/286	373/923	460/1183	215/436	178/361
8	2	8/13	8/13	8/13	8/13	9/14	9/14	9/14	9/14
9	3	20/27	16/23	13/20	17/24	17/25	19/27	19/26	15/22
10	2	28/35	22/31	22/29	24/31	29/59	32/62	31/50	32/57
11	2	65/103	77/138	42/57	46/65	57/155	75/212	69/167	59/153
Sum I		$*$	1120	900	1145	1483	1615	1357	1692
Sum F		$*$	1680	1079	1451	3614	3904	2699	3316

where c is chosen such that A becomes at least positive semidefinite. We tested NEBB with $(r, \phi_1, \phi_2, \nu_1(k), \nu_2(k)) = (1, 0.5, 0.5, k-1, k-2)$ for the problem (4.2) with various values of c. We should note the following:

- a solution of the problem (4.2) is $x_* = 0$,
- NEBB did not converge within "1000" iterations for the case $A = A_1$ with $c = 2$,
- A_2 with $c = 1$ is not positive definite but positive semidefinite.

Figure 1 gives the behavior of $\log_{10} \|x_k - x_*\|$ for the problem (4.2) with $A = A_1$ and $c = 2.1, 10, 100$, and Figure 2 gives the behavior of $\log_{10} \|x_k - x_*\|$ for the problem (4.2) with $A = A_2$ and $c = 1, 10, 100, 1000, 10000$. In each figure, (b) is the same as (a) except for the scale of transverse axis.

From Figures 1 and 2, we see that NEBB did not achieve fast rate of convergence when $\nabla^2 f(x_*)$ is ill-conditioned. On the other hand, NEBB achieved fast rate of convergence for large c.

5 Concluding Remarks

In this paper, we have proposed the extended Barzilai-Borwein method which includes the steepest descent method, the Barzilai-Borwein method and the gradient method with retards. We have established the global and Q-superlinear convergence properties of the proposed method. Moreover, numerical performance of our method has been investigated by numerical experiments. Our further interests are to find a suitable choice of parameters included in our method.

Table 6: Numerical results of cyclic NEBB with $M_c = 3$

r		0	1	0	1	0	1	0	1
ϕ_1		1	1	1	1	0.5	0.5	0.5	0.5
ϕ_2		0	0	0	0	0.5	0.5	0.5	0.5
M_c		3	3	3	3	3	3	3	3
m_1		1	1	3	3	1	1	3	3
m_2		–	–	–	–	2	2	4	4
P	n								
1	10000	119/248	110/243	110/431	123/423	95/213	89/207	126/484	189/526
2	10000	Failed	Failed	578/2294	440/1585	Failed	473/824	494/1922	425/1248
3	10000	70/99	86/98	78/171	85/112	74/100	77/78	68/128	90/119
4	10000	167/230	313/444	86/130	116/189	94/110	101/118	94/140	134/220
5	100	129/220	107/161	137/230	116/188	105/161	107/149	134/244	119/177
6	2	71/194	79/178	87/385	168/806	65/172	73/180	108/482	132/374
7	4	374/813	200/386	284/881	170/431	300/561	215/323	422/1325	197/468
8	2	9/14	9/14	7/12	7/12	8/13	8/13	7/12	7/12
9	3	17/24	18/25	23/31	23/30	17/24	17/24	20/28	20/27
10	2	26/42	23/38	29/59	35/82	34/54	26/40	29/55	45/96
11	2	62/155	41/84	90/310	57/185	62/133	50/103	64/213	51/164
Sum I		$*$	$*$	1509	1340	$*$	1186	1502	1358
Sum F		$*$	$*$	4934	4043	$*$	1956	4820	3267

r		0	0	1	1	0	0	1	1
ϕ_1		0.25	0.75	0.25	0.75	0.25	0.75	0.25	0.75
ϕ_2		0.75	0.25	0.75	0.25	0.75	0.25	0.75	0.25
M_c		3	3	3	3	3	3	3	3
m_1		1	1	1	1	3	3	3	3
m_2		2	2	2	2	4	4	4	4
P	n								
1	10000	118/255	116/241	125/264	92/215	125/434	80/288	179/454	161/447
2	10000	Failed	629/1423	533/849	531/915	392/1495	548/2114	512/1430	362/1098
3	10000	71/93	78/95	83/84	78/80	69/126	83/157	83/116	85/112
4	10000	95/107	Failed	110/131	132/168	147/259	95/138	105/171	143/262
5	100	116/151	126/187	93/119	109/158	122/217	128/215	119/180	128/185
6	2	128/313	80/204	59/147	72/186	134/585	74/276	138/414	80/259
7	4	438/811	425/791	305/443	212/349	293/838	389/1068	151/338	166/401
8	2	9/14	8/13	9/14	8/13	7/12	7/12	7/12	7/12
9	3	24/31	20/27	17/24	18/25	20/28	21/29	20/27	21/28
10	2	30/45	38/53	26/41	23/38	35/90	33/81	35/77	35/77
11	2	92/207	74/141	48/98	53/98	68/229	77/270	51/166	53/159
Sum I		$*$	$*$	1408	1328	1412	1535	1400	1241
Sum F		$*$	$*$	2214	2245	4313	4648	3385	3040

Table 7: Numerical results of cyclic NEBB with $M_c = 5$

r		0	1	0	1	0	1	0	1
ϕ_1		1	1	1	1	0.5	0.5	0.5	0.5
ϕ_2		0	0	0	0	0.5	0.5	0.5	0.5
M_c		5	5	5	5	5	5	5	5
m_1		1	1	3	3	1	1	3	3
m_2		–	–	–	–	2	2	4	4
P	n								
1	10000	182/663	132/426	217/675	87/270	167/592	132/402	132/397	87/251
2	10000	402/1402	342/1159	Failed	617/2053	427/1295	462/1371	Failed	452/1239
3	10000	68/131	87/107	72/150	112/157	68/154	87/108	87/193	97/116
4	10000	203/371	118/170	267/687	142/255	188/322	103/141	162/320	147/222
5	100	117/204	107/166	138/248	129/209	132/225	93/121	143/236	124/200
6	2	87/321	119/432	107/429	112/391	82/286	87/337	222/820	107/381
7	4	272/688	217/470	532/1793	177/385	272/661	327/651	327/1005	172/391
8	2	8/13	8/13	11/16	11/16	7/12	7/12	12/17	12/17
9	3	22/29	17/24	24/32	22/29	17/24	18/25	22/30	27/34
10	2	37/65	27/54	32/74	32/73	42/76	32/74	32/74	27/54
11	2	53/171	48/149	57/158	63/193	77/239	52/154	48/147	47/116
Sum I		1451	1222	$*$	1504	1479	1400	$*$	1299
Sum F		4058	3170	$*$	4031	3886	3396	$*$	3021

r		0	0	1	1	0	0	1	1
ϕ_1		0.25	0.75	0.25	0.75	0.25	0.75	0.25	0.75
ϕ_2		0.75	0.25	0.75	0.25	0.75	0.25	0.75	0.25
M_c		5	5	5	5	5	5	5	5
m_1		1	1	1	1	3	3	3	3
m_2		2	2	2	2	4	4	4	4
P	n								
1	10000	192/695	127/439	122/384	133/425	163/507	148/481	82/211	87/255
2	10000	627/1768	402/1183	462/1281	372/1054	Failed	Failed	577/1430	552/1532
3	10000	67/139	62/113	83/105	77/103	86/141	72/150	101/122	100/129
4	10000	192/302	212/381	112/142	107/141	109/183	95/154	120/177	382/743
5	100	127/215	127/181	97/135	110/160	147/256	113/178	132/194	123/189
6	2	79/342	87/302	147/465	92/401	182/725	202/740	98/365	152/498
7	4	277/703	447/1189	232/491	183/380	832/2409	448/1294	157/331	192/423
8	2	8/13	8/13	8/13	8/13	12/17	12/17	12/17	12/17
9	3	17/24	17/24	22/29	17/24	22/30	22/30	20/27	23/30
10	2	42/70	37/64	37/73	27/54	32/74	32/74	27/54	27/54
11	2	119/294	83/232	72/201	68/185	48/147	42/116	77/217	83/228
Sum I		1747	1609	1394	1194	$*$	$*$	1403	1733
Sum F		4565	4121	3319	2940	$*$	$*$	3145	4098

Table 8: Numerical results of typical CG methods

P	n	FR	HS	PR+	DY
1	10000	170 / 2001	43 / 350	69 / 622	43 / 372
2	10000	595 / 4627	173 / 1082	307 / 2207	634 / 4467
3	10000	403 / 1912	70 / 73	70 / 75	125 / 431
6	2	127 / 1501	29 / 238	95 / 939	46 / 454
7	4	301 / 3475	208 / 1969	197 / 1915	Failed
8	2	9 / 24	7 / 43	6 / 23	11 / 28
9	3	26 / 274	45 / 371	33 / 271	85 / 1360
10	2	41 / 305	15 / 95	31 / 229	31 / 213
11	2	48 / 470	81 / 699	140 / 1380	57 / 511

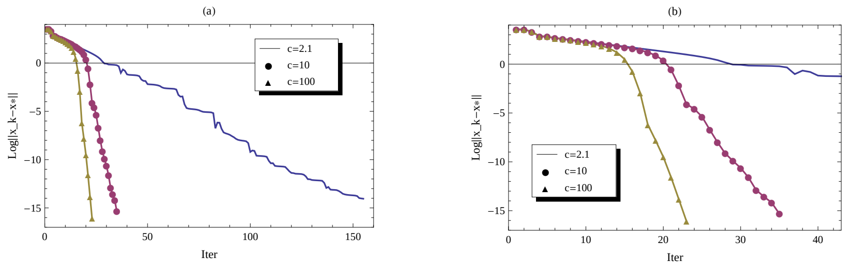

Figure 1: Behavior of $\log_{10} \|x_k - x^*\|$ for $A = A_1$

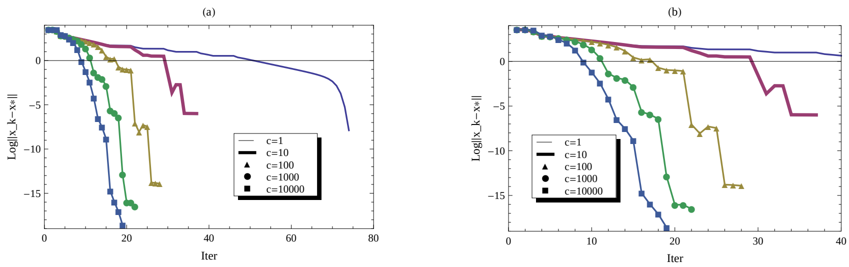

Figure 2: Behavior of $\log_{10} \|x_k - x^*\|$ for $A = A_2$

References

[1] J. Barzilai and J.M. Borwein, Two-point step size gradient methods, *IMA Journal of Numerical Analysis* 8 (1988) 141–148.

[2] Y.H. Dai, A nonmonotone conjugate gradient algorithm for unconstrained optimization, *Journal of System Science and Complexity* 15 (2002) 139-145.

[3] Y.H. Dai, On the nonmonotone line search, *Journal of Optimization Theory and Applications* 112 (2002) 315-330.

[4] Y.H. Dai, W.W. Hager, K. Schittkowski and H. Zhang, The cyclic Barzilai-Borwein method for unconstrained optimization, *IMA Journal of Numerical Analysis* 26 (2006) 604-627.

[5] Y.H. Dai and L.Z. Liao, R-linear convergence of the Barzilai and Borwein gradient method, *IMA Journal of Numerical Analysis* 22 (2002) 1–10.

[6] Y.H. Dai and Y. Yuan, Analysis of monotone gradient methods, *Journal of Industrial and Management Optimization* 1 (2005) 181–192.

[7] J.E. Dennis and J.J. Moré, A characterization of superlinear convergence and its application to quasi-Newton methods, *Mathematics of Computation* 28 (1974) 549–560.

[8] R. Fletcher, On the Barzilai-Borwein method, Numerical Analysis Report NA/207, Department of Mathematics, University of Dundee, 2001 (See also Optimization and Control with Applications, Springer series in Applied Optimization 96, Springer-Verlag, 2005, pp. 235–256.

[9] A. Friedlander, J.M. Martinez, B. Molina and M. Raydan, Gradient method with retards and generalizations, *SIAM Journal on Numerical Analysis* 36 (1999) 275–289.

[10] L. Grippo, F. Lampariello and S. Lucidi, A nonmonotone line search technique for Newton's method, *SIAM Journal on Numerical Analysis* 23 (1986) 707–716.

[11] L. Grippo, F. Lampariello and S. Lucidi, A truncated Newton method with nonmonotone line search for unconstrained optimization, *Journal of Optimization Theory and Applications* 60 (1989) 401–419.

[12] L. Grippo and M. Sciandrone, Nonmonotone globalization techniques for the Barzilai-Borwein gradient method, *Computational Optimization and Applications*, 23 (2002) 143–169.

[13] Matrix Market, http://math.nist.gov/MatrixMarket/

[14] J.J. Moré, B.S. Garbow and K.E. Hillstrom, Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* 7 (1981) 17–41.

[15] J. Nocedal and S.J. Wright, Numerical Optimization, second edition, Springer Series in Operations Research, Springer Verlag, New York, 2006.

[16] M. Raydan, On the Barzilai and Borwein choice of steplength for the gradient method, *IMA Journal of Numerical Analysis* 13 (1993) 321–326.

[17] M. Raydan, The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem, *SIAM Journal on Optimization* 7 (1997) 26–33.

[18] M. Raydan and B. Svaiter, Relaxed steepest descent and Cauchy-Barzilai-Borwein method, *Computational Optimization and Applications* 21 (2002) 155–167.

[19] Y.X. Yuan, A new stepsize for the steepest descent method, *Journal of Computational Mathematics* 24 (2006) 149–156.

[20] Y.X. Yuan, Gradient methods for large scale convex quadratic functions, Technical Report (see also Y.F. Wang, A. Yagola and C. Yang eds., Optimization and Regularization for Computational Inverse Problems and Applications, (Higher Education Press, 2009)).

[21] B. Zhou, L. Gao and Y.H. Dai, Gradient methods with adaptive step-size, *Computational Optimization and Applications* 35 (2006) 69–86.

Manuscript received 7 October 2008
revised 7 January 2010
accepted for publication 12 January 2010

Yasushi Narushima
Tokyo University of Science, Japan
E-mail address: narusima@rs.kagu.tus.ac.jp

Takahiko Wakamatsu
Hitach, Ltd., Software Division
E-mail address: takahiko.wakamatsu.ya@hitachi.com

Hiroshi Yabe
Tokyo University of Science, Japan
E-mail address: yabe@rs.kagu.tus.ac.jp