Optim Lett
DOI 10.1007/s11590-017-1150-9

ORIGINAL PAPER

A new adaptive Barzilai and Borwein method for unconstrained optimization

Hongwei Liu[1] · **Zexian Liu**[1,2] · **Xiaoliang Dong**[3]

Received: 9 June 2016 / Accepted: 28 April 2017

Abstract In this paper we view the Barzilai and Borwein (BB) method from a new angle, and present a new adaptive Barzilai and Borwein (NABB) method with a non-monotone line search for general unconstrained optimization. In the proposed method, the scalar approximation to the Hessian matrix is updated by the Broyden class formula to generate an adaptive stepsize. It is remarkable that the new stepsize is chosen adaptively in the interval which contains the two well-known BB stepsizes. Moreover, for the negative curvature direction, a strategy for the choice of the stepsize is designed to accelerate the convergence rate of the NABB method. Furthermore, we apply the NABB method without any line search to strictly convex quadratic minimization. The numerical experiments show the NABB method is very promising.

Keywords Barzilai and Borwein (BB) method · Adaptive stepsize · Nonmonotone line search · Broyden class update formula · Strictly convex quadratic minimization

✉ Zexian Liu
liuzexian2008@stu.xidian.edu.cn

Hongwei Liu
hwliu@mail.xidian.edu.cn

Xiaoliang Dong
dongxl@nun.edu.cn

1 School of Mathematics and Statistics, Xidian University, Xi'an 710126, People's Republic of China

2 School of Mathematics and Computer Science, Hezhou University, Hezhou 542899, People's Republic of China

3 School of Mathematics and Information, Beifang University of Nationalities, Yinchuan 710021, People's Republic of China

Published online: 22 May 2017

1 Introduction

The Barzilai and Borwein (BB) method [1], proposed by Barzilai and Borwein in 1988, has received extensive concerns due to its simplicity and numerical efficiency. The search direction of the BB method is always the negative gradient direction, but the choice of the stepsize is different from that of the steepest descent (SD) method [2]. It is generally accepted that the SD method, which is the simplest iterative method, is badly affected by ill conditioning and thus converges very slowly.

The BB method has exhibited surprising numerical behaviour and thus has enjoyed great development during the past decades. Most of these works focused on explaining the surprising behaviour of the BB method and investigating efficient BB-like methods [3]. Raydan [4] proved that the BB method was globally convergent for any dimension strictly convex quadratic functions, and Dai and Liao [5] established the R-linear convergence of the BB method. By incorporating the nonmonotone line search (GLL line search) introduced by Grippo, Lampariello and Lucidi [6], Raydan [7] presented the BB method for large scale unconstrained optimization problems. The numerical results of [7] showed that the BB method outperformed several famous conjugate gradient methods for large-scale optimization problems. Dai et al. [8] present the cyclic BB method and analyzed its important convergent properties. In recent years it has been made more and more clear that the convergence rate of the BB method depends on the spectrum of the Hessian matrix to great extend. Raydan et al. [9] observed that the potential effectiveness of the BB method was related to the relationship between the stepsize and the eigenvalues of the Hessian rather than to the decrease of the function value by numerical experiment. Dai [10] showed that for two dimensional strictly convex quadratic functions there was a superlinear convergence step in at most three consecutive steps, and discovered that the convergence rate of the BB method was related to both the starting point and the condition number.

However, it still remains to study how to design more robust and more efficient gradient methods for unconstrained optimization [26]. For large scale problems, more efficient stepsizes for gradient methods remain to be designed by using function values, gradients and the stepsizes at the previous iterates.

In this paper we present an new adaptive BB method with a nonmonotone line search, in which the scalar approximation of the Hessian matrix is updated by the so-called Broyden class formula to generate a new stepsize. For the curvature negative direction, a strategy for stepsize is developed to speed up the convergence rate of the proposed BB-like method. Furthermore, we apply the NABB method without any line search to strictly convex quadratic minimization.

The paper is organized as follows. Some preliminaries are made in Sect. 2. In Sect. 3 we view the BB method from a new angle, update the approximation to the Hessian matrix by the so-called Broyden class formula and derive a new adaptive stepsize. Moreover, we design a strategy for the stepsize to speed up the convergence rate of the proposed method. In Sect. 4 we present a new adaptive BB method with a nonmonotone line search and prove the global convergence of the proposed method under mild conditions. In Sect. 5 we apply the NABB method without any line search to strictly convex quadratic minimization. In Sect. 6 we illustrate the numerical results which indicate that the NABB method is very efficient and promising. Conclusions and discussions are made in the last section.

2 Preliminaries

Consider the following unconstrained optimization problem

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

where $f : R^n \to R$ is continuously differentiable.

Notation. Throughout this paper, $g(x) = \nabla f(x)$, $f_k = f(x_k)$ and $g_k = g(x_k)$ and $\|\cdot\|$ denotes the Euclidean norm.

The well-known BB method [1] for solving (2.1) takes the form

$$x_{k+1} = x_k - \alpha_k g_k, \tag{2.2}$$

where α_k is the BB stepsize given by

$$\alpha_k^{BB_1} = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}} \quad \text{or} \quad \alpha_k^{BB_2} = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}, \tag{2.3}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$.

During these years many efficient stepsizes based on the modified secant equations or some conic models have been developed for BB-like methods. For example, by constructing a quadratic model and a conic model and imposing some interpolation conditions, Dai et al. [13] interpreted the stepsize $\alpha_k^{BB_1}$ from the angle of interpolation and derived two new stepsizes

$$\alpha_k = \frac{s_{k-1}^T s_{k-1}}{2(f_{k-1} - f_k + g_k^T s_{k-1})} \tag{2.4}$$

and

$$\alpha_k = \frac{s_{k-1}^T s_{k-1}}{6(f_{k-1} - f_k) + 4g_k^T s_{k-1} + 2g_{k-1}^T s_{k-1}} \tag{2.5}$$

for BB-like methods. According to the modified secant equations of [14,15], Xiao et al. [16] also derived the stepsizes (2.4) and (2.5) and present four BB-like methods. Based on a fourth order conic model and the modified quasi Newton equation [17], Biglari and Solimanpur [12] developed some new stepsizes for BB-like methods, for example,

$$\alpha_k^{SBB_4} = \frac{s_{k-1}^T \overline{y}_{k-1}}{\overline{y}_{k-1}^T \overline{y}_{k-1}}, \tag{2.6}$$

where $\overline{y}_{k-1} = y_{k-1} + \frac{4(f_{k-1}-f_k)+2(g_k+g_{k-1})^T s_{k-1}}{s_{k-1}^T y_{k-1}} y_{k-1}$, and the numerical results of [12] indicated these BB-like methods were very efficient.

Zhou et al. [11] proposed an adaptive BB (ABB) method in which the stepsize is given by

$$\alpha_k^{ABB} = \begin{cases} \alpha_k^{BB_2}, & \text{if } \ \alpha_k^{BB_2}/\alpha_k^{BB_1} < \varsigma, \\ \alpha_k^{BB_1}, & \text{otherwise}, \end{cases} \tag{2.7}$$

where $\varsigma \in (0, 1)$.

It seems that if the stepsize is chosen dynamically in a interval which contains the stepsizes $\alpha_k^{BB_1}$ and $\alpha_k^{BB_2}$, the BB-like methods will exhibit good numerical performance. Therefore, a natural questions to ask is that: Can one develop a stepsize, which is picked adaptively in $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$, for BB-like method by making use of function value and gradient values at the previous iterates? The purpose of our work is to answer this question.

3 Derivation of new adaptive stepsize and its properties

In this section we develop an adaptive stepsize and study its important properties.

In the BB method, the stepsize $\alpha_k^{BB_1}$ is determined by solving the following problem:

$$\min_{\alpha} ||D_k s_{k-1} - y_{k-1}||_2^2,$$

where $D_k = \dfrac{1}{\alpha} I$ is regarded as an approximation to the Hessian matrix. It is clear that

$$D_k = \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2} I$$

satisfies the weak secant equation $s_{k-1}^T D_k s_{k-1} = s_{k-1}^T y_{k-1}$, which was firstly introduced by Dennis and Wolkowicz [18]. In general, the standard secant equation

$$B_k s_{k-1} = y_{k-1} \tag{3.1}$$

is not true for D_k, while the standard secant equation (3.1) plays an important role in the approximation to the Hessian matrix. Generally speaking, the scalar matrix can not approximate the Hessian matrix very well. Therefore, to allow D_k to satisfy the secant equation (3.1), we update D_k by the so-called Broyden class update formula [19]

$$B_k = D_k - \frac{D_k s_{k-1} s_{k-1}^T D_k}{s_{k-1}^T D_k s_{k-1}} + \frac{y_{k-1} y_{k-1}^T}{s_{k-1}^T y_{k-1}} + t w_{k-1} w_{k-1}^T, \tag{3.2}$$

where t is a scalar parameter and $w_{k-1} = (s_{k-1} D_k s_{k-1})^{\frac{1}{2}} \left(\frac{y_{k-1}}{s_{k-1}^T y_{k-1}} - \frac{D_k s_{k-1}}{s_{k-1}^T D_k s_{k-1}} \right)$. It is easy to see that (3.1) is true for B_k and B_k is symmetric. Let θ_k be the angle between

s_{k-1} and y_{k-1}. According to (1.2.40) of [19], we can get the maximal and minimal eigenvalues of B_k:

$$\lambda_{\max/\min}(t) = \frac{\|y_{k-1}\|}{\|s_{k-1}\|}\left(\cos\theta_k + \frac{t+1}{2}\frac{\sin^2\theta_k}{\cos\theta_k} \pm \sqrt{(t+1)^2\frac{\sin^4\theta_k}{4\cos^2\theta_k} + \sin^2\theta_k}\right).$$

Clearly, $\lambda_{\min}(-\tan^2\theta_k) = 0$ and $\lambda_{\min}(t)$ is monotonically increasing in the interval $(-\tan^2\theta_k, \infty)$.

For the choice of t in (3.2), we determine it based on the following observations.

On the one hand, similar to most quasi-Newton methods, B_k should be determined to ensure that the difference $\|B_k - D_k\|_F^2$ is as small as possible, that is,

$$\min_{t>-\tan^2\theta_k} \quad p_1(t) = \|B_k - D_k\|_F^2.$$

It is not difficult to see that, if $-\tan^2\theta_k < -1$, $p_1(t)$, as a quadratic function, attains its minimum at $t = -1$ and is monotonically increasing in the interval $[-1, \infty)$, otherwise $p_1(t)$ is monotonically increasing in the interval $(-\tan^2\theta_k, \infty)$.

On the other hand, it is well-known that matrix condition number plays an important role in the error analysis of a numerical problem related to a matrix in the sense of measuring sensitivity of the solution to data perturbations [20]. A matrix with a large condition number is called an ill conditioned matrix. In order to enhance numerical stability in the proposed method, it is reasonable to determine t by minimizing the condition number $k(B_k)$. In practice, we solve its equivalent problem

$$\min_{t>-\tan^2\theta_k} \quad p_2(t) = \ln\frac{\lambda_{\max}^2(t)}{\lambda_{\max}(t)\lambda_{\min}(t)}$$

instead of minimizing $k(B_k)$. It is not difficult to see that $p_2(t)$ attains its minimum at $t = 1$.

Denote $\tau_k = \max(-1, -\tan^2\theta_k)$. We know that $p_1(t)$ is monotonically increasing in the interval (τ_k, ∞) and approaches to its infimum $\inf\{p_1(t), t \in (\tau_k, \infty)\}$ as t tends to τ_k from the right side, while $p_2(t)$ is monotonically decreasing in the interval $(-\tan^2\theta_k, 1]$ and is monotonically increasing in the interval $[1, \infty)$. Therefore, t should be close to τ_k by the monotonicity of $p_1(t)$, while t should be close to 1 by the monotonicity of $p_2(t)$. As a result the choice of $t = 0$ is a tradeoff between the above two observations. It is also observed by numerical experiments that $t = 0$ is indeed a good choice for B_k. So the choice of $t = 0$ is used throughout this paper.

Since $0 > -\tan^2\theta_k$, we can easily obtain the following lemma.

Lemma 3.1 *Suppose that $s_{k-1}^T y_{k-1} > 0$. Then B_k is symmetric positive definite.*

We view the choice of the stepsize $\alpha_k^{BB_1}$ from a new angle. The stepsize $\alpha_k^{BB_1}$ can be also given by solving the following problem

$$\min_{\alpha} \quad \Phi(\alpha) = f(x_k) - \alpha g_k^T g_k + \frac{1}{2}\alpha^2 g_k^T D_k g_k.$$

More specifically, by imposing $\frac{d\Phi}{d\alpha} = -g_k^T g_k + \alpha {g_k}^T D_k g_k = 0$, we also get the stepsize $\alpha_k^{BB_1}$. Replacing D_k in $\Phi(\alpha)$ by B_k gives that

$$\min_{\alpha} \quad \Phi^*(\alpha) = f(x_k) - \alpha g_k^T g_k + \frac{1}{2}\alpha^2 g_k^T B_k g_k.$$

Similarly, we obtain that

$$\widetilde{\alpha}_k = \frac{1}{\frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2}\sin^2\beta_k + \frac{\|y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}\cos^2\omega_k}. \tag{3.3}$$

where β_k denotes the angel between g_k and s_{k-1}, and ω_k denotes the angel between g_k and y_{k-1}.

The new stepsize (3.3) can be rearranged as

$$\widetilde{\alpha}_k = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}} \frac{\cos^2\theta_k}{\sin^2\beta_k \cos^2\theta_k + \cos^2\omega_k}. \tag{3.4}$$

Obviously, if $\sin\beta_k = 1$ and $\cos\omega_k = 0$, then the new stepsize (3.3) reduces to $\alpha_k^{BB_1}$; if $\sin\beta_k = 0$ and $\cos\omega_k = 1$, then the new stepsize (3.3) reduces to $\alpha_k^{BB_2}$. It is clear that the new stepsize (3.4) is indeed a scaled version of $\alpha_k^{BB_1}$.

Lemma 3.2 *For the stepsize (3.3), we obtain that* $\widetilde{\alpha}_k \geq \frac{3}{4}\frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}$.

Proof The stepsize (3.3) can also be rearranged as

$$\widetilde{\alpha}_k = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2} \frac{1}{\sin^2\beta_k \cos^2\theta_k + \cos^2\omega_k}.$$

It is clear from the definitions of β_k, ω_k and θ_k and the search direction $-g_k$ that $\omega_k = \beta_k - \theta_k$. Denote $\cos c_k = \frac{\sin 2\theta_k}{\sqrt{\sin^2 2\theta_k + \sin^4\theta_k}}$. Then we have that $\sin c_k = \frac{\sin^2\theta_k}{\sqrt{\sin^2 2\theta_k + \sin^4\theta_k}}$. By the properties of the trigonometric functions we obtain that

$$\begin{aligned}
\sin^2\beta_k \cos^2\theta_k + \cos^2\omega_k &= \cos^2\theta_k \sin^2\beta_k + \cos^2(\beta_k - \theta_k) \\
&= \cos^2\theta_k \sin^2\beta_k + (\cos\beta_k \cos\theta_k + \sin\beta_k \sin\theta_k)^2 \\
&= \cos^2\theta_k \sin^2\beta_k + \cos^2\beta_k \cos^2\theta_k + \sin^2\beta_k \sin^2\theta_k \\
&\quad\ +2\sin\beta_k \sin\theta_k \cos\beta_k \cos\theta_k \\
&= \cos^2\theta_k + \frac{1-\cos 2\beta_k}{2}\sin^2\theta_k + \frac{1}{2}\sin 2\theta_k \sin 2\beta_k
\end{aligned}$$

$$
\begin{aligned}
&= \cos^2\theta_k + \frac{1}{2}\sin^2\theta_k + \frac{1}{2}\sqrt{\sin^2 2\theta_k + \sin^4\theta_k} \\
&\quad \left(\frac{\sin 2\theta_k \sin 2\beta_k}{\sqrt{\sin^2 2\theta_k + \sin^4\theta_k}} - \frac{\sin^2\theta_k \cos 2\beta_k}{\sqrt{\sin^2 2\theta_k + \sin^4\theta_k}} \right) \\
&= \cos^2\theta_k + \frac{1}{2}\sin^2\theta_k + \frac{1}{2}\sqrt{\sin^2 2\theta_k + \sin^4\theta_k}\,\sin(2\beta_k - c_k) \\
&\le 1 - \frac{1}{2}\sin^2\theta_k + \frac{1}{2}\sqrt{\sin^2 2\theta_k + \sin^4\theta_k} \\
&= 1 - \frac{1}{2}\sin^2\theta_k + \sqrt{\sin^2\theta_k - \frac{3}{4}\sin^4\theta_k}.
\end{aligned}
$$

Let us define $u = \sin^2\theta_k \in [0, 1]$ and $h(u) = 1 - \frac{1}{2}u + \sqrt{u - \frac{3}{4}u^2}$. Imposing $\frac{dh}{du} = -\frac{1}{2} + \frac{1 - \frac{3}{2}u}{2\sqrt{u - \frac{3}{4}u^2}} = 0$ we know $u = \frac{1}{3}$ is its root. Substituting $u = \frac{1}{3}$ into

$$
\frac{d^2h}{du^2} = -\frac{3}{4\sqrt{u - \frac{3}{4}u^2}} - \frac{\frac{1}{2} - \frac{3}{2}u + \frac{9}{8}u^2}{2\sqrt{\left(u - \frac{3}{4}u^2\right)^3}}
$$

yields that $\frac{d^2h}{du^2}|_{u=1/3} = -2$, which means that $h(u) \le h(1/3)$ holds for $u \in [0, 1]$. Therefore, we obtain that

$$
\begin{aligned}
\sin^2\beta_k \cos^2\theta_k + \cos^2\omega_k &\le 1 - \frac{1}{2}\sin^2\theta_k + \sqrt{\sin^2\theta_k - \frac{3}{4}\sin^4\theta_k} \\
&\le 1 - \frac{1}{2} \times \frac{1}{3} + \sqrt{\frac{1}{3} - \frac{3}{4} \times \frac{1}{3} \times \frac{1}{3}} = \frac{4}{3}
\end{aligned}
$$

and

$$
\widetilde{\alpha}_k = \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2} \frac{1}{\sin^2\beta_k \cos^2\theta_k + \cos^2\omega_k} \ge \frac{3}{4} \frac{s_{k-1}^T y_{k-1}}{\|y_{k-1}\|^2}.
$$

The proof is completed. □

We see from (3.3) that if $\cos\omega_k = 0$ and $\sin^2\beta_k < 1$, the new stepsize (3.3) is greater than $\alpha_k^{BB_1}$, which together with Lemma 3.2 implies that the new stepsize (3.3) is picked dynamically a interval which includes $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$. Lemma 3.2 gives a lower bound of the new stepsize (3.3), which indicates that the minimum of the stepsize (3.3) is not much less than the stepsize $\alpha_k^{BB_2}$. It is observed by numerical experiments that the bound $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$ for the new stepsize (3.3) is very preferable. Therefore, together with the success of the stepsizes $\alpha_k^{BB_1}$ and $\alpha_k^{BB_2}$, in practice we take the truncated form of the new stepsize (3.3):

$$\widetilde{\alpha}_k = \min\left\{\alpha_k^{BB_1}, \max\left\{\alpha_k^{BB_2}, \frac{1}{\frac{s_{k-1}^T y_{k-1}}{\| s_{k-1}\|^2}\sin^2\beta_k + \frac{\| y_{k-1}\|^2}{s_{k-1}^T y_{k-1}}\cos^2\omega_k}\right\}\right\}, \tag{3.5}$$

which implies the new stepsize (3.5) is chosen adaptively in the interval $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$. Since the stepsize (3.5) includes more information and is chosen adaptively in the interval $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$, it is reasonable to expect that (3.5) is better than $\alpha_k^{BB_1}$.

When the search direction is the negative curvature direction, that is , $s_{k-1}^T y_{k-1} \le 0$, in most BB-like methods the stepsize is set simply to $\alpha_k = \lambda_{\max}$, where $\lambda_{\max}$ is a pre-fixed large positive constant. It is too simple to result in expensive computational cost for BB-like methods for solving large scale unconstrained optimization to seek a suitable stepsize satisfying some nonmonotone or monotone conditions. It is well-known that for a quadratic function the stepsize $\alpha_k^{BB_1}$ is exactly equal to α_{k-1}^{SD}. Moreover, it also has been shown that if the retard technique is used in BB-like methods, the convergence rates are accelerated [21]. It seems that the stepsizes at the previous iterates may provide some important information for the current stepsize. Based on the above considerations, when $s_{k-1}^T y_{k-1} \le 0$, the stepsize is set to

$$\alpha_k = \delta\alpha_{k-1}, \tag{3.6}$$

where δ is a positive parameter.

4 The NABB Method with Zhang–Hager line search for general unconstrained optimization

In this section, we present a new BB-like method with a nonmonotone line search for general unconstrained optimization. As mentioned above, the stepsize (3.5) is picked adaptively in the interval $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$, so we call the proposed method new adaptive BB method (NABB) in this sense. The popular nonmonotone line search (GLL line search) [6] was embedded into the BB method by Raydan [7]. Although GLL line search permits some growth in the function value as the iteration process and works well in many cases, there are some drawbacks [22], for example, some good function values may be discarded, or the numerical performance depends very much on the choice of a pre-fixed memory constant. To address these drawbacks, Zhang and Hager [22] proposed a new nonmonotone line search (Zhang–Hager line search) which requires that an average of the successive function values decreases. It is also observed that for BB-like methods Zhang–Hager line search is preferable. Consequently, Zhang–Hager line search is adopted in the NABB method. The technique of [23] for the choice of a trial stepsize in a line search is introduced to accelerate the convergence rate of the NABB method. Now we describe the NABB method with Zhang–Hager line search for unconstrained optimization in detail.

New Adaptive BB method (NABB)

Step 0 Initiation: Given $x_0 \in R^n$, $\sigma \in (0, 1)$, $\lambda_{\min}, \lambda_{\max}, \varepsilon,\ \eta_{\min},\ \eta_{\max},\ \alpha_0^0,\ \eta_0 \in [\eta_{\min}, \eta_{\max}]$, $C_0 = f(x_0),\ \ Q_0 = 1$. Set $k := 0$.

Step 1 If $\| g_k \|_\infty \le \varepsilon$, stop.

Step 2 Compute α. If $k = 0$, set $\alpha = \alpha_0^0$ and go to Step 3. If $s_{k-1}^T y_{k-1} > 0$, compute $\widetilde{\alpha}_k$ by (3.5) and set $\alpha_k^0 = \min\{\max\{\widetilde{\alpha}_k, \lambda_{\min}\}, \lambda_{\max}\}$; otherwise, compute α_k by (3.6) and set $\alpha_k^0 = \min\{\max\{\alpha_k, \lambda_{\min}\}, \lambda_{\max}\}$. Set $\alpha = \alpha_k^0$.

Step 3 Zhang–Hager line search. If

$$f(x_k - \alpha g_k) \le C_k - \sigma\alpha \parallel g_k \|^2, \tag{4.1}$$

then go to Step 4; otherwise, update α by

$$\alpha = \begin{cases} \overline{\alpha}, & \text{if } \alpha > 0.1\alpha_k^0 \text{ and } \overline{\alpha} \in [0.1\alpha_k^0, 0.9\alpha], \\ 0.5\alpha, & \text{otherwise}, \end{cases} \tag{4.2}$$

where $\overline{\alpha}$ is the trial stepsize obtained by a quadratic interpolation at x_k and $x_k - \alpha g_k$ [23], go to Step 3.

Step 4 Choose $\eta_k \in [\eta_{\min}, \eta_{\max}]$ and update Q_{k+1}, C_{k+1} by

$$Q_{k+1} = \eta_k Q_k + 1, C_{k+1} = (\eta_k Q_k C_k + f(x_{k+1}))/Q_{k+1}. \tag{4.3}$$

Step 5 Set $\alpha_k = \alpha$, $x_{k+1} = x_k - \alpha_k g_k$ and $k := k + 1$, go to Step 1.

We turn to consider the convergence of the NABB method with Zhang–Hager line search for unconstrained optimization. Our convergence result utilizes the following assumptions.

A1. $f(x)$ is continuously differentiable on R^n.
A2. $f(x)$ is bounded below on R^n.
A3. The gradient $g(x)$ is Lipschitz continuous on R^n, namely, there exists $L > 0$ such that

$$\| g(x) - g(y) \| \le L \parallel x - y \|, \quad \forall x, y \in R^n.$$

Since $d_k = -g_k$, we have $\|d_k\| = \|g_k\|$ and $g_k^T d_k = -\|g_k\|^2$. Therefore we can directly obtain the following lemma according to Lemma 1.1 of [22].

Lemma 4.1 *Let* $A_k = \dfrac{1}{k+1}\sum_{i=0}^{k} f_i$. *Then the iterates generated by the NABB method satisfy*

$$f_k \le C_k \le A_k.$$

Furthermore, suppose that A1 and A2 hold. There exists α_k satisfying the nonmonotone condition (4.1).

Lemma 4.2 *Suppose that A1 and A3 hold. Then there exists $\xi > 0$ such that*

$$\alpha_k \geq \xi.$$

Proof If α_k^0 satisfies the condition (4.1), then $\alpha_k = \alpha_k^0 \geq \lambda_{\min}$. Otherwise, if $\alpha_k \leq 0.05\alpha_k^0$, by the mechanism of the NABB method and Lemma 4.1, for α_k we have that

$$f(x_k - 2\alpha_k g_k) > C_k - 2\sigma\alpha_k g_k^T g_k \geq f(x_k) - 2\sigma\alpha_k g_k^T g_k.$$

By the mean-value theorem and the Lipschitz continuous of the gradient $g(x)$, we know there exists $v_k \in (0, 1)$ such that

$$-2\alpha_k g(x_k - 2\alpha_k v_k g_k)^T g_k \geq -2\sigma\alpha_k g_k^T g_k,$$

and thus get that

$$2L\alpha_k \parallel g_k \parallel_2^2 \geq 2v_k L\alpha_k \parallel g_k \parallel_2^2 \geq -g(x_k - 2v_k\alpha_k g_k)^T g_k + g_k^T g_k \geq (1-\sigma) \parallel g_k \parallel_2^2 .$$

Therefore,

$$\alpha_k \geq \frac{(1-\sigma)}{2L}.$$

If the inequality $\alpha_k \leq 0.05\alpha_k^0$ does not hold, then $\alpha_k > 0.05\alpha_k^0 \geq 0.05\lambda_{\min}$. Let $\xi = \min\{\frac{(1-\sigma)}{2L}, 0.05\lambda_{\min}\}$. In conclusion, we have that

$$\alpha_k \geq \xi.$$

The proof is completed. □

Since $d_k = -g_k$ and $\alpha_k \leq \lambda_{\max}$, we can obtain directly the following two theorems according to Theorem 2.2 and Theorem 3.1 of [22], Lemmas 4.1 and 4.2.

The following two theorems show that the NABB method is globally convergent and is R-linearly convergent.

Theorem 4.1 *Suppose that A1, A2 and A3 hold. Let $\{x_k\}$ be the sequence generated by the NABB method. Then*

$$\lim_{k\to\infty} \inf \parallel g(x_k) \parallel = 0.$$

Furthermore, if $\eta_{\max} < 1$, then

$$\lim_{k\to\infty} \parallel g(x_k) \parallel = 0.$$

Hence, every convergent subsequence of the $\{x_k\}$ approaches a stationary point x^*.

Theorem 4.2 *Suppose that A1, A2 and A3 hold. f is strongly convex with unique minimizer x^* and $\eta_{\max} < 1$. Then there exists $\zeta \in (0, 1)$ such that*

$$f(x_k) - f(x^*) \leq \varsigma^k \left(f(x_0) - f(x^*)\right),$$

for each $k \geq 0$.

5 The NABB method without any line search for strictly convex quadratic minimization

In this section we prove the NABB method without any line search is convergent for strictly convex quadratic minimization

$$\min_{x \in R^n} f(x) = \frac{1}{2} x^T A x - b^T x, \tag{5.1}$$

where $b \in R^n$ and $A \in R^{n \times n}$ is a symmetric positive definite matrix.

Assume A has the eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ with the associated orthonormal eigenvectors $\{v_1, v_2, \ldots, v_n\}$. We investigate the relation of the stepsize (3.3) and the BB stepsizes $\alpha_k^{BB_1}$ and $\alpha_k^{BB_2}$ in the following lemma.

Lemma 5.1 *For strictly convex quadratic minimization* (5.1)*, the stepsize* (3.3) *satisfies*

$$\frac{3}{4} \alpha_k^{BB_2} \leq \widetilde{\alpha}_k \leq \bar{c} \alpha_k^{BB_1},$$

where

$$\bar{c} = \left[(\lambda_1 + \lambda_n)^2 / (4 \lambda_1 \lambda_n) + 1 + \sqrt{\left((\lambda_1 + \lambda_n)^2 / (4 \lambda_1 \lambda_n) + 1\right)^2 - 4} \right] \Big/ 2.$$

Proof Denote $u_k = \dfrac{s_k^T A^2 s_k}{s_k^T A s_k}$ and $v_k = \dfrac{s_k^T A s_k}{s_k^T s_k}$, it is clear that $\lambda_1 \leq u_k, v_k \leq \lambda_n$ hold for all $k \geq 0$. The minimal and maximal eigenvalues of B_k in (3.2) can be rewritten as

$$\lambda_{\min}(B_k) = \frac{1}{2} \left(u_{k-1} + v_{k-1} - \sqrt{u_{k-1}^2 + 2 u_{k-1} v_{k-1} - 3 v_{k-1}^2} \right) \tag{5.2}$$

and

$$\lambda_{\max}(B_k) = \frac{1}{2} \left(u_{k-1} + v_{k-1} + \sqrt{u_{k-1}^2 + 2 u_{k-1} v_{k-1} - 3 v_{k-1}^2} \right). \tag{5.3}$$

Since

$$\frac{u_{k-1}}{v_{k-1}}=\frac{s_{k-1}^T A^2 s_{k-1} s_{k-1}^T s_{k-1}}{\left(s_{k-1}^T A s_{k-1}\right)^2}=\frac{\left(s_{k-1}^T A^{1/2}\right) A \left(A^{1/2} s_{k-1}\right) \left(s_{k-1}^T A^{1/2}\right) A^{-1} \left(A^{1/2} s_{k-1}\right)}{\left[\left(s_{k-1}^T A^{1/2}\right)\left(A^{1/2} s_{k-1}\right)\right]^2},$$

it follows from Kantorovich Inequality [19] that

$$1 \le \frac{u_{k-1}}{v_{k-1}} \le \frac{(\lambda_1+\lambda_n)^2}{4\lambda_1\lambda_n}, \tag{5.4}$$

which implies that

$$c \le \frac{2}{\left(u_{k-1}/v_{k-1}+1+\sqrt{(u_{k-1}/v_{k-1})^2+2u_{k-1}/v_{k-1}-3}\right)} \le 1,$$

where

$$c = 2\Big/\left[(\lambda_1+\lambda_n)^2/(4\lambda_1\lambda_n)+1+\sqrt{\left((\lambda_1+\lambda_n)^2/(4\lambda_1\lambda_n)+1\right)^2-4}\right]. \tag{5.5}$$

Due to

$$\begin{aligned}\lambda_{\min}(B_k) &= \frac{1}{2}(u_{k-1}+v_{k-1}-\sqrt{u_{k-1}^2+2u_{k-1}v_{k-1}-3v_{k-1}^2}) \\ &= v_{k-1}\frac{2}{\left(u_{k-1}/v_{k-1}+1+\sqrt{(u_{k-1}/v_{k-1})^2+2u_{k-1}/v_{k-1}-3}\right)},\end{aligned}$$

we have $cv_{k-1} \le \lambda_{\min}(B_k)$. It is clear from (5.3) that $\lambda_{\max}(B_k)$ is monotonically increasing relative to u_{k-1}. Therefore, Fixing u_{k-1} and imposing that $\frac{d\lambda_{\max}(B_k)}{dv_{k-1}}=0$, we obtain that $v_{k-1}=\frac{2}{3}u_{k-1}$, which satisfies that $\frac{v_{k-1}}{u_{k-1}} \le 1$. By substituting v_{k-1} into $\lambda_{\max}(B_k)$, it is not difficult to obtain that

$$\lambda_{\max}(B_k) \le \frac{4}{3}u_{k-1}.$$

Since B_k is symmetric positive definite, it follows that

$$cv_{k-1} \le \lambda_{\min}(B_k) \le \frac{1}{\widetilde{\alpha}_k} = \frac{g_k^T B_k g_k}{g_k^T g_k} \le \lambda_{\max}(B_k) \le \frac{4}{3}u_{k-1}.$$

where c is given by (5.5). Therefore, by the fact that $\alpha_k^{BB_1}=\frac{1}{v_{k-1}}$ and $\alpha_k^{BB_2}=\frac{1}{u_{k-1}}$, we obtain that

$$\frac{3}{4}\alpha_k^{BB_2} \le \widetilde{\alpha}_k \le \bar{c}\alpha_k^{BB_1},$$

where $\bar{c} = 1/c$. The proof is completed. □

It is observed by the numerical experiments that for strictly convex quadratic minimization the bound $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$ for the stepsize (3.3) is also very preferable. Together with the success of the BB stepsizes $\alpha_k^{BB_1}$ and $\alpha_k^{BB_2}$, we also take the truncated form (3.5) of the stepsize (3.3). For convenience in proof, the NABB method without any line search takes the form

$$x_{k+1} = x_k - \frac{1}{\overline{\alpha}_k} g_k, \tag{5.6}$$

where $\overline{\alpha}_k = \dfrac{1}{\widetilde{\alpha}_k}$ and $\widetilde{\alpha}_k$ is given by (3.5).

Let x^* be the unique minimizer of f and $\{x_k\}$ is the sequence generated by the NABB method without any line search. Denote $e_k = x^* - x_k$, according to (5.6) and the fact that $g_k = Ax_k - b$, we have, for all $k \ge 0$,

$$Ae_k = \alpha_k s_k.$$

Substituting $s_k = e_k - e_{k+1}$ in the above equality implies $e_{k+1} = \dfrac{1}{\overline{\alpha}_k}(\overline{\alpha}_k I - A)e_k$ for any $k \ge 0$. For the initial error e_0, there exist constants $d_1^0, d_2^0, \ldots, d_n^0$ such that $e_0 = \sum_{i=1}^n d_i^0 v_i$. Therefore, we have

$$e_{k+1} = \sum_{i=1}^{n} d_i^{k+1} v_i, \tag{5.7}$$

where

$$d_i^{k+1} = \left(\frac{\overline{\alpha}_k - \lambda_i}{\overline{\alpha}_k}\right) d_i^k = \prod_{j=0}^{k} \left(\frac{\overline{\alpha}_j - \lambda_i}{\overline{\alpha}_j}\right) d_i^0. \tag{5.8}$$

Since

$$\lambda_1 \le \frac{1}{\alpha_k^{BB_1}} \le \overline{\alpha}_k \le \frac{1}{\alpha_k^{BB_2}} \le \lambda_n, \tag{5.9}$$

which is implied by (3.5), we can easily obtain the following two lemmas according to Lemma 1 and Lemma 2 of [4]. The following two lemmas will be used in the proof of the convergence of the NABB method without any line search.

Lemma 5.2 *The sequence $\{d_1^k\}$ converges to zero Q-linearly with convergence factor $c = 1 - \lambda_1/\lambda_n$.*

Lemma 5.3 *If the sequence* $\{d_1^k\}, \{d_1^k\}, \ldots, \{d_l^k\}$ *all converge to zero for a fixed integer* l, $1 \le l < n$, *Then*

$$\liminf_{k\to\infty} |d_{l+1}^k| = 0.$$

Proof We prove the lemma by contradiction. Suppose that there exists a constant $\varepsilon > 0$ such that

$$\left(d_{l+1}^k\right)^2 \lambda_{l+1}^2 \ge \varepsilon \quad \text{and} \quad \left(d_{l+1}^k\right)^2 \lambda_{l+1}^3 \ge \varepsilon \tag{5.10}$$

for all $k \ge 0$. Due to (5.9), there exists $t_{k+1} \in [0, 1]$ such that

$$\begin{aligned}\overline{\alpha}_{k+1} &= t_{k+1}\frac{1}{\alpha_k^{BB_1}} + (1-t_{k+1})\frac{1}{\alpha_k^{BB_2}} \\ &= t_{k+1}\frac{e_k^T A^3 e_k}{e_k^T A^2 e_k} + (1-t_{k+1})\frac{e_k^T A^4 e_k}{e_k^T A^3 e_k}.\end{aligned}$$

Then, it follows from (5.7) and the orthonormality of the eigenvectors $\{v_1, v_2, \ldots, v_n\}$ that

$$\overline{\alpha}_{k+1} = t_{k+1}\frac{\sum_{i=1}^n \left(d_i^k\right)^2 \lambda_i^3}{\sum_{i=1}^n \left(d_i^k\right)^2 \lambda_i^2} + (1-t_{k+1})\frac{\sum_{i=1}^n \left(d_i^k\right)^2 \lambda_i^4}{\sum_{i=1}^n \left(d_i^k\right)^2 \lambda_i^3}. \tag{5.11}$$

Since the sequence $\{d_1^k\}, \{d_1^k\}, \ldots, \{d_l^k\}$ all convergence to zero, there exists $\widehat{k}$ sufficiently large such that

$$\sum_{i=1}^{l} \left(d_i^k\right)^2 \lambda_i^2 \le \frac{\varepsilon}{2} \qquad \text{and} \qquad \sum_{i=1}^{l} \left(d_i^k\right)^2 \lambda_i^3 \le \frac{\varepsilon}{2}. \tag{5.12}$$

for all $k \ge \widehat{k}$. By (5.11) and (5.12), we have, for any $k \ge \widehat{k}$,

$$t_{k+1}\frac{\lambda_{l+1}\sum_{i=l+1}^n \left(d_i^k\right)^2 \lambda_i^2}{\varepsilon/2 + \sum_{i=l+1}^n \left(d_i^k\right)^2 \lambda_i^2} + (1-t_{k+1})\frac{\lambda_{l+1}\sum_{i=l+1}^n \left(d_i^k\right)^2 \lambda_i^3}{\varepsilon/2 + \sum_{i=l+1}^n \left(d_i^k\right)^2 \lambda_i^3} \le \overline{\alpha}_{k+1} \le \lambda_n. \tag{5.13}$$

Since

$$\sum_{i=l+1}^{n} \left(d_i^k\right)^2 \lambda_i^2 \ge \left(d_{l+1}^k\right)^2 \lambda_{l+1}^2 \ge \varepsilon \;\; \text{and} \;\; \sum_{i=l+1}^{n} \left(d_i^k\right)^2 \lambda_i^3 \ge \left(d_{l+1}^k\right)^2 \lambda_{l+1}^3 \ge \varepsilon,$$

it follows from (5.13) that

$$\frac{2}{3}\lambda_{l+1} \leq \overline{\alpha}_{k+1} \leq \lambda_n \tag{5.14}$$

for all $k \geq \widehat{k}$, which means the bound

$$\left|1 - \frac{\lambda_{l+1}}{\overline{\alpha}_k}\right| \leq \max\left\{\frac{1}{2}, 1 - \frac{\lambda_{l+1}}{\lambda_{\max}}\right\}$$

holds for all $k \geq \widehat{k} + 1$. Finally, according to (5.14) and the first part of (5.8), we obtain, for all $k \geq \widehat{k} + 1$,

$$\left|d_{l+1}^{k+1}\right| = \left|1 - \frac{\lambda_{l+1}}{\overline{\alpha}_k}\right| \left|d_{l+1}^k\right| \leq \widehat{c}\left|d_{l+1}^k\right|,$$

where

$$\widehat{c} = \max\left\{\frac{1}{2}, 1 - \frac{\lambda_{l+1}}{\lambda_{\max}}\right\}. \tag{5.15}$$

This conclusion contradicts (5.10), therefore the lemma is true. The proof is completed. □

According to Theorem 1 of [4], it is not difficult to obtain the following theorem which indicates the NABB method, without any line search, is globally convergent for strictly convex quadratic minimization.

Theorem 5.1 *Let $f(x)$ be a strictly convex quadratic function, $\{x_k\}$ be the sequence by the NABB method without any line search and x^* is the unique minimizer of $f(x)$. Then, either $x_k = x^*$ for some finite iterate k, or the sequence $\{x_k\}$ converges to x^*.*

Proof It suffices to prove $\lim_{k\to\infty} \|e_k\|_2^2 = 0$ when $e_k = x^* - x_k \neq 0$ for all $k \geq 0$. It follows from (5.7) and the orthonormality of the eigenvectors of A that

$$\|e_k\|_2^2 = \sum_{i=1}^{n} \left(d_i^k\right)^2.$$

Therefore, the sequence of errors $\{e_k\}$ converges to zero if and only if each one of the sequences $\{d_k^i\}$ for any $i = 1, 2, \ldots, n$ converges to zero. Lemma 5.2 gives that $\{d_1^k\}$ converges to zero, and we only need to prove that $\{d_p^k\}$ converges to zero for $2 \leq p \leq n$ by induction on p. For any $2 < p \leq n$, we assume that $\left\{d_1^k\right\}, \left\{d_1^k\right\}, \ldots, \left\{d_{p-1}^k\right\}$ all tend to zero. Then for any given $\varepsilon > 0$, there exists $\widehat{k}$ sufficiently large such that

$$\sum_{i=1}^{p-1} \left(d_i^k\right)^2 \lambda_i^2 \leq \frac{\varepsilon}{2} \quad \text{and} \quad \sum_{i=1}^{p-1} \left(d_i^k\right)^2 \lambda_i^3 \leq \frac{\varepsilon}{2}\lambda_p \tag{5.16}$$

hold for all $k \geq \widehat{k}$. We know from (5.11) and (5.16)

$$t_{k+1}\frac{\lambda_p \sum_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^2}{\varepsilon/2+\sum_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^2}+(1-t_{k+1})\frac{\lambda_p \sum_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^3}{\lambda_p\varepsilon/2+\sum_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^3}\leq \overline{\alpha}_{k+1}\leq \lambda_n \tag{5.17}$$

for all $k \geq \widehat{k}$. Moreover, Lemma 5.3 implies that there exists $k_p \geq \widehat{k}$ such that

$$\left(d_p^{k_p}\right)^2\lambda_p^2 < \varepsilon.$$

Then $\left(d_p^{k_p}\right)^2\lambda_p^3 < \lambda_p\varepsilon$. Now let $k_0 > k_p$ be any integer such that $\left(d_p^{k_0-1}\right)^2\lambda_p^2 < \varepsilon$ and $\left(d_p^{k_0}\right)^2\lambda_p^2 \geq \varepsilon$. Then $\left(d_p^{k_0-1}\right)^2\lambda_p^3 < \lambda_p\varepsilon$ and $\left(d_p^{k_0}\right)^2\lambda_p^3 \geq \lambda_p\varepsilon$. It is clear that

$$\sum_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^2 \geq \left(d_p^k\right)^2\lambda_p^2 \geq \varepsilon \quad \text{and} \quad \sum_{i=p}^{n}\left(d_i^k\right)^2\lambda_p^3 \geq \left(d_p^k\right)^2\lambda_p^3 \geq \lambda_p\varepsilon, \tag{5.18}$$

hold for all $k_0 \leq k \leq j-1$. Here j is the first integer greater than k_0 such that $\left(d_p^{j-1}\right)^2\lambda_p^2 \geq \varepsilon$ and $\left(d_p^{j}\right)^2\lambda_p^2 < \varepsilon$. Therefore, $\left(d_p^{j-1}\right)^2\lambda_p^3 \geq \lambda_p\varepsilon$ and $\left(d_p^{j}\right)^2\lambda_p^3 < \lambda_p\varepsilon$. By (5.17) and (5.18), we have

$$\frac{2}{3}\lambda_{l+1} \leq \overline{\alpha}_{k+1} \leq \lambda_n \tag{5.19}$$

for all $k_0 \leq k \leq j-1$. Thus, by (5.19) and the first part of equation (5.8), we obtain

$$\left|d_p^{k+2}\right| \leq \widehat{c}\left|d_p^{k+1}\right|$$

for all $k_0 \leq k \leq j-1$, where $\widehat{c}$ is the constant (5.15), which satisfies $\widehat{c} < 1$. Finally, using the bound $\left|d_p^{k_0+1}\right| \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_1}\right)^2\left|d_p^{k_0-1}\right|$ implied by (5.9) and the first part of equation (5.8), we have

$$\left(d_p^k\right)^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_1}\right)^4\left(d_p^{k_0-1}\right)^2 \leq \left(\frac{\lambda_n - \lambda_1}{\lambda_1}\right)^4\frac{\varepsilon}{\lambda_p^2}$$

for all $k_0+1 \leq k \leq j+1$. Further, (5.8) implies the inequality $\left(d_p^{k_0}\right)^2 \leq (\frac{\lambda_n - \lambda_1}{\lambda_1})^2\left(d_p^{k_0-1}\right)^2$. It follows from the condition on k_0 and j that $\left(d_p^k\right)^2$ is bounded above by a constant multiple of ε for all $k \geq k_0-1$. Hence, since $\varepsilon > 0$ can be chosen arbitrarily small, we obtain $\lim_{k\to\infty}\left|d_p^k\right| = 0$ as required. Therefore,

$$\lim_{k\to\infty}\left|d_i^k\right| = 0, \qquad i = 1, 2, \ldots, n.$$

The proof is completed. □

Remark From Lemmas 5.2, 5.3 and Theorem 5.1, we know that the mehtod, which is the form of $x_{k+1} = x_k - \alpha_k g_k$ where $\alpha_k \in \left[\frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}\right]$, is globally convergent for strictly convex quadratic minimization (5.1).

6 Numerical results

6.1 Numerical results for general unconstrained optimization

We do some numerical experiments to test the effectiveness of the NABB method. The standard test function set includes 80 nonlinear unconstrained problems which contains most of problems in [24]. The standard test function set and their Fortran 77 code can be found in Andrei's website http://camo.ici.ro/neculai/AHYBRIDM. All methods are written in Fortran 77 and run on a PC with 3.20 GHz CPU processor, 4 GB RAM memory and Windows 7.

We have implemented the six BB-like methods [12], and find that the BB-like method with $\alpha_k^{SBB_4}$ (SBB4) outperforms the other five methods. Therefore, the SBB4 method with the stepsize (2.6) is chosen to be compared with the NABB method. Besides, the BB method with $\alpha_k^{BB_1}$ and the ABB method with the stepsize (2.7) are considered. Noted that the parameter ς in (2.7) is set to $\varsigma = 0.5$.

In the numerical experiments, we set the dimension of each problem to 10000 and choose these parameters: $\alpha_0^0 = 1/||g_0||_\infty$, $\varepsilon = 10^{-6}, \delta = 13, \lambda_{\min} = 10^{-30}, \lambda_{\max} = 10^{30}, \sigma = 10^{-4}, \eta_k = 1$ and $\eta_{\max} = \eta_{\min} = 1$. Clearly, in the new stepsizes mentioned in Sect. 3, $\cos\omega_k = \frac{g_k^T y_{k-1}}{\|g_k\|\,\|y_{k-1}\|}$, $\cos\theta_k = \frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|\,\|y_{k-1}\|}$ and $\cos\beta_k = \frac{g_k^T s_{k-1}}{\|g_k\|\,\|s_{k-1}\|}$. All methods adopt Zhang–Hager line search with strategy (4.2). The iteration is stopped if the inequality $\| g_k\|_\infty \leq 10^{-6}$ is satisfied, the number of iterations exceeds 30000, or the number of function evaluations exceeds 50000. All detailed numerical results are listed in Table 1. In Table 1, "Niter" and "Nf" denote the number of iteration and the number of function value evaluation, respectively, "Tcpu" denotes the CPU time (second), and "-" indicates the method is not stopped by the inequality $\| g_k\|_\infty \leq 10^{-6}$ when solving the corresponding problem. "NABB without $\delta\alpha_{k-1}$" in Table 1 and in Figs. 1, 2 and 3 denotes the variant of the NABB method, which is different from the NABB method only in that in the Step 2 of the NABB method $\alpha_k = \delta\alpha_{k-1}$ is replaced by $\alpha_k = \lambda_{\max}$ when $s_{k-1}^T y_{k-1} \leq 0$.

The performance profiles introduced by Dolan and More [25] are used to display the performances of the methods compared.

Table 1 Numerical results for the methods for general unconstrained optimization

P	NABB		SBB4		ABB		BB		NABB without $\delta\alpha_{k-1}$	
	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu
1	30/36	0.016	24/29	0.016	41/47	0.031	93/157	0.078	30/36	0.031
2	92/94	0.265	112/285	0.733	424/1028	2.668	101/103	0.281	92/94	0.250
3	104/112	0.047	72/379	0.047	118/426	0.062	79/229	0.047	87/297	0.047
4	32/33	0.016	27/28	0.016	36/37	0.016	36/39	0.016	32/33	0.016
5	31/32	0.016	29/30	0.016	54/55	0.047	44/48	0.031	31/32	0.016
6	53/54	0.016	47/48	0.016	53/54	0.016	53/54	0.016	53/54	0.031
7	1462/1463	0.577	1311/1312	0.484	1224/1225	0.484	1287/1288	0.452	1462/1463	0.515
8	1204/1205	0.655	1052/1053	0.562	1147/1148	0.577	962/964	0.515	1204/1205	0.624
9	1/2	0.000	1/2	0.000	1/2	0.000	1/2	0	1/2	0.000
10	229/230	0.109	718/719	0.328	551/552	0.234	78/79	0.047	229/230	0.109
11	399/417	0.218	412/420	0.218	369/381	0.203	501/809	0.312	399/417	0.203
12	1098/1099	1.014	–	–	1504/1505	1.342	1411/1412	1.248	1098/1099	0.967
13	108/112	0.078	710/1174	0.608	99/103	0.062	105/109	0.062	108/112	0.078
14	26/27	0.031	28/29	0.031	34/35	0.031	29/30	0.031	26/27	0.016
15	27/28	0.016	20/25	0.000	32/33	0.016	28/29	0.016	27/28	0.016
16	9/11	0.000	10/12	0.016	12/14	0.016	12/14	0.016	9/11	0.016
17	54/56	0.062	155/256	0.156	52/155	0.078	52/54	0.047	54/56	0.047
18	3/4	0.000	4/5	0.000	3/4	0.016	3/4	0	3/4	0.016
19	742/900	1.342	318/1222	1.342	248/1059	1.123	390/2266	2.028	269/1100	1.186
20	14/17	0.000	13/14	0.000	13/15	0.016	15/18	0	14/17	0.000
21	107/108	0.062	120/123	0.062	298/300	0.140	134/135	0.078	107/108	0.047
22	267/268	0.140	314/315	0.156	220/221	0.109	301/302	0.140	267/268	0.140
23	14/15	0.016	14/15	0.016	15/16	0.016	15/16	0.031	14/15	0.016

Table 1 continued

P	NABB		SBB4		ABB		BB		NABB without $\delta\alpha_{k-1}$	
	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu
24	19/25	0.016	14/114	0.031	25/127	0.031	19/124	0.031	21/123	0.031
25	19/25	0.000	20/122	0.031	19/121	0.016	19/121	0.016	18/120	0.031
26	84/85	9.703	80/81	9.095	80/81	9.033	91/92	10.343	84/85	9.423
27	54/92	0.031	20/45	0.031	27/63	0.016	59/128	0.047	54/92	0.031
28	803/804	0.328	793/794	0.312	957/958	0.390	868/869	0.343	803/804	0.328
29	15595/15599	18.611	–	–	–	–	–	–	–	–
30	18/19	0.031	19/123	0.218	48/672	1.248	48/672	1.264	48/672	1.186
31	2779/2780	2.574	3292/3293	2.995	3739/3740	3.385	7519/7629	6.958	2779/2780	2.574
32	–	–	–	–	–	–	–	–	–	–
33	84/89	0.031	93/101	0.047	106/113	0.047	115/177	0.062	84/89	0.047
34	4503/4504	1.794	9122/9123	3.541	4846/4847	1.888	8851/8855	3.416	4503/4504	1.747
35	3/4	0.000	4/5	0.000	3/4	0.000	3/4	0	3/4	0.000
36	8/9	0.000	9/10	0.016	9/10	0.000	9/13	0	8/9	0.000
37	622/653	0.234	251/444	0.109	269/565	0.125	387/609	0.173	248/546	0.125
38	58/61	0.016	38/231	0.016	51/245	0.031	86/220	0.031	37/231	0.031
39	97/98	0.062	195/278	0.125	80/81	0.047	83/84	0.047	97/98	0.047
40	29/30	0.000	43/44	0.016	14/15	0.016	24/25	0.016	29/30	0.016
41	121/125	0.125	–	–	–	–	–	–	–	–
42	31/32	0.031	73/154	0.078	34/35	0.031	28/29	0.016	31/32	0.016
43	87/88	0.047	114/213	0.078	78/79	0.031	78/79	0.047	87/88	0.047
44	1209/1210	0.359	1096/1097	0.328	1199/1200	0.343	1712/1714	0.499	1209/1210	0.359
45	72/73	0.031	58/60	0.031	58/59	0.031	103/104	0.047	72/73	0.031

Table 1 continued

P	NABB		SBB4		ABB		BB		NABB without $\delta\alpha_{k-1}$	
	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu
46	88/89	0.047	176/283	0.094	88/89	0.047	93/94	0.047	88/89	0.047
47	29/30	0.016	56/57	0.031	28/29	0.016	32/33	0.016	29/30	0.016
48	31/32	0.016	28/29	0.016	32/33	0.016	32/33	0.031	31/32	0.016
49	33/34	0.016	22/23	0.000	32/33	0.016	35/141	0.016	33/34	0.016
50	34/35	0.031	36/133	0.062	51/246	0.094	49/244	0.094	64/562	0.203
51	46/47	0.016	41/140	0.031	48/147	0.047	52/151	0.031	58/258	0.047
52	713/897	1.014	131/840	0.827	190/997	0.967	221/1086	0.889	174/1127	1.092
53	16/17	0.000	15/16	0.000	16/17	0.016	16/17	0.016	16/17	0.016
54	11/12	0.000	11/12	0.000	12/13	0.000	12/13	0	11/12	0.000
55	10083/10100	3.448	10104/10119	3.432	8658/8674	2.902	–	–	10083/10100	3.416
56	5165/5178	1.778	7905/7925	2.683	4521/4531	1.544	16693/17660	5.85	5165/5178	1.732
57	5373/5374	1.934	–	–	–	–	4764/4818	1.669	–	–
58	–	–	28547/28551	11.887	–	–	–	–	–	–
59	24/25	0.016	19/112	0.016	1039/33210	4.259	1039/33210	4.399	1039/33210	4.228
60	29/36	0.016	13/100	0.031	159/4812	1.076	159/4812	1.108	159/4812	1.061
61	7793/7794	2.808	12119/12120	4.306	–	–	13472/13473	4.774	7793/7794	2.746
62	11932/11933	4.259	21119/21120	7.457	–	–	13018/13019	4.524	11932/11933	4.228
63	7976/7977	2.980	18776/18777	6.958	–	–	10539/10540	3.853	7976/7977	2.964
64	3540/3541	8.689	–	–	–	–	–	–	14246/15603	37.660
65	1/2	0.000	1/2	0.000	1/2	0.000	1/2	0	1/2	0.000
66	115/117	0.109	–	–	71/272	0.281	86/219	0.2028	–	–
67	–	–	17955/17956	7.660	–	–	–	–	–	–

Table 1 continued

P	NABB		SBB4		ABB		BB		NABB without $\delta\alpha_{k-1}$	
	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu	Nitr/Nf	Tcpu
68	8/10	0.016	13/15	0.016	8/10	0.016	8/10	0.0156	8/10	0.016
69	1/2	0.000	1/2	0.000	1/2	0.000	1/2	0	1/2	0.000
70	9/10	0.000	17/93	0.109	9/10	0.016	9/10	0	9/10	0.016
71	53/54	0.016	43/44	0.016	53/54	0.016	53/54	0.016	53/54	0.016
72	29/36	0.047	150/3108	1.139	136/1680	0.640	136/1680	0.655	11/197	0.062
73	9/10	0.000	8/9	0.016	9/10	0.016	9/10	0.016	9/10	0.016
74	8/9	0.016	7/8	0.016	8/9	0.016	8/9	0.016	8/9	0.016
75	11/12	0.016	10/11	0.000	11/12	0.016	11/12	0.016	11/12	0.016
76	12/13	0.016	12/13	0.031	12/13	0.016	12/13	0.016	12/13	0.016
77	233/234	0.328	319/320	0.437	249/250	0.343	346/347	0.484	233/234	0.328
78	40/41	0.016	60/160	0.047	–	–	–	–	–	–
79	50/56	0.031	49/58	0.016	57/62	0.016	51/95	0.031	50/56	0.031
80	31/34	0.016	63/66	0.031	29/32	0.016	30/33	0.016	31/34	0.016

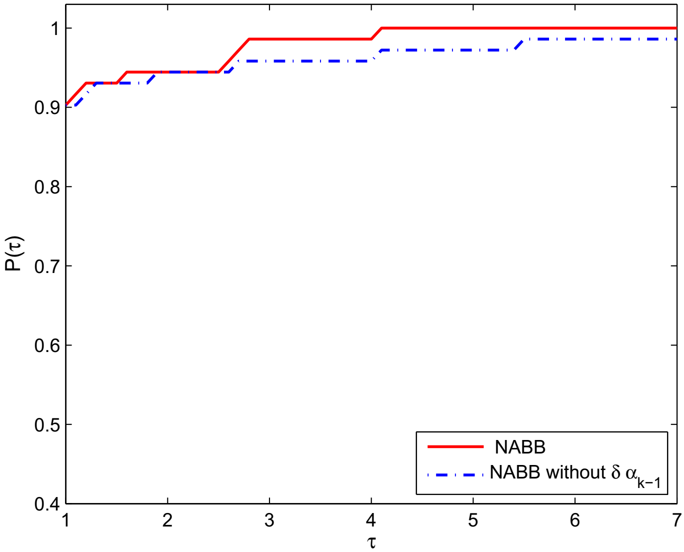

Fig. 1 Performance profile based on the number of iterations

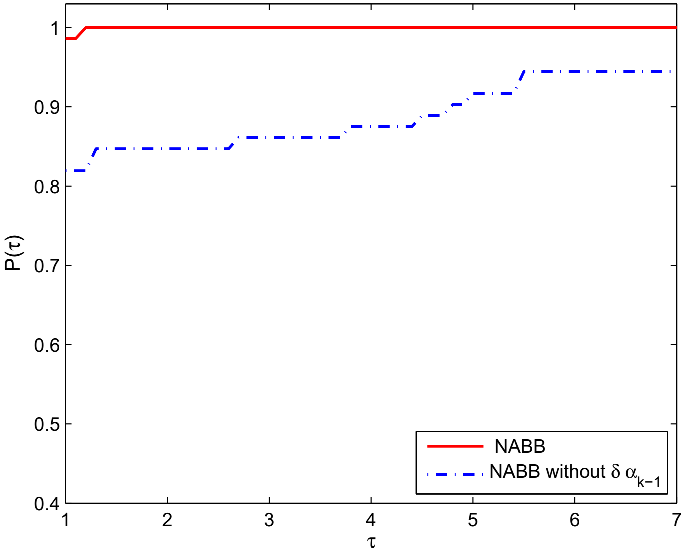

Fig. 2 Performance profile based on the number of function evaluations

We firstly examine the effectiveness of the stepsize (3.6) by comparing the NABB method with its variant. We see from Table 1 that the NABB method successfully solves 77 problems, which are 5 problems more than the variant. After eliminating those problems for which the NABB method or its variant is not stopped by the inequality $\| g_k \|_\infty \leq 10^{-6}$, 72 problems are left, and we only consider the performance for the

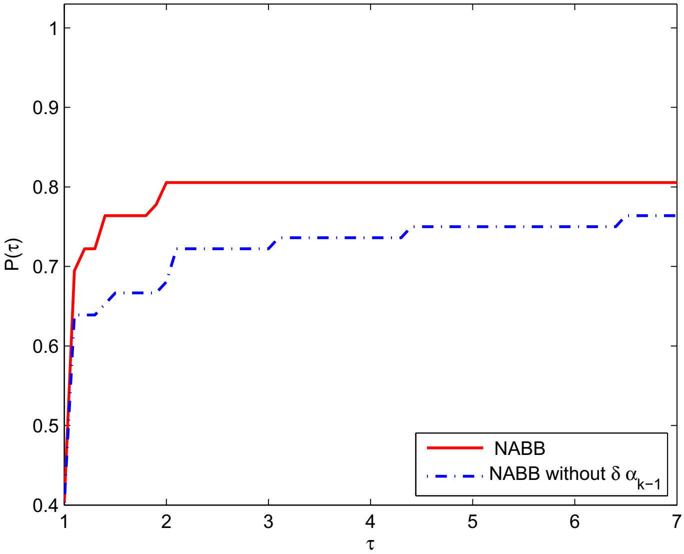

Fig. 3 Performance profile based on the CPU time

remaining 72 problems in the following analysis. Since the performance profile of the number of gradient evaluation is similar very much to that of the number of iteration, it will be discarded. Figures 1, 2 and 3 plot the performance profiles of the NABB method and the variant relative to the number of iterations, the number of function value evaluations and CPU time. As shown in Fig. 1, the NABB method requires less iterations than the variant. In Fig. 2, we see that NABB method requires much less function value evaluations than its variant, since the stepsize (3.6) is used when $s_{k-1}^T y_{k-1} \leq 0$. Figure 3 shows that the NABB method is faster than its variant. It indicates that the stepsize (3.6) is very efficient for the NABB method. So the NABB method is considered in the following experiments.

In what follows, we compare the NABB method with the SBB4 method, the ABB method and the BB method. As shown in Table 1, the NABB method successfully solves 96.25% test problems (77 out of 80) , while the SBB4 method , the ABB method and the BB method successfully solve 91.25% (73 out of 80), 86.25% (69 out of 80) and 90% (72 out of 80), respectively. After eliminating those problems for which one or more of the four methods is not stopped by the inequality $\| g_k \|_\infty \leq 10^{-6}$, 66 problems are left, and we only consider the performance for the remaining 66 problems in the following analysis. Similarly, the performance profile of the number of gradient evaluation is discarded. We observe from Fig. 4 that the NABB method is slightly better than the other three methods relative to the number of iterations. As shown in Fig. 5, the NABB method has a great advantage over the SBB4 method, the ABB method and the BB method, since with the least number of function value evaluations the NABB method successfully solves about 48% of the remaining problems, while the percentages of the SBB4 method, the ABB method and the BB method are about 38%, 32% and 20%, respectively. We see from Table 1 that for many problems the

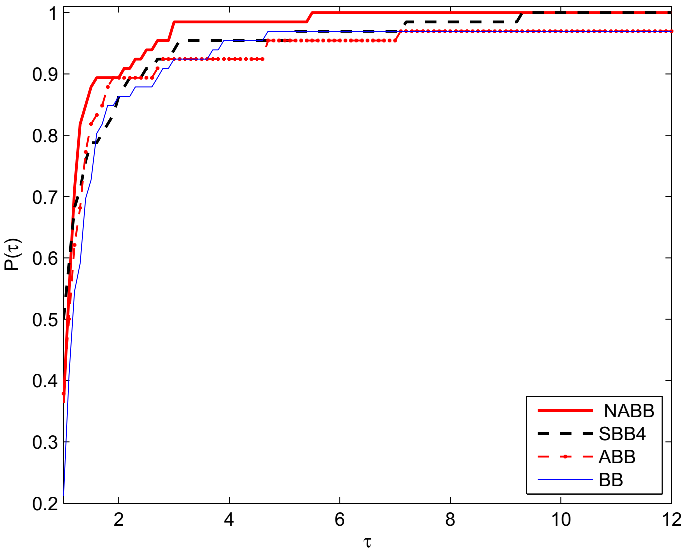

Fig. 4 Performance profile based on number of iterations

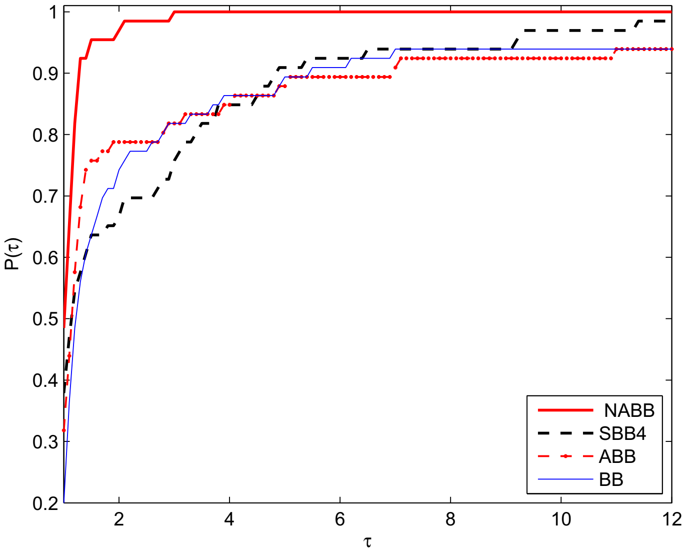

Fig. 5 Performance profile based on number of function evaluations

stepsize (3.5) or (3.6) always satisfies the condition (4.1) and thus Zhang–Hager line search is not evoked during the iterations. Figure 6 shows that the NABB method is faster than the other three methods. It indicates that the NABB method is very efficient.

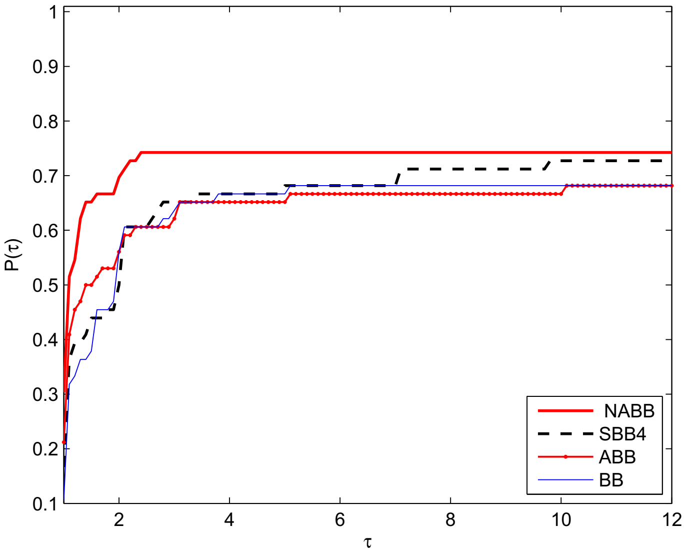

Fig. 6 Performance profile based on the CPU time

6.2 Numerical results for strictly convex quadratic minimization

We compare the NABB method with the BB method, the ABB method with the stepsize (2.7), the alternate step gradient method (AS) [26] and the Yuan's method (YM) [27] without any line search for strictly convex quadratic minimization (5.1). Noted that the parameter ς in (2.7) is set to $\varsigma = 0.5$. These methods are coded in Matlab. The stopping criterion is chosen as

$$\| g_k \|_2 \leq \eta \| g_0 \|_2$$

with different η, and the iteration is also stopped if the number of iteration exceeds 10000.

Three sets of test problems are considered. The first set of test problems are randomly generated in the same way as [28]. The dimension of each problem is set to 1000. The matrix A is obtained by running Matlab function *sprandsym* with $density = 0.8$, $kind = 1$, and the different condition numbers $\kappa(A) = 10^1, 10^2, \ldots, 10^7$. For each instance of A, the optimal solution x^* is generated by running Matlab function *rand* with entries in [10, 10] and $b = Ax^*$, and for each problem, 5 starting points are generated by running Matlab function *rand* with entries in [10, 10].

In Table 2 we report the average number of iterations required by the four methods with 5 different starting points for the first set of test problems. As shown in Table 2, the NABB method is superior to the BB method and the AS method, and the NABB method has an distinct advantage over the YM method for high tolerances or large condition number $\kappa(A)$. When the condition number $\kappa(A)$ is small, the NABB method and the ABB method have similar performance, and as

Table 2 The average number of iterations required by the five methods with 5 different starting points for the first set of problems

η	$\kappa(A)$	NABB	BB	ABB	AS	YM
10^{-2}	10^1	8	10.4	10.4	8.2	7.6
	10^2	17	17.8	18	16.2	14.6
	10^3	17	20.2	22.8	19.6	18
	10^4	17	19.4	19.2	19.4	18
	10^5	17	19.6	20	21.2	17.6
	10^6	17.6	19.4	19.8	17.4	17.4
	10^7	18.6	20	20	23.6	19.2
10^{-4}	10^1	17	18	19.4	17.2	15.8
	10^2	46.8	55.6	46.4	47.2	51
	10^3	108.8	119.6	107.6	129.2	227.6
	10^4	185.8	194.6	203.2	208.4	633.2
	10^5	189	239.2	219.8	266.4	730
	10^6	184.8	215.8	252.6	237.4	638.6
	10^7	204.4	209.2	212	210.8	617.2
10^{-6}	10^1	24	25.8	26.2	26.8	25
	10^2	78.2	78	70.6	79.4	108.2
	10^3	208.8	218	206.4	216.6	603.6
	10^4	542.8	696.8	601.2	650.4	3859.6
	10^5	1375.4	1514.2	1716	1506.2	>10000
	10^6	2669.4	2857.2	3706.8	3138.6	>10000
	10^7	2518	2671.4	4090.6	3034.2	>10000

the condition number $\kappa(A)$ increases, the NABB method outperforms the ABB method.

The second set of test problems are randomly generated in the same way as [29]. The matrix A is the form of $A = QDQ^T$, where $Q = (I - 2\omega_3\omega_3^T)(I - 2\omega_2\omega_2^T)(I - 2\omega_1\omega_1^T)$, $D = Diag(\sigma_1, \ldots, \sigma_n)$ is a diagonal matric and $n = 5000$. Here ω_1, ω_2 and ω_3 are unitary random vectors, $\sigma_1 = 1$, $\sigma_n = \kappa(A)$, and σ_j is randomly generated by running Matlab command *rand* with entries in $[1, \kappa(A)]$. The constant vector b are randomly generated by running Matlab command *rand* with entries in [10, 10]. The starting point is set to $x_0 = (0, 0, \ldots, 0)^T$.

From Table 3 we see that the NABB method outperforms the BB method, the AS method and the YM method, and the NABB method performs similar to the ABB method.

The third set of test problems, which are from [29], are the linear systems $Ax = b$ with $A \equiv (a_{ij})$ given by

$$a(i,j) = \begin{cases} 2/h^2, & \text{if } i = j, \\ -1/h^2, & \text{if } i = j \pm 1, \\ 0, & \text{otherwise,} \end{cases}$$

Table 3 The number of iterations required by the five methods for the second set of problems

θ	$\kappa(A)$	NABB	BB	ABB	AS	YM
10^{-2}	10^1	10	17	14	13	8
	10^2	25	31	29	24	26
	10^3	56	63	56	70	66
	10^4	140	260	151	173	424
	10^5	200	338	235	304	826
	10^6	1726	4868	3498	2289	>1000
10^{-4}	10^1	17	23	26	18	18
	10^2	55	58	54	55	69
	10^3	167	206	144	238	169
	10^4	493	804	504	794	2154
	10^5	1198	1817	2268	1635	>10000
	10^6	2438	5574	1876	6080	>10000
10^{-6}	10^1	25	30	34	26	26
	10^2	81	82	80	102	81
	10^3	268	323	267	273	352
	10^4	876	1095	799	1144	5806
	10^5	1907	3287	6030	2784	>10000
	10^6	7598	>10000	8912	>10000	>10000

Table 4 The average number of iterations required by the five methods with 5 different starting points for the third set of problems

θ	NABB	BB	ABB	AS	YM
10^{-2}	9	9	9.4	10.8	8.4
10^{-3}	24.8	28.2	26.6	31.4	26
10^{-4}	65.4	70	65.2	79	83.2
10^{-5}	150.4	229.8	192.6	203.2	478
10^{-6}	437.2	541	546	570.4	2695.6

where $h = 11/n$ and $n = 1000$. The solution x^*, the initial point x_0 and b are generated with the same way as that in the first set of test problems. The kind of problems often appear in the numerical solution of two-point boundary value problems. As shown in Table 4, the NABB method performs better than the BB method, the AS method and the ABB method, and the NABB method has obvious advantage over YM method for high precision.

7 Conclusion and discussion

In this paper we have proposed a new adaptive BB (NABB) method with Zhang–Hager line search for unconstrained optimization problem. It is remarkable that the stepsize

(3.3) is picked adaptively in the interval which contains the two BB stepsizes. Besides, for the case of $s_{k-1}^T y_{k-1} \leq 0$ a strategy for the stepsize is developed to speed up the convergence rate of the NABB method. Furthermore, we apply the NABB method without any line search to strictly convex quadratic minimization

Although the choice of $t = 0$ in (3.2) is considered, we think the NABB method will illustrate better performance if t dynamically varies based on some rules. Two natural questions to ask are that how the choice of the parameter t affects the behaviour of the NABB method and what is the best choice of the parameter t for the NABB method. Besides, for the case of $s_{k-1}^T y_{k-1} \leq 0$ the stepsize is given by (3.6) and its effectiveness is tested, however we feel that there may be better choices.

Future research includes developing some more efficient rules for better choice of the parameter t and exploiting more efficient choice for the stepsize for the case of $s_{k-1}^T y_{k-1} \leq 0$.

Acknowledgements We would like to thank Professor Dai, Y. H. and Dr. Cou Caixia for their Fortran code for the BB method. We also wish to thank Dr. Huang yakui for his careful reading and helpful comments, and thank two anonymous referees for their kind and valuable comments. These comments will help to improve the quality of this paper. This research is supported by National Science Foundation of China (Nos. 11461021, 11601012), Guangxi Science Foundation (Nos. 2014GXNSFAA118028, 2015GXNSFAA139011), Scientific Research Foundation of Guangxi Education Department (No. 2013YB236), Scientific Research Project of Hezhou University (Nos. 2014YBZK06, 2016HZXYSX03). Guangxi Colleges and Universities Key Laboratory of Symbolic Computation and Engineering Data Processing.

References

1. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**, 141–148 (1988)
2. Cauchy, A.: Méthode générale pour la résolution des systéms déquations simultanées. Comp. Rend. Sci. Paris **25**, 46–89 (1847)
3. Birgin, E.G., Martínez, J.M., Raydan, M.: Spectral projected gradient methods: review and perspectives. J. Stat. Softw. **60**(3), 1–21 (2014)
4. Raydan, M.: On the Barzilai and Borwein choice of steplength for the gradient method. IMA J. Numer. Anal. **13**, 321–326 (1993)
5. Dai, Y.H., Liao, L.Z.: R-linear convergence of the Barzilai and Borwein gradient method. IMA J. Numer. Anal. **22**(1), 1–10 (2002)
6. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**, 707–716 (1986)
7. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Optim. **7**, 26–33 (1997)
8. Dai, Y.H., Hager, W.W., Schittkowski, K., et al.: The cyclic Barzilai–Borwein method for unconstrained optimization. IMA J. Numer. Anal. **26**(3), 604–627 (2006)
9. Glunt, W., Hayden, T.L., Raydan, M.: Molecular conformations from distance matrices. J. Comput. Chem. **14**(1), 114–120 (1993)
10. Dai, Y.H.: A new analysis on the Barzilai-Borwein gradient method. J. Oper. Res. Soc. China **1**(2), 187–198 (2013)
11. Zhou, B., Gao, L., Dai, Y.H.: Gradient methods with adaptive stepsizes. Comput. Optim. Appl. **35**(1), 69–86 (2006)
12. Biglari, F., Solimanpur, M.: Scaling on the spectral gradient method. J. Optim. Theory Appl. **158**(2), 626–635 (2013)
13. Dai, Y.H., Yuan, J.Y., Yuan, Y.X.: Modified two-point stepsize gradient methods for unconstrained optimization. Comput. Optim. Appl. **22**, 103–109 (2002)

14. Zhang, J.Z., Deng, N.Y., Chen, L.H.: New quasi-Newton equation and related methods for unconstrained optimization. J. Optim. Theory Appl. **102**, 147–167 (1999)
15. Wei, Z.X., Li, G.Y., Qi, L.Q.: New quasi-Newton methods for unconstrained optimization problems. Appl. Math. Comput. **175**, 1156–1188 (2006)
16. Xiao, Y.H., Wang, Q.Y., Wang, D., et al.: Notes on the Dai-Yuan-Yuan modified spectral gradient method. J. Comput. Appl. Math. **234**(10), 2986–2992 (2010)
17. Biglari, F., Hassan, M.A., Leong, W.J.: New quasi-Newton methods via higher order tensor models. J. Comput. Appl. Math. **235**(8), 2412–2422 (2011)
18. Dennis, J.E., Wolkowicz, H.: Sizing and least change secant methods. SIAM J. Numer. Anal. **30**, 1291–1313 (1993)
19. Yuan, Y.X., Sun, W.Y.: Theory and Methods of Optimization. Science Press of China, Beijing (1999)
20. Watkins, D.S.: Fundamentals of matrix computations. Phys. Plasmas **59**, 199 (2002)
21. Luengo, F., Raydan, M.: Gradient method with dynamical retards for large-scale optimization problems. Electron. Trans. Numer. Anal. ETNA **16**(539), 186–193 (2003)
22. Zhang, H.C., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. **14**, 1043–1056 (2004)
23. Birgin, E.G., Martínez, J.M., Raydan, M.: Nonmonotone spectral projected gradient methods for convex sets. SIAM J. Optim. **10**(4), 1196–1211 (2000)
24. Andrei, N.: An unconstrained optimization test functions collection. Adv. Model. Optim. **10**, 147–161 (2008)
25. Dolan, E.D., More, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**, 201–213 (2002)
26. Dai, Y.H.: Alternate step gradient method. Optimization **52**(4–5), 395–415 (2010)
27. Yuan, Y.X.: A new stepsize for the steepest descent method. J. Comput. Math. **24**, 149–156 (2006)
28. Asmundis, R.D., Serafino, D.D., Riccio, F., et al.: On spectral properties of steepest descent methods. IMA J. Numer. Anal. **33**(4), 1416–1435 (2013)
29. Friedlander, A., Martinez, J.M., Molina, B., et al.: Gradient method with retards and generalizations. SIAM J. Numer. Anal. **36**, 275–289 (1999)