OPTIMIZATION, 2017
https://doi.org/10.1080/02331934.2017.1399392

Taylor & Francis
Taylor & Francis Group

An efficient gradient method with approximate optimal stepsize for the strictly convex quadratic minimization problem

Zexian Liu[a,b], Hongwei Liu[a], Xiaoliang Dong[c]

[a]School of Mathematics and Statistics, Xidian University, Xi'an, P.R. China; [b]School of Mathematics and Computer Science, Hezhou University, Hezhou, P.R. China; [c]School of Mathematics and Information, Beifang University of Nationalities, Yinchuan, P.R. China

ABSTRACT

In this paper, a new type of stepsize, approximate optimal stepsize, for gradient method is introduced to interpret the Barzilai–Borwein (BB) method, and an efficient gradient method with an approximate optimal stepsize for the strictly convex quadratic minimization problem is presented. Based on a multi-step quasi-Newton condition, we construct a new quadratic approximation model to generate an approximate optimal stepsize. We then use the two well-known BB stepsizes to truncate it for improving numerical effects and treat the resulted approximate optimal stepsize as the new stepsize for gradient method. We establish the global convergence and *R*-linear convergence of the proposed method. Numerical results show that the proposed method outperforms some well-known gradient methods.

ARTICLE HISTORY
Received 2 March 2017
Accepted 25 October 2017

KEYWORDS
Barzilai–Borwein (BB) method; BFGS update formula; approximating optimal stepsize; strictly convex quadratic minimization; gradient method

1. Introduction

In this paper, we study the strictly convex quadratic minimization problem. The oldest and simplest iterative method for solving the problem is the steepest descent (SD) method [1]. The SD method, despite of the minimal storage requirements and low computational cost per iteration, has long been considered very bad and ineffective because of its slow convergence rate and its oscillatory behaviour [2]. Since the Barzilai–Borwein (BB) method [3] was proposed by Barzilai and Borwein in 1988, the gradient method has achieved extensive concern due to its surprising numerical behavior. As a result, the interest for gradient methods has been renewed and many efficient gradient methods have been developed. The BB method was proved to be superlinearly convergent for two dimensional strictly convex quadratic functions [3], and to be globally and *R*-linearly convergent [4,5] for any dimensional strictly convex quadratic functions. Dai [6] presented the alternate step gradient method (AS) which used the SD stepsize and the BB stepsize alternately. Dai and Yuan [7] also proposed the alternate minimization gradient method (AM) whose stepsize was determined by alternately minimizing the function value and minimizing the gradient norm along the line of the negative gradient direction. Zhou et al. [8] proposed an adaptive BB method (ABB) whose stepsize was adaptively chosen between the two well-know BB stepsizes. It is generally accepted that the ABB method is very efficient when the precision requirement increases or the condition number is large. Besides, Yuan [9] presented a new gradient method (YM) which was very efficient for small scale problems. Recently, Liu et al. [10] proposed a new adaptive BB method for strictly convex quadratic minimization. Zheng and Zheng [11] presented a modified BB method (MBB) based on a multi-step quasi-Newton condition [12].

CONTACT Zexian Liu ✉ liuzexian2008@163.com

In this paper we present an efficient gradient method with an approximate optimal stepsize for the strictly convex quadratic minimization problem. We first introduce a new type of stepsize, approximate optimal stepsize, for gradient method, and use it to interpret the BB method. Based on a multi-step quasi-Newton condition, we construct a new quadratic approximation model to generate an approximate optimal stepsize, use the two well-known BB stepsizes to truncate it for improving numerical effects, and treat the resulting approximate optimal stepsize as the new stepsize for gradient method for strictly convex quadratic minimization problem. We also establish the global convergence and the R-linear convergence of the proposed method. Numerical results on three sets of problems show that our method is competitive to the conjugate gradient method and is superior to the BB method, the AS method, the YM method, the MBB method, the ABB method and the AM method.

The rest of the paper is organized as follows. In Section 2 we introduce a new type of stepsize, approximate optimal stepsize, and construct a new quadratic approximation model to design an efficient approximate optimal stepsize and truncate it for gradient method, and we also do a simple numerical experiment to show the performance of the proposed method, the BB method and the SD method. In Section 3 we consider the global convergence and the R-linear convergence of the proposed method. In Section 4 we illustrate the numerical results. Conclusions and discussions are made in the last section.

2. Gradient method with approximate optimal stepsize

Consider the problem of minimizing a convex quadratic function:

$$\min_{x\in R^n} f(x) = \frac{1}{2}x^TAx - b^Tx, \tag{2.1}$$

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

The gradient method for solving (2.1) takes the following form:

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

where $g_k = Ax_k - b$ and α_k is the stepsize which depends on the method under consideration. In the SD method, the stepsize is computed by

$$\alpha_k^{SD} = \arg\min_{\alpha>0} f(x_k - \alpha g_k) = \frac{g_k^Tg_k}{g_k^TAg_k},$$

and in the BB method, the stepsize is given by

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

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

Now we first introduce a new class of stepsize, approximate optimal stepsize, for gradient method, and use it to interpret the BB method.

Definition 2.1: Let $\phi(\alpha)$ be an approximation model of $f(x_k - \alpha g_k)$. A positive number α^* is called **approximate optimal stepsize** associated to $\phi(\alpha)$ for gradient method, if α^* satisfies

$$\alpha^* = \arg\min_{\alpha>0} \phi(\alpha).$$

Different approximation models lead to different approximate optimal stepsizes and different approximate optimal stepsizes correspond to different gradient methods. Clearly, the effectiveness of approximate optimal stepsize will rely on the approximation model $\phi(\alpha)$. The approximate optimal stepsize is different from α_k^{SD} which will lead to expensive computational cost and poor numerical behavior. The approximate optimal stepsize is generally calculated easily, can avoid producing the zig phenomenon and can be extended to general unconstrained optimization.

If one takes

$$\phi_1(\alpha) = f(x_k) - \alpha \parallel g_k\|^2 + \frac{1}{2}\alpha^2 g_k^T \frac{s_{k-1}^T y_{k-1}}{\parallel s_{k-1}\|^2} I g_k, \tag{2.3}$$

where $\frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2} I$ can be regarded as an approximation to the Hessian matrix A in (2.1), as an approximation model of $f(x_k - \alpha g_k)$, it is clear that

$$\alpha_k^{BB_1} = \frac{\|s_{k-1}\|^2}{s_{k-1}^T y_{k-1}} = \arg \min_{\alpha>0} \phi_1(\alpha),$$

which implies by the Definition 2.1 that $\alpha_k^{BB_1}$ is an approximate optimal stepsize associated to $\phi_1(\alpha)$ for gradient method. The approximation model $\phi_1(\alpha)$ in (2.3) seems nice and thus the BB method is very efficient, which explains partly that why the BB method is able to exhibit so surprising numerical behaviour.

In what follows, we will construct a new quadratic approximation model to generate an approximate optimal stepsize, use the two well-known BB stepsizes to truncated it and treat the resulted stepsize as the new stepsize for gradient method.

Let us consider the approximation model of $f(x_k - \alpha g_k)$:

$$\phi(\alpha) = f(x_k) - \alpha \parallel g_k\|^2 + \frac{1}{2}\alpha^2 g_k^T B_k g_k, \tag{2.4}$$

where B_k, which can be regarded as an approximation to the Hessian matrix A in (2.1), is symmetric and positive definite. It is clear that the approximate optimal stepsize associated to $\phi(\alpha)$ is

$$\alpha_k = \frac{\|g_k\|^2}{g_k^T B_k g_k}. \tag{2.5}$$

Obviously, the matrix B_k is very important to the effectiveness of the approximate optimal stepsize (2.5).

In the BB method, $\frac{s_{k-1}^T y_{k-1}}{\|s_{k-1}\|^2} I$ is a approximation to the Hessian matrix A. However, it is generally said the scalar matrix can not approximate the Hessian matrix A very well though it is easy to calculate. Consequently, consider computational cost and storage cost, we adopt a full matrix B_k, which is generated by imposing the BFGS update formula on a scalar matrix $\lambda_k I$. Clearly, the choice of λ_k is also very crucial to the effectiveness of approximate optimal stepsize.

Using interpolation condition, Ford and Moghrabi [12] presented a multi-step quasi-Newton method based on the modified secant equation $B_k r_{k-1} = w_{k-1}$ where

$$r_{k-1} = s_{k-1} - \xi_{k-1} s_{k-2}, \quad w_{k-1} = y_{k-1} - \xi_{k-1} y_{k-2} \tag{2.6}$$

and ξ_{k-1} is a scaling parameter, for unconstrained optimization problems. In the MBB method [11], the stepsize is determined by

$$\alpha_k^{MBB_1} = \frac{r_{k-1}^T r_{k-1}}{r_{k-1}^T w_{k-1}},$$

where r_{k-1} and w_{k-1} are given by (2.6). The other stepsize based on the above modified secant equation can be denoted by $\alpha_k^{MBB_2} = \frac{r_{k-1}^T w_{k-1}}{w_{k-1}^T w_{k-1}}$.

It is accepted usually that $\alpha_k^{BB_1}$ is superior to $\alpha_k^{BB_2}$ in many cases. We also observe by numerical experiments that, however, for some quadratic minimization problems the BB method with $\alpha_k^{BB_2}$ outperforms the BB method with $\alpha_k^{BB_1}$. This is the case for $\alpha_k^{MBB_1}$ and $\alpha_k^{MBB_2}$. Together with the effectiveness of the MBB method, we do not want to abandon any of two stepsizes $\alpha_k^{MBB_1}$ and $\alpha_k^{MBB_2}$ in the constructing the scalar matrix.

Base on the above observations, we take the scalar matrix as

$$\lambda_k I = \left[(1-\mu_k)\frac{r_{k-1}^T w_{k-1}}{\| r_{k-1}\|^2} + \mu_k \frac{\| w_{k-1}\|^2}{r_{k-1}^T w_{k-1}} \right] I, \tag{2.7}$$

where $\mu_k \in [0,1]$ is an parameter which measures the contributions of $\frac{r_{k-1}^T w_{k-1}}{\|r_{k-1}\|^2} I$ and $\frac{\|w_{k-1}\|^2}{r_{k-1}^T w_{k-1}} I$ to the scalar matrix $\lambda_k I$. Usually, $\mu_k < 0.3$, which means that the matrix $\frac{r_{k-1}^T w_{k-1}}{\|r_{k-1}\|^2} I$ plays a dominate role in the scalar matrix $\lambda_k I$, is preferable.

Imposing the BFGS update formula [13] on $\lambda_k I$ in (2.7) gives

$$B_k = \lambda_k I - \frac{\lambda_k I s_{k-1} s_{k-1}^T \lambda_k I}{s_{k-1}^T \lambda_k I s_{k-1}} + \frac{y_{k-1} y_{k-1}^T}{s_{k-1}^T y_{k-1}}. \tag{2.8}$$

Since A is symmetric and positive definite, it is easy to verify that $\frac{r_{k-1}^T r_{k-1}}{r_{k-1}^T w_{k-1}} > 0$, which implies that $\lambda_k > 0$ and thus the resulted matrix B_k in (2.8) is symmetric and positive definite.

By (2.5) and (2.8) we obtain the approximate optimal stepsize associated to $\phi\left(\alpha\right)$ in (2.4) with (2.8):

$$\widehat{\alpha}_k^{GM\_AOS} = \frac{\|g_k\|^2}{g_k^T B_k g_k} = \frac{\|g_k\|^2}{\lambda_k \left(\|g_k\|^2 - \left(g_k^T s_{k-1}\right)^2 / \|s_{k-1}\|^2 \right) + \left(g_k^T y_{k-1}\right)^2 / s_{k-1}^T y_{k-1}}. \tag{2.9}$$

It is observed that the stepsize (2.9) lies in the interval $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$ in many cases and the bound $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$ for the stepsize (2.9) is very preferable in the numerical experiments. Together with the success of the stepsize $\alpha_k^{BB_1}$ and $\alpha_k^{BB_2}$, it is reasonable to impose the bound $\left[\alpha_k^{BB_2}, \alpha_k^{BB_1}\right]$ on the stepsize (2.9). Therefore, in practice we take the truncation form of the approximate optimal stepsize:

$$\alpha_k^{GM\_AOS} = \min\left\{\alpha_k^{BB_1}, \max\left\{\widehat{\alpha}_k^{GM\_AOS}, \alpha_k^{BB_2}\right\}\right\}, \tag{2.10}$$

where $\widehat{\alpha}_k^{GM\_AOS}$ is given by (2.9), as the new stepsize for gradient method. The gradient method with approximate optimal stepsize (2.10) is denoted by GM_AOS in this paper.

Since the stepsize (2.10) is computed by (2.9) which is truncated by the two well-known BB stepsizes and the scalar matrix is designed by (2.7) which is updated to B_k, it is reasonable to expect that GM_AOS illustrates good numerical behaviours.

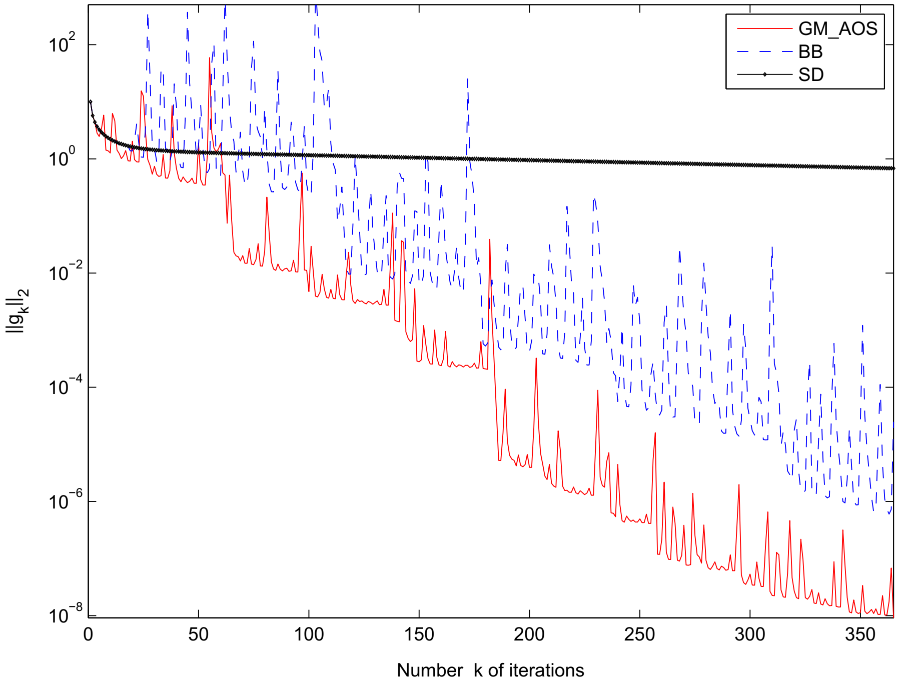

Figure 1. Performances of GM_AOS, BB and SD for a 100-dimensional problem.

Remark: It seems that GM_AOS is a little similar to the MBB method, but there is essential difference: In GM_AOS, we first determine the scalar matrix $\lambda_k I$ by (2.7), apply the BFGS update formula on the scalar matrix $\lambda_k I$ to generate B_k, use (2.5) to compute the approximate optimal stepsize (2.9), truncate it by two well-known BB stepsizes and obtain the approximate optimal stepsize (2.10). While the stepsize $\alpha_k^{MBB_1}$ in the MBB method only corresponds to the scalar $\frac{1}{\lambda_k}$ in (2.7) with $\mu_k = 0$.

It is clear that the approximate optimal stepsize (2.10) is a little similar to α_k^{SD} in form and is related to the BB stepsizes $\alpha_k^{BB_1}$ and $\alpha_k^{BB_2}$, which arises a question: What about the performances of GM_AOS, the BB method and the SD method. Notice that the choice of stepsize for the BB method is $\alpha_k^{BB_1}$ in this paper.

To illustrate the behaviour of GM_AOS, the BB method and the SD method, a simple test problem [8] is done (see Section 4 for more numerical experiments with more well-known gradient methods). Let $A = diag(a_1, \ldots, a_{100})$ with $a_1 = 0.1$ and $a_i = i$ for $i = 2, \ldots, 100$, $b = (1, \ldots, 1)^T$ and the initial guess is chosen as $x_0 = 0 \in R^{100}$. In GM_AOS, $\xi_k = 0.1$, $\mu_k = 0.2$. In order to get $\|g_k\| \le 10^{-9}\|g_0\|$, the BB method and the SD method require 463 iterations and 9384 iterations, respectively, while GM_AOS only requires 364 iterations. Figure 1 reports the gradient norms for GM_AOS, the BB method and the SD method at each iteration. We observe from Figure 1 that GM_AOS outperforms the BB method and the SD method.

3. Convergence analysis

In this section we consider the global convergence and R-linear convergence of GM_AOS. Let $\{x_k\}$ be the sequence generated by GM_AOS and x^* be the optimal solution of problem (2.1). Assume A has the eigenvalues $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ with the associated with orthonormal eigenvectors $\{v_1, v_2, \ldots, v_n\}$.

For convenience in proof, we rewrite the iterate form of GM_AOS as

$$x_{k+1} = x_k - \frac{1}{\overline{\alpha}_k} g_k, \tag{3.1}$$

where $\overline{\alpha}_k = 1/\alpha_k^{GM\_AOS}$ and $\alpha_k^{GM\_AOS}$ is given by (2.10). Denote $e_k = x^* - x_k$. By (3.1) and the fact that $g_k = Ax_k - b$, we have, for all $k \geq 0$,

$$Ae_k = \overline{\alpha}_k s_k.$$

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 $k \geq 0$. 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

$$e_{k+1} = \sum_{i=1}^{n} d_i^{k+1} v_i, \tag{3.2}$$

where

$$d_i^{k+1} = \left(\frac{\overline{\alpha}_k - \lambda_i}{\overline{\alpha}_k}\right) d_i^k = \prod_{j=0}^{k} \left(\frac{\overline{\alpha}_j - \lambda_i}{\overline{\alpha}_j}\right) d_i^0. \tag{3.3}$$

Since $\lambda_1 \leq \frac{1}{\alpha_k^{BB_1}} \leq \overline{\alpha}_k \leq \frac{1}{\alpha_k^{BB_2}} \leq \lambda_n$, there exists $\tau_{k+1} \in [0, 1]$ such that

$$\overline{\alpha}_{k+1} = \tau_{k+1} \frac{1}{\alpha_k^{BB_1}} + (1 - \tau_{k+1}) \frac{1}{\alpha_k^{BB_2}} = \tau_{k+1} \frac{e_k^T A^3 e_k}{e_k^T A^2 e_k} + (1 - \tau_{k+1}) \frac{e_k^T A^4 e_k}{e_k^T A^3 e_k}. \tag{3.4}$$

We can directly obtain the following lemma by Lemma 1 of [4].

Lemma 3.1: *The sequence $\{d_1^k\}$ converges to zero Q-linearly with convergence factor $c = 1 - \lambda_1/\lambda_n$.*

The following lemma and Lemma 3.1 will be used in the proof of the convergence of GM_AOS.

Lemma 3.2: *If the sequence $\left\{d_1^k\right\}, \left\{d_1^k\right\}, \ldots, \left\{d_l^k\right\}$ all converge to zero for a fixed integer l where $1 \leq l < n$, Then*

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

***Proof*:** Suppose, by the way of contradiction, that there exists a constant $\varepsilon > 0$ such that

$$\left(d_{l+1}^k\right)^2 \lambda_{l+1}^3 \geq \varepsilon \quad \text{and} \quad \left(d_{l+1}^k\right)^2 \lambda_{l+1}^2 \geq \varepsilon \tag{3.5}$$

for all $k \geq 0$. It follows from (3.2), (3.4) and the orthonormality of the eigenvectors $\{v_1, v_2, \ldots, v_n\}$ that

$$\overline{\alpha}_{k+1} = \tau_{k+1} \frac{\sum\limits_{i=1}^{n} \left(d_i^k\right)^2 \lambda_i^3}{\sum\limits_{i=1}^{n} \left(d_i^k\right)^2 \lambda_i^2} + (1 - \tau_{k+1}) \frac{\sum\limits_{i=1}^{n} \left(d_i^k\right)^2 \lambda_i^4}{\sum\limits_{i=1}^{n} \left(d_i^k\right)^2 \lambda_i^3}. \tag{3.6}$$

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

$$\sum_{i=1}^{l}\left(d_i^k\right)^2\lambda_i^2\le\frac{\varepsilon}{2}\qquad\text{and}\qquad\sum_{i=1}^{l}\left(d_i^k\right)^2\lambda_i^3\le\frac{\varepsilon}{2}. \tag{3.7}$$

for all $k\ge\widehat{k}$. By (3.6) and (3.7), we have, for any $k\ge\widehat{k}$,

$$\tau_{k+1}\frac{\lambda_{l+1}\sum\limits_{i=l+1}^{n}\left(d_i^k\right)^2\lambda_i^2}{\varepsilon/2+\sum\limits_{i=l+1}^{n}\left(d_i^k\right)^2\lambda_i^2}+(1-\tau_{k+1})\frac{\lambda_{l+1}\sum\limits_{i=l+1}^{n}\left(d_i^k\right)^2\lambda_i^3}{\varepsilon/2+\sum\limits_{i=l+1}^{n}\left(d_i^k\right)^2\lambda_i^3}\le\overline{\alpha}_{k+1}\le\lambda_n. \tag{3.8}$$

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 (3.8) that

$$\frac{2}{3}\lambda_{l+1}\le\overline{\alpha}_{k+1}\le\lambda_n \tag{3.9}$$

for all $k\ge\widehat{k}$, which means the bound $\left|1-\frac{\lambda_{l+1}}{\overline{\alpha}_k}\right|\le\max\left\{\frac{1}{2},1-\frac{\lambda_{l+1}}{\lambda_n}\right\}$ holds for all $k\ge\widehat{k}+1$. Finally, according to (3.9) and the first part of (3.3), we obtain, for all $k\ge\widehat{k}+1$,

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

where

$$\widehat{c}=\max\left\{\frac{1}{2},1-\frac{\lambda_{l+1}}{\lambda_n}\right\}. \tag{3.10}$$

This conclusion contradicts (3.5), therefore the lemma is true. The proof is completed. □

Theorem 3.1: *Let $f(x)$ be a strictly convex quadratic function, $\{x_k\}$ be the sequence generated by GM_AOS 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 $\lim\limits_{k\to\infty}\|e_k\|_2^2=0$ when $e_k=x^*-x_k\neq 0$ for all $k\ge 0$. It follows from (3.2) 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 $i=1,2,\dots,n$ converges to zero. Lemma 3.1 gives that $\{d_1^k\}$ converges to zero, and we only need to prove that $\{d_p^k\}$ converges to zero for $2\le p\le n$ by induction on p. For any $2<p\le n$, we assume that $\left\{d_1^k\right\},\left\{d_1^k\right\},\dots,\left\{d_{p-1}^k\right\}$ all tend to zero. Then for any given $\varepsilon>0$, there exists $\widehat{k}$

sufficiently large such that

$$\sum_{i=1}^{p-1}\left(d_i^k\right)^2\lambda_i^2 \le \frac{\varepsilon}{2} \quad \text{and} \quad \sum_{i=1}^{p-1}\left(d_i^k\right)^2\lambda_i^3 \le \frac{\varepsilon}{2}\lambda_p \tag{3.11}$$

hold for all $k \ge \widehat{k}$. We know from (3.6) and (3.11)

$$\tau_{k+1}\frac{\lambda_p\sum\limits_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^2}{\varepsilon/2+\sum\limits_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^2}+(1-\tau_{k+1})\frac{\lambda_p\sum\limits_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^3}{\lambda_p\varepsilon/2+\sum\limits_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^3} \le \overline{\alpha}_{k+1} \le \lambda_n \tag{3.12}$$

for all $k \ge \widehat{k}$. Moreover, Lemma 3.2 implies that there exists $k_p \ge \widehat{k}$ 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 $\left(d_p^{k_0}\right)^2\lambda_p^2 \ge \varepsilon$. Then $\left(d_p^{k_0-1}\right)^2\lambda_p^3 < \lambda_p\varepsilon$ and $\left(d_p^{k_0}\right)^2\lambda_p^3 \ge \lambda_p\varepsilon$. It is clear that

$$\sum_{i=p}^{n}\left(d_i^k\right)^2\lambda_i^2 \ge \left(d_p^k\right)^2\lambda_p^2 \ge \varepsilon \quad \text{and} \quad \sum_{i=p}^{n}\left(d_i^k\right)^2\lambda_p^3 \ge \left(d_p^k\right)^2\lambda_p^3 \ge \lambda_p\varepsilon, \tag{3.13}$$

hold for all $k_0 \le k \le j-1$. Here j is the first integer greater than k_0 such that $\left(d_p^{j-1}\right)^2\lambda_p^2 \ge \varepsilon$ and $\left(d_p^{j}\right)^2\lambda_p^2 < \varepsilon$. Therefore, $\left(d_p^{j-1}\right)^2\lambda_p^3 \ge \lambda_p\varepsilon$ and $\left(d_p^{j}\right)^2\lambda_p^3 < \lambda_p\varepsilon$. By (3.12) and (3.13), we have

$$\frac{2}{3}\lambda_{l+1} \le \overline{\alpha}_{k+1} \le \lambda_n \tag{3.14}$$

for all $k_0 \le k \le j-1$. Thus, by (3.14) and the first part of Equation (3.3), we obtain $\left|d_p^{k+2}\right| \le \widehat{c}\left|d_p^{k+1}\right|$ for all $k_0 \le k \le j-1$, where $\widehat{c}$ is the constant (3.10), which satisfies $\widehat{c} < 1$. Finally, using the bound $\left|d_p^{k_0+1}\right| \le \left(\frac{\lambda_n - \lambda_1}{\lambda_1}\right)^2\left|d_p^{k_0-1}\right|$ implied by (3.3) and the first part of Equation (3.3), we have

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

for all $k_0+1 \le k \le j+1$. Further, (3.3) implies the inequality $\left(d_p^{k_0}\right)^2 \le (\frac{\lambda_n - \lambda_1}{\lambda_1})^2\left(d_p^{k_0-1}\right)^2$. 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 $k \ge k_0-1$. Hence, since $\varepsilon > 0$ can be chosen arbitrarily small, we obtain $\lim_{k\to\infty}\left|d_p^k\right| = 0$ as required. Therefore,

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

□

According to (3.4), similar to the proof of Theorem 2.5 in [5] and the proof of the above theorem, we can obtain the R-linear convergence of GM_AOS.

Theorem 3.2: *Let $f(x)$ be a strictly convex quadratic function, $\{x_k\}$ be the sequence generated by GM_AOS and x^* be the unique minimizer of $f(x)$. Then, either $g_k = 0$ for some finite k, or the sequence $\{\|g_k\|_2\}$ converges to zero R-linearly.*

Remark: From Lemmas 3.1, 3.2 and Theorems 3.1, 3.2, we know that the method, 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 and R-linearly convergent for strictly convex quadratic minimization (2.1).

4. Numerical experiments

We compare GM_AOS with the MBB method, the ABB method, the AS method, the BB method, the AM method, the YM method and the conjugate gradient (CG) method. In GM_AOS, $\xi_k = 0.1$, $\mu_k = 0.2$, in the MBB method, $\xi_k = 0.2$, and in the ABB method, $\kappa = 0.5$. The exact stepsize α_0^{SD} is adopted at the first iteration for all methods. All methods are coded in Matlab, and the iteration is stopped if the inequality

$$\| g_k \|_2 \le \eta \| g_0 \|_2,$$

where $\eta > 0$, is satisfied or the number of iterations exceeds 10,000.

Three sets of test problems, which are used often, are chosen to examine the effectiveness of GM_AOS.

In the first set of test problems [2], each problem is generated by using Matlab commands, that is,

$$n = 1000, A = spandsym(n, 0.8, 1/condA, 1), x^* = -10 + 20 * rand(n, 1), \quad b = A * x^*.$$

where 'condA' denotes the condition number of the matrix A. The initial point is obtained by running Matlab command, that is, $x_0 = -10 + 20 * rand(n, 1)$.

Table 1 reports the average number of iterations required by the eight methods with 5 different starting points for the first set of problems. As shown in Table 1, GM_AOS outperforms the MBB method, the BB method, the AS method, the YM method and the AM method.

It is generally said that the ABB method is very efficient when high precision is required or the condition number $\kappa(A)$ is large. However, GM_AOS is superior to the ABB method for this set of problems, since GM_AOS requires less iterations than the ABB method for 91.67% problems (33 out of 36), and as the condition number $\kappa(A)$ is large or the required precision requirement increases, GM_AOS has distinct advantage over the ABB method. We also observe from Table 1 that when the precision is high or the condition number is neither too large nor too small, GM_AOS is on par with the CG method in the performance for the set of problems.

In the second set of problems [14], the matrix A with $n = 5000$ is the form of $A = QDQ^T$, where

$$Q = (I - 2\omega_3\omega_3^T)(I - 2\omega_2\omega_2^T)(I - 2\omega_1\omega_1^T),$$

ω_1, ω_2 and ω_3 are generated by Matlab commands, that is, $\omega_i = rand(n, 1), \quad \omega_i = \omega_i/norm(\omega_i), \quad i = 1, 2, 3$, and D is a diagonal matric generated by

$$D = diag(1, \sigma_2, \ldots, \sigma_{n-1}, condA), \quad \sigma_j = 1 + (condA - 1) * rand(1, n - 2), \quad j = 2, \ldots, n - 1.$$

The constant vector b are randomly generated by $b = -10 + 20 * rand(n, 1)$, and the starting point is set as $x_0 = (0, 0, \ldots, 0)^T \in R^n$.

Table 2 reports the number of iterations required by the eight methods for the second set of problems. From Table 2 we see that GM_AOS performs better than the other six gradient methods. As the condition number $\kappa(A)$ becomes large and a high precision is required, GM_AOS clearly requires much less iterations than the other gradient methods. GM_AOS is on par with the YM method in performance when the condition number $\kappa(A)$ is small or the required precision is low,

Table 1. The average number of iterations required by the eight methods with 5 different starting points for the first set of problems.

η	$\kappa(A)$	GM_AOS	MBB	BB	ABB	AS	YM	AM	CG
10^{-1}	10^1	4	4.6	5.6	5.6	5	3	3	3
	10^2	4.6	7	6	5.4	5	4	5	4
	10^3	**4**	6.6	6	5	5	4	5	**4**
	10^4	4.8	6.6	6.6	5.2	5	4	5	4
	10^5	4.6	6.6	6	5.2	5	4	5	4
	10^6	4.6	7	6.2	5.2	5	4	5	4
10^{-2}	10^1	8	11.6	9.6	9.6	7.4	7.2	7	7
	10^2	**14**	18.2	18	18.8	17.4	14	19.8	**15**
	10^3	**15.8**	20.2	19.4	18.2	19	17.8	21	**20.4**
	10^4	**16.2**	19.6	19.8	18.6	19.4	17.8	21.6	**21**
	10^5	**16.8**	19.8	20.4	20.4	20.8	17.8	21	**20.6**
	10^6	**15.8**	20.4	18.4	19.8	18.8	17.8	22.2	**19.8**
10^{-3}	10^1	13.2	15.2	12	12	12	11	13	10.2
	10^2	31.4	36	33	33	33.8	33.4	28.6	26.2
	10^3	**51.6**	59.4	51.2	58.8	61	69.2	55.8	**55.4**
	10^4	**57.4**	65.4	70	64	81	92.4	68.6	**90.6**
	10^5	**59.2**	61.2	59.6	67.8	69.8	76.4	64.2	**89.6**
	10^6	**61.6**	68.6	77.8	69	75	101.2	71.8	**92.2**
10^{-4}	10^1	15.8	18.6	16.2	16.2	17.4	15.6	19	14
	10^2	48.2	46.6	50	44.8	50.8	52	45	37.2
	10^3	104.4	114	115.8	109.4	123	213.2	138.8	91.2
	10^4	**180.6**	201.2	214.4	199.4	215.6	596.4	327.8	**198**
	10^5	**192.6**	204.8	229	236	215.4	686.8	319	**309.2**
	10^6	**188.6**	208.8	205.8	236	245.6	722.8	301.2	**325.6**
10^{-5}	10^1	20.2	21.4	20.8	20.8	22	20.4	25	17.8
	10^2	59.8	64	69.8	56.2	60.8	75.4	67.2	49
	10^3	150.8	153.8	170.2	153.6	174	378.8	213.8	125.8
	10^4	341.6	352.8	417	393	388.4	1897	609.4	301.6
	10^5	**610.8**	764.6	689	736	931.2	4975.2	1435.4	**654.4**
	10^6	**664.6**	855.6	845.8	976	831.4	4628.4	2144.2	**1107**
10^{-6}	10^1	24.8	25.8	25.2	25.2	27.4	24.4	31.4	21
	10^2	74.2	74.8	74.4	77.2	82.6	100.2	85.4	60.4
	10^3	206.2	222.8	230.4	211	225.2	597.2	306.2	161
	10^4	496.2	605.2	620.8	553.2	584.6	3756.4	1104.2	408.6
	10^5	1168.6	1514.6	1382.6	1393.2	1620.2	>10,000	4407.4	941.6
	10^6	**2040**	2623.4	2786.2	3227.6	3593.8	>10,000	8725.2	**2087.4**

and as the condition number $\kappa(A)$ becomes large and the required precision increases, GM_AOS shows great advantage over the YM method. GM_AOS is also competitive to the ABB method, since GM_AOS requires less iterations than the ABB method for 87.5% problems (35 out of 40), and when $\kappa(A) = 10^7$ and $\eta = 10^{-3}$, only GM_AOS and the ABB method can successfully solve the problem and GM_AOS requires much less iterations , when $\kappa(A) = 10^7$ and $\eta = 10^{-4}$, only GM_AOS can successfully solve the problem. The CG method is undoubtedly the most efficient approach. However, it seems not so appealing when a low precision is required.

In the third set of problems [14], the matrix $A \equiv \left(a_{i,j}\right)$ is given by

$$a_{i,i} = 2/h^2, a_{i,i-1} = -1/h^2 \ \text{ if } i \neq 1, \quad a_{i,i+1} = -1/h^2 \ \text{ if } i \neq n, \ \ i = 1, \ldots, n,$$

where $h = 11/n$ and $n = 1000$. The solution x^* and the initial point x_0 and b are generated by the way which is the same as that of the first set of problems.

The kind of problems often appear in the numerical solution of two-point boundary value problems. In Table 3 we report the average number of iterations required by the eight methods

Table 2. The number of iterations required by the eight methods for the second set of problems.

η	$\kappa(A)$	GM_AOS	MBB	BB	ABB	AS	YM	AM	CG
10^{-1}	10^1	5	6	6	6	6	4	5	4
	10^2	**9**	14	19	13	16	10	11	**10**
	10^3	**14**	22	25	19	23	12	15	**23**
	10^4	**14**	25	31	18	19	12	15	**47**
	10^5	**14**	20	20	18	27	16	15	**86**
	10^6	**14**	25	31	16	19	16	15	**57**
	10^7	**14**	21	20	17	19	12	15	**34**
10^{-2}	10^1	9	11	17	14	11	8	9	8
	10^2	23	27	26	29	27	26	23	22
	10^3	63	80	72	75	79	88	81	61
	10^4	166	189	178	195	248	326	337	179
	10^5	351	590	409	444	868	1102	1679	296
	10^6	801	1575	1058	1108	1404	>10,000	5449	251
	10^7	3359	6982	>10,000	4154	6416	>10,000	>10,000	435
10^{-3}	10^1	13	15	20	24	15	14	15	11
	10^2	40	50	47	40	56	46	51	34
	10^3	105	115	150	117	184	158	221	100
	10^4	256	288	251	346	316	602	549	238
	10^5	894	1312	1209	2557	2013	>10,000	3239	257
	10^6	1656	3234	3760	1917	4827	>10,000	>10,000	202
	10^7	4278	>10,000	>10,000	6020	>10,000	>10,000	>10,000	393
10^{-4}	10^1	16	19	22	21	19	18	20	15
	10^2	56	65	72	56	55	76	67	45
	10^3	166	195	179	175	229	294	315	132
	10^4	429	477	474	703	572	2942	871	280
	10^5	1397	1674	1904	1239	1974	>10,000	>10,000	486
	10^6	3727	5979	7306	8680	6436	>10,000	>10,000	347
	10^7	7103	>10,000	>10,000	>10,000	>10,000	>10,000	>10,000	403
10^{-5}	10^1	21	22	24	26	21	22	27	18
	10^2	71	70	76	62	90	76	89	56
	10^3	195	207	243	208	274	354	275	150
	10^4	534	816	839	531	674	2348	2553	228
	10^5	1879	3080	3151	1995	2965	>10,000	>10,000	428
	10^6	3796	5985	6043	5242	7451	>10,000	>10,000	500
	10^7	>10,000	>10,000	>10,000	>10,000	>10,000	>10,000	>10,000	430
10^{-6}	10^1	25	27	26	27	25	26	33	22
	10^2	81	80	82	87	91	94	95	67
	10^3	284	248	289	239	339	401	470	206
	10^4	774	858	999	930	829	3430	1649	349
	10^5	1979	3009	2605	2313	1290	>10,000	>10,000	388
	10^6	4980	>10,000	9199	4509	>10,000	>10,000	>10,000	459
	10^7	>10,000	>10,000	>10,000	>10,000	>10,000	>10,000	>10,000	633

with 5 different starting points for the third set of problems . As shown in Table 3, GM_AOS is superior to the MBB method, the BB method, the ABB method, the AS method, the AM method and the YM method , and GM_AOS illustrates distinct advantage as the required precision increases. GM_AOS also performs slightly better than the CG method for this set of problems.

From Tables 1–3, we see that for the three sets of problems (84 problems) GM_AOS successfully solves 82 problems, while MBB, BB, ABB, AS, YM and AM successfully solve 79, 79, 81, 79, 68 and 72 problems, respectively.

Since our main concern focus on the performance of GM_AOS and the other gradient methods, the CG method will be excluded in the following analysis. We adopt the performance profile [15] introduced by Dolan and More to illustrate the whole performance of these gradient methods for the

Table 3. The average number of iterations required by the eight methods with 5 different starting points for the third set of problems.

θ	GM_AOS	MBB	BB	ABB	AS	YM	AM	CG
10^{-2}	8.4	11	9	9.2	10	8	9	8.8
10^{-3}	**24.8**	27.8	28.4	27.8	29.2	25.8	23.2	**27.6**
10^{-4}	**67.6**	78.4	81	77.2	88	88.4	77.8	**91.4**
10^{-5}	**180.2**	188.2	198.4	213	227.8	457	240.6	**260**
10^{-6}	**370.6**	436	403.6	437	445.8	1758.8	761.8	**658.6**
10^{-7}	1106	1379	1370.4	1425.2	1604.4	8828.4	3522.6	952.4

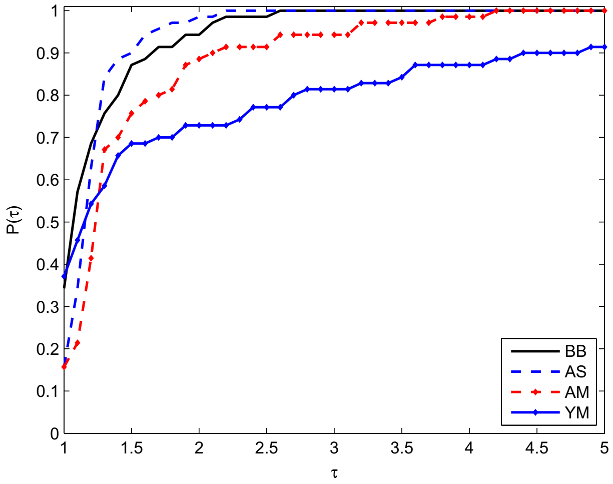

Figure 2. Performance profile based on the number of iterations of the BB, AS, AM and YM methods for the three sets of problems.

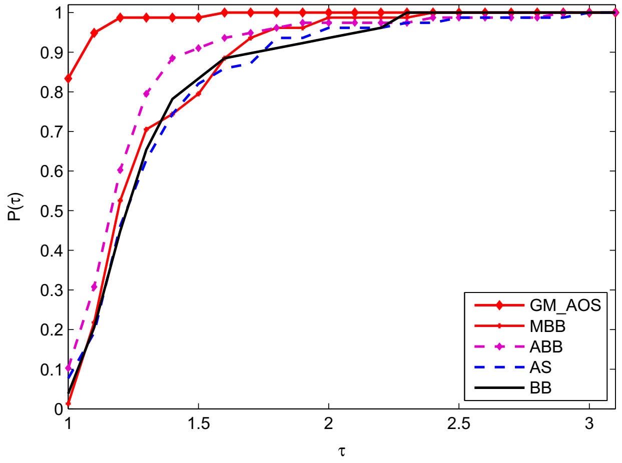

Figure 3. Performance profile based on the number of iterations of the MBB, ABB, AS, BB methods and GM_AOS for the three sets of problems.

three sets of problems based on Tables 1-3. We first compare the performance of the BB method, the AS method, the YM method and the AM method, and select the two best to compare the performance of GM_AOS, the MBB method and the ABB method.

After eliminating those problems for which there exists at least one of the four methods tested whose iterations exceed 10,000, 68 problems are left, and we only consider the performance of the remaining 68 problems in the following analysis. As shown in Figure 2, we observed that the AS method and the BB method are competitive to the AM method and the YM method relative to the number of iterations. As a result, the AS method and the BB method are selected to compare with GM_AOS, the MBB method and the ABB method.

After eliminating those problems for which there exists at least one of the five methods tested whose iterations exceed 10,000, 78 problems are left, and we only consider the performance of the remaining 78 problems in the following analysis. We see from Figure 3 that GM_AOS has a great advantage over the MBB method, the ABB method, the AS method and the BB method, since with the least number of iterations GM_AOS successfully solves about 83% problems, while the percentages of the ABB method, the AS method, the BB method and the MBB method are about 10, 8, 5 and 3% problems, respectively.

It indicates that GM_AOS is superior to the other gradient methods for the three sets of test problems.

5. Conclusion and discussion

In this paper we introduce a new type of stepsize, approximate optimal stepsize, to interpret the BB method, and present an efficient gradient method with an approximate optimal stepsize. The global convergence and R-linear convergence of GM_AOS are established. Numerical results show that GM_AOS is very efficient. It seems that neither of a exact stepsize and very inexact stepsize is efficient for gradient method, and a stepsize with a high precision is preferable for gradient method. This explains why GM_AOS can illustrate good numerical behaviors. Different approximation models lead to different approximate optimal stepsizes and different approximate optimal stepsizes correspond to different gradient methods. Together with the nice numerical performance of GM_AOS, we call the gradient method with an approximate optimal stepsize or its variant as approximate optimal gradient method. The approximate optimal gradient methods will have vaster foreground for general unconstrained optimization problems and should be paid more attentions.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This research is supported by National Science Foundation of China [grant number 11461021], [grant number 11601012]; National Science Foundation of Shangxi [grant number 2017JM1014]; National Science Foundation of Guangxi [grant number 2014GXNSFAA118028], [grant number 2015GXNSFAA139011]; Scientific Research Foundation of Guangxi Education Department [grant number 2013YB236]; Scientific Research Project of Hezhou University [grant number 2014YBZK06], [grant number 2016HZXYSX03]. Guangxi Colleges and Universities Key Laboratory of Symbolic Computation and Engineering Data Processing (FH201701).

References

[1] Cauchy A. Méthode générale pour la résolution des systéms déquations simultanées. Comp Rend Sci Paris. 1847;25:46–89.

[2] Asmundis RD, Serafino DD, Riccio F, Toraldo G. On spectral properties of steepest descent methods. IMA J Numer Anal. 2013;33(4):1416–1435.

[3] Barzilai J, Borwein JM. Two-point step size gradient methods. IMA J Numer Anal. 1988;8(1):141–148.

[4] Raydan M. On the Barzilai and Borwein choice of steplength for the gradient method. IMA J Numer Anal. 1993;13(3):321–326.

[5] Dai YH, Liao LZ. R-linear convergence of the Barzilai and Borwein gradient method. IMA J Numer Anal. 2002;22(1):1–10.

[6] Dai YH. Alternate step gradient method. Optimization. 2010;52(4–5):395–415.

[7] Dai YH, Yuan YX. Alternate minimization gradient method. IMA J Numer Anal. 2003;23(3):377–393.

[8] Zhou B, Gao L, Dai YH. Gradient methods with adaptive stepsizes. Comput Optim Appl. 2006;35(1):69–86.

[9] Yuan YX. A new stepsize for the steepest descent method. J Comput Math. 2006;24:149–156.

[10] Liu HW, Liu ZX, Dong XL. A new adaptive Barzilai and Borwein method for unconstrained optimization. Optim Lett. 2017. DOI:10.1007/s11590-017-1150-9

[11] Zheng YT, Zheng B. A New modified Barzilai-Borwein gradient method for the quadratic minimization problem. J Optim Theory Appl. 2017;172(1):179–186.

[12] Ford J, Moghrabi I. Multi-step quasi-Newton methods for optimization. J Comput Math. 1994;50(93):305–323.

[13] Yuan YX, Sun WY. Theory and methods of optimization. Science Press of China; 1999.

[14] Friedlander A, Martinez JM, Molina B, Raydan M. Gradient method with retards and generalizations. SIAM J Numer Anal. 1999;36:275–289.

[15] Dolan ED, More JJ. Benchmarking optimization software with performance profiles. Math Program. 2002;91:201–213.