



# New stepsizes for the gradient method

Cong Sun<sup>1</sup> · Jin-Peng Liu<sup>2,3</sup>

Received: 15 January 2018 / Accepted: 22 November 2019  
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

## Abstract

Gradient methods are famous for their simplicity and low complexity, which attract more and more attention for large scale optimization problems. A good stepsize plays an important role to construct an efficient gradient method. This paper proposes a new framework to generate stepsizes for gradient methods applied to convex quadratic function minimization problems. By adopting different criterions, we propose four new gradient methods. For 2-dimensional unconstrained problems with convex quadratic objective functions, we prove that the new methods either terminate in finite iterations or converge R-superlinearly; for  $n$ -dimensional problems, we prove that all the new methods converge R-linearly. Numerical experiments show that the new methods enjoy lower complexity and outperform the existing gradient methods.

**Keywords** Gradient method · Steepest descent · R-linear convergence rate · Finite termination

## 1 Introduction

The gradient method plays an important role in solving large scale optimization problems. It is widely used in many applications, e.g., machine learning and inverse problems [24]. Thus it is necessary to exploit efficient stepsizes for gradient methods and their theoretical analysis.

---

This work is partially supported by NSFC Grants 11771056, 91630202 and 11871115, and the Young Elite Scientists Sponsorship Program by CAST 2017QNRC001.

---

✉ Cong Sun  
suncong86@bupt.edu.cn  
Jin-Peng Liu  
jliu1219@terpmail.umd.edu

<sup>1</sup> School of Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

<sup>2</sup> School of Mathematics and Systems Science, Beihang University, Beijing, China

<sup>3</sup> Present Address: Department of Mathematics, University of Maryland, College Park, MD 20742, USA

Consider the unconstrained convex quadratic optimization problem:

$$\min_{x \in \mathbb{R}^n} f(x) = b^T x + \frac{1}{2} x^T H x, \quad (1)$$

where  $b \in \mathbb{R}^n$ , and  $H \in \mathbb{R}^{n \times n}$  is symmetric positive definite. We want to find the minimal value of  $f(x)$  by means of gradient methods. Let  $x_k$  be the iterate point achieved from the  $k$ th iteration, and  $g_k = \nabla f(x_k) = Hx_k + b$  be the gradient vector at  $x_k$ . The iteration formulation of the gradient method is:

$$x_{k+1} = x_k - \alpha_k g_k. \quad (2)$$

Different choices of the stepsize  $\alpha_k$  lead to various performances. One particular choice is the exact linesearch, which is also called Cauchy stepsize [2,3]:

$$\alpha_k^* = \operatorname{argmin}_{\alpha > 0} f(x_k - \alpha g_k).$$

It has the closed form as:

$$\alpha_k^* = \frac{\|g_k\|_2^2}{g_k^T H g_k}.$$

By using exact linesearch steps, the gradient method becomes the steepest descent method. It has linear convergence rate [23], but suffers from the zigzag phenomenon.

Another choice of stepsize was proposed by Barzilai and Borwein [1]. The BB stepsizes are designed as the approximation of the inverse of the Hessian matrix, and have certain quasi-Newton property. This provides two BB stepsizes as:

$$\begin{aligned} \alpha_k^{BB1} &= \frac{s_k^T s_k}{s_k^T y_k} = \operatorname{argmin}_{\alpha} \|y_k - \alpha^{-1} s_k\|^2, \\ \alpha_k^{BB2} &= \frac{s_k^T y_k}{y_k^T y_k} = \operatorname{argmin}_{\alpha} \|s_k - \alpha y_k\|^2, \end{aligned}$$

where  $s_k = x_k - x_{k-1}$ ,  $y_k = g_k - g_{k-1}$ . For 2-dimensional convex quadratic objective functions, Barzilai and Borwein [1] showed that the gradient method (2) with the BB stepsize  $\alpha_k^{BB1}$  converges R-superlinearly and the R-order is  $\sqrt{2}$ . Its R-linear convergence rate for general  $n$ -dimensional problems is proved in [8].

This result has inspired many studies on the gradient method [4–6,8,11,15,17–20,25]. For example, Dai [4] suggested to use an alternate step (AS) method, which combines the Cauchy stepsize with the BB stepsize:

$$\alpha_k^{AS} = \begin{cases} \alpha_k^*, & \text{mod}(k, 2) = 1; \\ \alpha_k^{BB}, & \text{mod}(k, 2) = 0. \end{cases}$$

Yuan [21] proposed a new kind of stepsize (Yuan stepsize) which ensures 3-step termination for minimizing 2-dimensional convex quadratic objective functions:

$$\alpha_k = \begin{cases} \alpha_k^Y = 2 \left( \sqrt{\left( \frac{1}{\alpha_{k-1}^*} - \frac{1}{\alpha_k^*} \right)^2 + \frac{4\|g_k\|_2^2}{(\|s_{k-1}\|_2)^2}} + \frac{1}{\alpha_{k-1}^*} + \frac{1}{\alpha_k^*} \right)^{-1}, & \text{mod}(k, 4) = 0; \\ \alpha_k^*, & \text{otherwise.} \end{cases} \quad (3)$$

Later on, Dai and Yuan [7] proposed a variant strategy of Yuan stepsize, which improves the aforementioned method and keeps the finite termination property for 2-dimensional convex quadratic function minimization:

$$\alpha_k^{DY} = \begin{cases} \alpha_k^*, & \text{mod}(k, 4) = 1, 2; \\ \alpha_k^{YV} = 2 \left( \sqrt{\left( \frac{1}{\alpha_{k-1}^*} - \frac{1}{\alpha_k^*} \right)^2 + \frac{4\|g_k\|_2^2}{(\alpha_{k-1}^* \|g_{k-1}\|_2)^2}} + \frac{1}{\alpha_{k-1}^*} + \frac{1}{\alpha_k^*} \right)^{-1}, & \text{mod}(k, 4) = 0, 3. \end{cases} \quad (4)$$

In the framework of (3),  $\alpha_k^Y$  is equivalent to  $\alpha_k^{YV}$ . But  $\alpha_k^{YV}$  in (4) is different from  $\alpha_k^Y$ . The new method (4), short as DY method, performs very competitive with the BB method for large scale problems. Due to the numerical experiences, Yuan [22] believed that “a good gradient method would use at least one exact linesearch in every few iterations”. Based on this idea, a number of gradient methods have been discussed. Recent work by De Asmundis et al. [10] modified the DY method and proposed the SDC gradient method. In one loop, it consists of  $h + l$  inner iterations, where the first  $h$  iterations ( $h \geq 2$ ) apply the Cauchy stepsizes, and the rest  $l$  iterations ( $l \geq 1$ ) use the fixed steplengths generated from Yuan step. The detailed stepsize is updated as follows:

$$\alpha_k^{SDC} = \begin{cases} \alpha_k^*, & \text{mod}(k, h + l) < h; \\ \alpha_k^{YV}, & \text{mod}(k, h + l) = h; \\ \alpha_{k-1}, & \text{otherwise.} \end{cases} \quad (5)$$

The work [10] also exploited the asymptotic spectral behaviour of the DY method, and proved that the SDC algorithm has R-linear convergence rate for minimizing convex quadratic functions. The SDC method outperforms the DY method with proper parameters  $h$  and  $l$ .

In [12], di Serafino et al. analyzed several gradient methods, including the SDC method, Adaptive BB method [14], Limited Memory Steepest Descent (LMSD) method [13] and so on. The methods were also extended to general unconstrained problems. It showed that the gradient methods with groups of short steps followed by several long steps are efficient for strongly convex quadratic problems. Similar analysis for the SD method was presented in [16].

Inspired by the SDC method, we propose new gradient methods by modifying (5). The modification mainly include two aspects. On one hand, the basic framework is improved, and only two exact linesearch steps are required in a loop. On the other hand, several new stepsizes are plugged in the new framework. For 2-dimensional problems, we prove that the corresponding gradient methods either terminate in finite iterations or converge R-superlinearly. For general  $n$ -dimensional problems ( $n > 2$ ), we prove

that all the proposed gradient methods converge R-linearly. The rest of the paper is organized as follows. The new methods are introduced in Sect. 2. The convergence analysis is shown in Sect. 3. In Sect. 4, numerical results show that our new gradient algorithms outperform the state of the art.

## 2 New gradient methods

In this section, new gradient methods are proposed.

### 2.1 New framework

Consider problem (1). We construct new gradient methods by proposing new stepsizes  $\alpha_k$ . First, a cyclic framework that combines Cauchy steps with fixed steplengths is proposed as follows:

$$\alpha_k = \begin{cases} \alpha_k^*, & \text{if } \text{mod}(k, m) < 2; \\ \alpha_{k-1}^F, & \text{if } \text{mod}(k, m) = 2; \\ \alpha_{k-1}, & \text{otherwise.} \end{cases} \quad (6)$$

Here the stepsize  $\alpha_k^F$  has different choices, which is analyzed in the next subsection. In each loop, we take 2 Cauchy steps followed by  $m - 2$  gradient steps with fixed stepsizes, where the fixed stepsize is computed according to the 2 Cauchy steps.

Such framework is different from the SDC method [10], in the following two aspects. First, the new framework only requires 2 exact linesearch steps, while (5) requires  $h + 1$ . Thus the computational cost is reduced compared to SDC. Second, the key stepsize  $\alpha_{k-1}^F$  is constructed by the information of the last  $((k - 1)\text{th})$  iteration rather than the current  $(k\text{th})$  iteration, which makes the generated iterate points completely different from SDC.

### 2.2 New stepsizes

In the new framework, we apply various types of stepsizes. Here we consider four different choices of  $\alpha_k^F$  as follows:

$$\alpha_k^{F(1)} = \alpha_k^{YV} \quad (7)$$

$$\alpha_k^{F(2)} = \tilde{\alpha}_k = \left( \frac{1}{\alpha_{k-1}^*} + \frac{1}{\alpha_k^*} \right)^{-1}, \quad (8)$$

$$\alpha_k^{F(3)} = \alpha_k^{Fmin} = \min\{\alpha_{k-1}^*, \alpha_k^*\}, \quad (9)$$

$$\alpha_k^{F(4)} = \alpha_k^{Fmax} = \max\{\alpha_{k-1}^*, \alpha_k^*\}. \quad (10)$$

The stepsizes (7) and (8) were proposed in [7,9], respectively. As analyzed in [10], the stepsizes (7) and (8) approximate the inverse of the largest eigenvalue and that of

the sum of the largest and smallest eigenvalues of the Hessian matrix, respectively. If the approximation is sufficiently accurate, we can reduce the gradient component corresponding to the largest eigenvalue significantly. In this case the problem is reduced into a lower dimensional problem. Hopefully the dimension reduction happens frequently. The stepsizes (9) and (10) are proposed in this paper, with the purpose to approximate the inverse eigenvalues of the Hessian matrix  $H$  on the 2-dimensional subspace spanned by  $g_k$  and  $g_{k-1}$ . Let  $v_k = \frac{g_k}{\|g_k\|}$  and  $v_{k-1} = \frac{g_{k-1}}{\|g_{k-1}\|}$ . Then the Hessian matrix  $H$  on the subspace  $\mathcal{S} = \text{Span}\{g_k, g_{k-1}\}$  is expressed as

$$H_{\mathcal{S}} = \begin{pmatrix} v_k^T H v_k & v_k^T H v_{k-1} \\ v_{k-1}^T H v_k & v_{k-1}^T H v_{k-1} \end{pmatrix} = \begin{pmatrix} \frac{1}{\alpha_k^*} & v_k^T H v_{k-1} \\ v_{k-1}^T H v_k & \frac{1}{\alpha_{k-1}^*} \end{pmatrix}.$$

The eigenvalues of  $H_{\mathcal{S}}$  can be approximated by the diagonal elements (9) and (10). Compared to the stepsizes (7) and (8), the computational cost of (9) and (10) are lower. In summary, all these four stepsizes are designed to approximate the inverse of the eigenvalues of the Hessian matrix  $H$ ; the new framework uses the fewest Cauchy steps to generate the aforementioned four stepsizes, and combines them to reduce the problem into lower and lower dimensional subspaces.

It is easy to show the following relations among the four stepsizes:

$$\tilde{\alpha}_k \leq \alpha_k^{YV} \leq \alpha_k^{Fmin} \leq \alpha_k^{Fmax}. \quad (11)$$

Let  $\lambda_1 \geq \dots \lambda_n > 0$  be the eigenvalues of the Hessian matrix  $H$ . Then the stepsizes  $\alpha_k^{YV}$ ,  $\alpha_k^{Fmin}$  and  $\alpha_k^{Fmax}$  are bounded by the inverse of the eigenvalues:

$$\lambda_1^{-1} \leq \alpha_k^{YV} \leq \alpha_k^{Fmin} \leq \alpha_k^{Fmax} \leq \lambda_n^{-1}. \quad (12)$$

This can be verified by the fact that  $\lambda_1^{-1} \leq \alpha_k^* \leq \lambda_n^{-1}$  holds for all  $k$ . Yet  $\tilde{\alpha}_k$  is possibly smaller than  $\lambda_1^{-1}$ .

By using (7)–(10), the gradient method with stepsizes (6) gives four different algorithms. We call the new gradient methods as Alg. 1 to Alg. 4 corresponding to (7)–(10), respectively. The detailed framework of Alg. 1 is shown in Algorithm 1, and the frameworks for the other three algorithms are similar.

**Input** : Initial point  $x_0$ ;  $k = 0$ ; parameter  $m$ ; the stopping parameter  $\epsilon$ .  
**Output**:  $x_k$   
**repeat**  
    1. If  $\text{mod}(k, m) < 2$ , let  $\alpha_k = \alpha_k^*$  and go to Step 4;  
    2. If  $\text{mod}(k, m) = 2$ , let  $\alpha_k = \alpha_{k-1}^{YV}$  and go to Step 4;  
    3.  $\alpha_k = \alpha_{k-1}$ ;  
    4. Update the iteration point:  $x_{k+1} = x_k - \alpha_k g_k$ ;  $k := k + 1$ ;  
**until**  $\|g_k\| < \epsilon$ ;

**Algorithm 1:** Algorithm 1

The proposed algorithms make full use of the computed Cauchy steps. In one loop of each proposed algorithm, we only compute two Cauchy steps in the first two iterations, and then easily construct the fixed step length  $\alpha_{k-1}^F$  with simple calculation by using the two Cauchy steps. In this way, the computational cost is reduced compared to the state of the art such as SDC.

### 3 Theoretical analysis

In this section we analyze the convergence properties of the new gradient algorithms proposed in Sect. 2. The analysis for 2-dimensional cases and general  $n$  ( $n > 2$ ) dimensional cases are considered individually.

#### 3.1 Convergence analysis for 2-dimensional problems

Assume that the four proposed gradient algorithms are applied to problem (1) with  $n = 2$ . We consider the special 2-dimensional case to show that the proposed algorithms are able to avoid the zigzag phenomenon caused by Cauchy steps.

First, we show that Alg. 1 has finite termination property.

**Theorem 1** *Apply the gradient method Alg. 1 to problem (1) with  $n = 2$ .  $x_0$  is any initial point. Suppose that  $\lambda_1 = \lambda > \lambda_2 > 0$  are the two eigenvalues of the Hessian matrix  $H$  of the objective function, and  $d_1$  and  $d_2$  are the corresponding orthonormal eigenvectors. The gradient vector  $g_k = Hx_k + b$  can be represented as*

$$g_k = \mu_1^k d_1 + \mu_2^k d_2$$

with  $\mu_i^k = d_i^T g_k$  ( $i = 1, 2$ ). If the initial point  $x_0$  satisfies that  $\mu_1^0 \neq 0$  and  $\mu_2^0 \neq 0$ , the method terminates in finite iterations.

**Proof** Without loss of generality we assume that  $d_1 = (1, 0)^T$ ,  $d_2 = (0, 1)^T$  and  $\lambda_2 = 1$ . In this case,  $g_k = (\mu_1^k, \mu_2^k)^T$  and

$$H = \begin{pmatrix} \lambda & 0 \\ 0 & 1 \end{pmatrix}.$$

Suppose that  $\mu_1^k \neq 0$ . Otherwise the problem degenerates into one-dimensional subspace, and the algorithm will terminate after the next Cauchy step. Define

$$q_k = \frac{(\mu_2^k)^2}{(\mu_1^k)^2}. \quad (13)$$

Since  $g_{k+1} = (I - \alpha_k H)g_k$ , we can establish that

$$\mu_1^{k+1} = (1 - \lambda\alpha_k)\mu_1^k, \quad \mu_2^{k+1} = (1 - \alpha_k)\mu_2^k \quad (14)$$

and

$$\|g_{k+1}\|_2^2 = \frac{1+q_{k+1}}{1+q_k} (1-\lambda\alpha_k)^2 \|g_k\|_2^2. \quad (15)$$

For Cauchy steps, we have

$$\alpha_k^* = \frac{\|g_k\|_2^2}{g_k^T H g_k} = \frac{1+q_k}{\lambda+q_k}$$

and

$$q_{k+1} = \frac{1}{q_k}. \quad (16)$$

From the update strategy (6), we have  $\alpha_0 = \alpha_0^* = \frac{1+q_0}{\lambda+q_0}$  and  $\alpha_1 = \alpha_1^* = \frac{1+q_1}{\lambda+q_1} = \frac{1+q_0}{\lambda q_0 + 1}$ . Introducing

$$\begin{aligned} \tilde{\alpha}_1 &= \left( \frac{1}{\alpha_0^*} + \frac{1}{\alpha_1^*} \right)^{-1} = \left( \frac{\lambda+q_0}{1+q_0} + \frac{\lambda q_0 + 1}{1+q_0} \right)^{-1} = \frac{1}{\lambda+1}, \\ \rho_1 &= \frac{1}{\alpha_0^* \alpha_1^*} - \frac{\|g_1\|_2^2}{(\alpha_0^* \|g_0\|_2)^2} = \frac{(\lambda+q_0)(\lambda q_0 + 1)}{(1+q_0)^2} - \frac{q_0(\lambda-1)^2}{(1+q_0)^2} = \lambda, \end{aligned}$$

we can deduce that

$$\alpha_2 = \alpha_1^{YV} = 2(\sqrt{(\lambda+1)^2 - 4\lambda} + \lambda + 1)^{-1} = \frac{1}{\lambda}.$$

Then (14) implies that  $\mu_1^3 = 0$ . Consequently  $\mu_1^k$  keeps 0 for  $k \geq 3$ . The gradient residue  $\mu_2^k$  can be further eliminated by the exact linesearch in the  $(m+1)$ th iteration. Thus Alg. 1 terminates in at most  $m+1$  iterations.  $\square$

Next the other three proposed gradient algorithms are analyzed.

**Theorem 2** Apply Alg. 2, Alg. 3 and Alg. 4 to problem (1) with  $n = 2$ . Under the assumptions in Theorem 1, the three methods converges  $R$ -superlinearly.

**Proof** In all the three methods,  $\alpha_{mk} = \alpha_{mk}^*$ ,  $\alpha_{mk+1} = \alpha_{mk+1}^*$  and  $q_{mk+1} = q_{mk}^{-1}$ . For Alg. 2, due to (16), we have

$$\alpha_{mk+i} = \tilde{\alpha}_{mk+1} = \left( \frac{1}{\alpha_{mk}^*} + \frac{1}{\alpha_{mk+1}^*} \right)^{-1} = \frac{1}{\lambda+1}, i = 2, \dots, m-1.$$

From (13) and (14), we can deduce that

$$q_{mk+i} = \begin{cases} \frac{1}{q_{mk+i-1}}, & \text{if } i = 1, 2; \\ \lambda^2 q_{mk+i-1}, & \text{if } 2 < i \leq m \end{cases} \quad (17)$$

and thus conclude that

$$q_{mk} = \lambda^{(2m-4)k} q_0. \quad (18)$$

Taking (17) into (15), we have

$$\|g_{mk+m}\|_2^2 = \frac{1 + \lambda^{2m-4} q_{mk}}{1 + q_{mk}} \frac{(\lambda - 1)^4}{(\lambda + 1)^{2m-4} (\lambda + q_{mk}^{-1})^2 (\lambda + q_{mk})^2} \|g_{mk}\|_2^2.$$

For simplicity, define  $q = q_{mk}$ . From (18) we have  $q > q_0$ , with  $\lambda > 1$  and  $m > 2$ .

From  $q_{mk+1} = \lambda^{2m-4} q = \frac{q^2}{q_0}$ , we have

$$\begin{aligned} & \frac{1 + \lambda^{2m-4} q}{1 + q} \cdot \frac{1}{(\lambda + q^{-1})^2 (\lambda + q)^2} \\ &= \frac{q_0 + q^2}{q_0(1 + q)} \cdot \frac{1}{[\lambda^2 + \lambda(q^{-1} + q) + 1]^2} \\ &\leq \frac{q_0 + q^2}{q_0(1 + q)} \cdot \frac{1}{\lambda^2(q_0 + q^2)} \leq \lambda^{-2} q_0^{-1} q^{-1}. \end{aligned}$$

Thus

$$\|g_{mk+m}\|_2^2 \leq C_1 q_0^{-2} \lambda^{-(2m-4)k} \|g_{mk}\|_2^2 \leq C_1^k q_0^{-2k} \lambda^{-(m-2)(k^2+k)} \|g_0\|_2^2,$$

where  $C_1(\lambda, m) = \frac{(\lambda-1)^4}{\lambda^2(\lambda+1)^{2m-4}}$  is a constant. When  $k$  goes to infinity,

$$\sqrt[k]{\|g_{mk+m}\|_2} \leq \sqrt[k]{C_1} \sqrt[k]{\|g_0\|_2} q_0^{-1} \lambda^{-\frac{m-2}{2}(k+1)} \rightarrow 0.$$

The above analysis implies that Alg. 2 converges R-superlinearly.

Similarly, we can prove that Alg. 3 and Alg. 4 converges R-superlinearly.  $\square$

### 3.2 Convergence analysis for $n$ -dimensional problems

Besides the convergence results for 2-dimensional problems, we also analyze the convergence property of the proposed gradient algorithms for general  $n$ -dimensional problems.

**Lemma 1** Suppose  $d_1, \dots, d_n$  are the orthonormal eigenvectors corresponding to  $\lambda_1, \dots, \lambda_n$ . The proposed gradient methods are applied to problem (1). Suppose that the initial point  $x_0$  satisfies  $g_0^T d_1 \neq 0$  and  $g_0^T d_n \neq 0$ . Let  $g_k = \sum_{i=1}^n \mu_i^k d_i$  and define  $h(k, l) = \sum_{i=l}^n (\mu_i^k)^2$ . Then all the proposed gradient methods satisfy the following two properties:

- (i) there exists a constant  $M_1 \geq \lambda_n$ , such that  $\lambda_n \leq \alpha_k^{-1} \leq M_1$  holds for all  $k \geq 0$ ;
- (ii) if there exists a positive integer  $k_0$  and a positive constant  $M_2$ , such that  $h(k - j, l) < \epsilon$  and  $(\mu_{l-1}^{k-j})^2 < M_2 \epsilon$  holds for any  $j \in \{0, \dots, \min\{k, k_0\}\}$ , then  $\alpha_k^{-1} \geq \frac{2}{3} \lambda_{l-1}$ .



**Proof** First, we prove that all the four proposed algorithms satisfy property (i). By the framework (6), we only need to prove that both the Cauchy steps and  $\alpha_k^F$  in each method satisfy property (i).

Since  $\alpha_k^* = \frac{g_k^T g_k}{g_k^T H g_k}$ , it is trivial to show that

$$\lambda_n \leq (\alpha_k^*)^{-1} \leq \lambda_1. \quad (19)$$

For Alg. 1 and Alg. 2, we have from (11) that

$$\lambda_1 \leq \min\{(\alpha_k^*)^{-1}, (\alpha_{k-1}^*)^{-1}\} \leq (\alpha_k^{YV})^{-1} \leq (\tilde{\alpha}_k)^{-1} = \alpha_k^{-1} + \alpha_{k-1}^{-1} \leq 2\lambda_n.$$

Let  $M_1 = 2\lambda_n$ . We conclude that both algorithms satisfy property (i).

For Alg. 3 and Alg. 4, (19) holds, thus

$$\lambda_n \leq \alpha_k^{Fmin} = \min\{(\alpha_k^*)^{-1}, (\alpha_{k-1}^*)^{-1}\} \leq \alpha_k^{Fmax} = \max\{(\alpha_k^*)^{-1}, (\alpha_{k-1}^*)^{-1}\} \leq \lambda_1.$$

Let  $M_1 = \lambda_1$ , and property (i) holds for both methods.

Next, we prove that property (ii) holds for all the four proposed algorithms.

For Alg. 1, Alg. 2, and Alg. 3, from (11), we have that

$$\tilde{\alpha}_k \leq \alpha_k^{YV} \leq \alpha_k^{Fmin} \leq \alpha_k^*.$$

For the considered three algorithms, given  $k$ , if  $\text{mod}(k, m) \geq 2$ , then  $\alpha_k = \alpha_{r+1} = \alpha_r^F \leq \alpha_r^*$  holds, where  $r = \text{argmax}\{i \leq k | \alpha_i = \alpha_i^*\}$  is the iteration number of the last Cauchy step during all the  $k$  iterations. Thus  $\alpha_k^{-1} \geq (\alpha_r^*)^{-1}$ . Let  $k_0 = m - 2$  and  $M_2 = 2$ . Then  $\max\{k - k_0, 0\} \leq r \leq k$ . We can deduce that

$$\begin{aligned} (\alpha_r^*)^{-1} &= \frac{g_r^T H g_r}{g_r^T g_r} = \frac{\sum_{i=1}^n (\mu_i^r)^2 \lambda_i}{\sum_{i=1}^n (\mu_i^r)^2} \geq \frac{\sum_{i=1}^{l-1} (\mu_i^r)^2 \lambda_i}{\sum_{i=1}^{l-1} (\mu_i^r)^2 + h(r, l)} \\ &\geq \frac{\lambda_{l-1} \sum_{i=1}^{l-1} (\mu_i^r)^2}{\sum_{i=1}^{l-1} (\mu_i^r)^2 + \epsilon} = \frac{\lambda_{l-1}}{1 + \frac{\epsilon}{\sum_{i=1}^{l-1} (\mu_i^r)^2}} \\ &\geq \frac{\lambda_{l-1}}{1 + \frac{\epsilon}{(\mu_{l-1}^r)^2}} \geq \frac{\lambda_{l-1}}{1 + \frac{\epsilon}{2\epsilon}} = \frac{2}{3} \lambda_{l-1}. \end{aligned} \quad (20)$$

Thus  $\alpha_k^{-1} \geq (\alpha_r^*)^{-1} \geq \frac{2}{3} \lambda_{l-1}$ . The considered three algorithms satisfy property (ii).

For Alg. 4, we have

$$(\alpha_k^{Fmax})^{-1} \geq \min\{(\alpha_k^*)^{-1}, (\alpha_{k-1}^*)^{-1}\}.$$

Given  $k$ ,  $\alpha_k^{-1} \geq (\alpha_{r_0}^*)^{-1}$  holds, where

$$r_0 = \begin{cases} r, & \text{if } (\alpha_r^*)^{-1} \leq (\alpha_{r-1}^*)^{-1}; \\ r-1, & \text{otherwise} \end{cases}$$

and  $r = \operatorname{argmax}\{i \leq k | \alpha_i = \alpha_i^*\}$ . Let  $k_0 = m - 1$  and  $M_2 = 2$ . Then  $\max\{k - k_0, 0\} \leq r_0 \leq k$ . Similar to (20) we can deduce that  $\alpha_k^{-1} \geq (\alpha_{r_0}^*)^{-1} \geq \frac{2}{3}\lambda_{l-1}$ . Thus we conclude that property (ii) also holds for this algorithm.  $\square$

**Theorem 3** [4, Theorem 4.1] *Consider the same assumptions in Lemma 1. As long as the stepsizes satisfy property (i) and (ii), the gradient method either terminates in finite iterations or converges R-linearly.*

From the above theorem, we conclude that all the four proposed gradient algorithms converge R-linearly when they are applied to  $n$ -dimensional convex quadratic functions.

## 4 Numerical Results

In this section, we show the performances of the proposed algorithms. The algorithms are compared with several gradient methods mentioned in Sect. 1. For all the generated test problems, the Hessian matrix is set as a diagonal matrix, where  $H = \operatorname{Diag}(\lambda_1, \dots, \lambda_n)$ . For each setting, we generate 10 examples, and show the average iteration numbers as well as the computational time. All the tests are done on Matlab R2014a, Intel core i7-6700HQ.

For Alg. 1, two different versions are implemented. In version A, we use  $\alpha_k^{YV}$  computed by (4); while in version B, we replace  $\alpha_k^{YV}$  by

$$\alpha_k^{New} = \tilde{\alpha}_k [1 + \rho_k (\alpha_k^{YV})^2],$$

where  $\rho_k = \frac{1}{\alpha_{k-1}^* \alpha_k^*} - \frac{\|g_k\|_2^2}{(\alpha_{k-1}^* \|g_{k-1}\|_2)^2}$ . Theoretically  $\alpha_k^{New} = \alpha_k^{YV}$ , but computationally they give slightly different numerical performances. Here we report the better results between the two versions.

We consider four kinds of test problems. Given dimension  $n$  and condition number  $\kappa$ , their detailed settings are as follows:

**Test 1.** [10] In the Hessian matrix, we set  $\lambda_1 = \kappa$ ,  $\lambda_n = 1$ ;  $\lambda_j$  are randomly generated in  $[1, \kappa]$ , for  $j = 2, \dots, n - 1$ ;  $b$  is randomly generated in  $[-5, 5]^n$ . The stopping parameter  $\epsilon = 10^{-8} \|g_0\|$ .

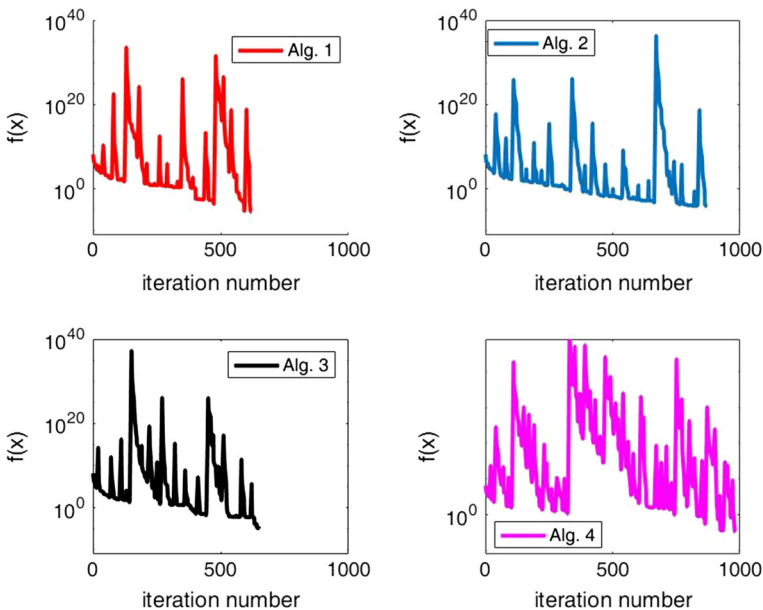
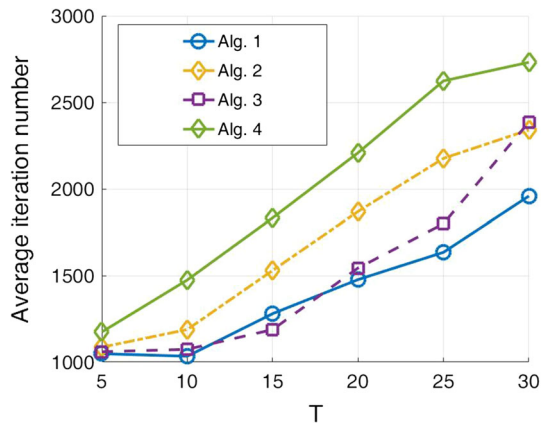
**Test 2.** [10, 12] Let  $\lambda_j = \kappa^{\frac{n-j}{n-1}}$ , for  $j = 1, \dots, n$ , and  $b$  be a zero vector. The initial point  $x_0$  is randomly generated in  $[-5, 5]^n$ .

**Test 3.** [12] Let  $\lambda_j = 1 + (\kappa - 1)s_j$  and  $b$  be a zero vector.  $s_j$  is randomly generated from  $[0.8, 1]$  for  $j = 1, \dots, \frac{n}{2}$ , and randomly generated from  $[0, 0.2]$  for  $j = \frac{n}{2} + 1, \dots, n$ . The stopping parameter  $\epsilon = 10^{-6}$ . The initial point  $x_0$  is randomly generated on a unit sphere.

**Test 4.** [16] Let  $\lambda_j = \frac{\kappa}{2} (\cos \frac{n-j}{n-1} \pi + 1)$  and  $b$  be a zero vector. The stopping parameter  $\epsilon = 10^{-6}$ . The initial point  $x_0$  is randomly generated on a unit sphere.

First, the property of the proposed algorithms are explored. Fig. 1 depicts the average iteration numbers of the considered four algorithms with respect to different values of

**Fig. 1** Iteration number with respect to different  $m$  for “Test 2”



**Fig. 2** Iteration curves for “Test 1”

the parameter  $m$ , where “Test 2” is solved. Here  $n = 10^3$ ,  $\kappa = 10^4$  and the stopping parameter  $\epsilon = 10^{-6}$ . It shows that Alg. 1 and Alg. 3 with  $m = 10$  outperform the other settings. We also plot the iteration curves representing function values of the four algorithms in Fig. 2. “Test 1” is considered, with  $n = 10^3$ ,  $\kappa = 10^4$ ,  $m = 10$  and  $x_0$  randomly generated in  $[-5, 5]^n$ . It indicates that all the four algorithms are nonmonotonic. The good performances of Alg. 1 and Alg. 3 are observed again.

Next, we test some problems with various dimensions and condition numbers. Here we compare the proposed algorithms with the effective gradient methods analyzed in [12], such as SDC,  $ABB_{\min}$  and LMSD. Let  $m = 10$ . The parameters in SDC,  $ABB_{\min}$  and LMSD are the same as in [12]. In “Test 1”,  $n = 10^4$ ,  $\kappa = 10^4$  and  $x_0$  is randomly

**Table 1** Comparison with the existing methods for different test problems

	Alg. 1(10)	Alg. 2(10)	Alg. 3(10)	Alg. 4(10)	SDC	ABB <sub>min</sub>	LMSD
Iter. No. (Test 1)	658.4	725	589.9	1115.5	564.8	<b>526.4</b>	672.6
time (Test 1)	68.2729	105.3888	<b>61.5247</b>	161.2105	79.7266	181.3855	117.9999
Iter. No. (Test 2)	<b>5326.9</b>	6154.2	5449.4	8750.3	7153.9	5428.5	19301
time (Test 2)	<b>582</b>	1068.3	582.4	1495.1	1087.7	2133.1	4067.1
Iter. No. (Test 3)	151.1	325.4	174.9	356	<b>119.9</b>	128.7	123.6
time (Test 3)	<b>0.1519</b>	0.6515	0.2142	0.7186	0.1795	0.4667	0.2565
Iter. No. (Test 4)	470.8	5443.1	483	8617.3	683	<b>371.8</b>	1551.3
time (Test 4)	<b>0.3833</b>	5.9411	0.3941	9.2585	0.7759	1.0145	2.1966

generated in  $[-5, 5]^n$ ; in “Test 2”,  $n = 10^4$ ,  $\kappa = 10^6$  and  $\epsilon = 10^{-8} \|g_0\|$ ; in “Test 3”,  $n = 10^3$  and  $\kappa = 10^3$ ; in “Test 4”,  $n = 10^3$  and  $\kappa = 10^5$ . Table 1 shows the iteration numbers as well as the computational time of the compared methods for the four kinds of test problems. The best results are marked in boldface type. It turns out that Alg. 1 and Alg. 3 outperform the compared algorithms generally, in terms of computational time. Due to the fixed step lengths in each loop, the proposed algorithms have lower per iteration computational cost<sup>1</sup>. Consequently they require lower total computational cost compared to the state of the art. But Alg. 2 and Alg. 4 are overperformed by the compared algorithms. The reason may be that the stepsize (8) in Alg. 2 is sometimes too short, and that (10) in Alg. 4 is sometimes too long. From the four tests, we can also observe that the computational complexity of all the considered gradient methods increases with the increment of the dimension and the condition number.

With various types of problems and different settings, the above numerical tests indicate the superior performances of the proposed algorithms Alg. 1 and Alg. 3.

## 5 Conclusion

In this paper, we proposed four gradient methods for convex quadratic programming. First, a new framework to update the stepsizes was proposed. In each outer loop, only 2 exact linesearch steps are calculated and the other  $m - 2$  iterations use the fixed steplengths. Four different stepsizes were applied in the new framework, and thus four new gradient algorithms were proposed. Theoretically, for 2-dimensional convex quadratic function minimization problems, we proved that the proposed algorithms have either finite terminations or R-superlinear convergence rate. For general  $n$ -dimensional problems, we proved that all new algorithms converge R-linearly. Numerical results indicate the effectiveness and the robustness of the proposed Alg. 1 and Alg. 3.

<sup>1</sup> For example, in this test problem, the per iteration computational cost of ABB<sub>min</sub> is  $n^2 + 3n + \frac{5}{2}$ , while that of Alg. 3 is  $\frac{1}{5}n^2 + \frac{2}{5}n + \frac{3}{10}$ .

**Acknowledgements** The authors would like to thank Prof. Ya-xiang Yuan from Chinese Academy of Sciences, the editor and the two anonymous reviewers for their valuable suggestions and comments.

## 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 systèmes d'équations simultanées. *Comp. Rend. Sci. Paris* **25**, 46–89 (1847)
3. Curry, H.B.: The method of steepest descent for nonlinear minimization problems. *Q. Appl. Math.* **2**, 258–261 (1944)
4. Dai, Y.-H.: Alternate step gradient method. *Optimization* **52**(4–5), 395–415 (2003)
5. Dai, Y.-H.: A new analysis on the Barzilai–Borwein gradient method. *J. Oper. Res. Soc. China* **1**(2), 187–198 (2013)
6. Dai, Y.-H., Fletcher, R.: On the asymptotic behaviour of some new gradient methods. *Math. Program.* **103**(3), 541–559 (2005)
7. Dai, Y.-H., Yuan, Y.-X.: Analysis of monotone gradient methods. *J. Ind. Manag. Optim.* **1**, 181–192 (2005)
8. Dai, Y.-H., Liao, L.-Z.:  $R$ -linear convergence of the Barzilai and Borwein gradient method. *IMA J. Numer. Anal.* **22**, 1–10 (2002)
9. De Asmundis, R., di Serafino, D., Riccio, F., Toraldo, G.: On spectral properties of steepest descent methods. *IMA J. Numer. Anal.* **33**(4), 1416–1435 (2013)
10. De Asmundis, R., di Serafino, D., Hager, W.W., Toraldo, G., Zhang, H.-C.: An efficient gradient method using the Yuan steplength. *Comp. Opt. Appl.* **59**(3), 541–563 (2014)
11. de Klerk, E., Glineur, F., Taylor, A.B.: On the worst-case complexity of the gradient method with exact linesearch for smooth strongly convex functions. *Optim. Lett.* **11**, 1185–1199 (2017)
12. di Serafino, D., Ruggiero, V., Toraldo, G., Zanni, L.: On the steplength selection in gradient methods for unconstrained optimization. *Appl. Math. Comput.* **318**, 176–195 (2018)
13. Fletcher, R.: A limited memory steepest descent method. *Math. Program. Ser. A* **135**, 413–436 (2012)
14. Frassoldati, G., Zanni, L., Zanghirati, G.: New adaptive stepsize selections in gradient methods. *J. Ind. Manag. Optim.* **4**(2), 299–312 (2008)
15. Friedlander, A., Martinez, J.M., Molina, B., Raydan, M.: Gradient method with retards and generalizations. *SIAM J. Numer. Anal.* **36**, 275–289 (1999)
16. Gonzaga, C., Schneider, R.M.: On the steepest descent algorithm for quadratic functions. *Comput. Optim. Appl.* **63**(2), 523–542 (2016)
17. Nocedal, J., Sartenauer, A., Zhu, C.: On the behavior of the gradient norm in the steepest descent method. *Comput. Optim. Appl.* **22**(1), 5–35 (2002)
18. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.* **7**, 26–33 (1997)
19. Raydan, M., Svaiter, B.F.: Relaxed steepest descent and Cauchy–Barzilai–Borwein method. *Comput. Optim. Appl.* **21**, 155–167 (2002)
20. Vrahatis, M.N., Androulakis, G.S., Lambrinos, J.N., Magoulas, G.D.: A class of gradient unconstrained minimization algorithms with adaptive step-size. *J. Comput. Appl. Math.* **114**(2), 367–386 (2000)
21. Yuan, Y.-X.: A new stepsize for the steepest descent method. *J. Comput. Math.* **24**(2), 149–156 (2006)
22. Yuan, Y.-X.: Step-sizes for the gradient method. *AMS IP Stud. Adv. Math.* **42**(2), 785–796 (2008)
23. Yuan, Y.-X.: A short note on the  $Q$ -linear convergence of the steepest descent method. *Math. Program.* **123**, 339–343 (2010)
24. Yuan, Y.-X.: Gradient methods for large scale convex quadratic functions. In: Wang, Y., Yang, C., Yagola, A.G. (eds.) *Optimization and regularization for computational inverse problems and applications*, pp. 141–155. Springer, Berlin (2010)
25. Zheng, Y., Zheng, B.: A new modified Barzilai–Borwein gradient method for the quadratic minimization problem. *J. Optim. Theory Appl.* **172**(1), 179–186 (2017)