Noname manuscript No.
(will be inserted by the editor)

Quadratic regularization projected alternating Barzilai–Borwein method for constrained optimization

Yakui Huang · Hongwei Liu · Sha Zhou

Received: date / Accepted: date

Abstract In this paper, based on the regularization techniques and projected gradient strategies, we present a quadratic regularization projected alternating Barzilai–Borwein (QRPABB) method for minimizing differentiable functions on closed convex sets. We show the convergence of the QRPABB method to a constrained stationary point for a nonmonotone line search. When the objective function is convex, we prove the error in the objective function at iteration k is bounded by $\frac{a}{k+1}$ for some a independent of k. Moreover, if the objective function is strongly convex, then the convergence rate is R-linear. Numerical comparisons of methods on box-constrained quadratic problems and nonnegative matrix factorization problems show that the QRPABB method is promising.

Keywords Constrained optimization · Projected Barzilai–Borwein method · Quadratic regularization · Linear convergence · Nonnegative matrix factorization

1 Introduction

We consider the following constrained minimization problem

$$\begin{aligned} &\min \ \ f(x) \\ &s.t. \ \ x \in \Omega, \end{aligned} \tag{1}$$

where $\Omega \subseteq \mathbb{R}^n$ is a nonempty closed convex set and $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable on an open set that contains Ω.

Yakui Huang · Hongwei Liu · Sha Zhou
School of Mathematics and Statistics, Xidian University, Xi'an 710071, PR China
E-mail: huangyakui2006@gmail.com

Throughout this paper, we assume that f is bounded below and $\nabla f(x)$ is Lipschitz continuous on Ω with Lipschitz constant L_f, that is,

$$\|\nabla f(x) - \nabla f(y)\| \leq L_f \|x - y\|, \ \forall \ x, y \in \Omega.$$

Moreover, for a given $x \in \mathbb{R}^n$ it is easy to compute its orthogonal projection onto the set Ω denoted by

$$P(x) = \arg\min_{z \in \Omega} \ \|x - z\|.$$

Projected gradient (PG) methods are popular for solving problem (1), see [5,11,25,26,32,40,43] for example. However, early PG methods suffer from slow convergence like steepest descent [6]. In [2], Barzilai and Borwein (BB) introduced two ingenious stepsizes that significantly improve the performance of gradient methods. For two-dimensional strongly convex quadratic functions, they established the surprising R-superlinear convergence of the BB method which is superior to the classic steepest descent method. Recent literature have shown very promising performance obtained by the BB method and its variants [19,21,24,33,34,36,50]. By combining nonmonotone schemes with classical PG strategies, BB-like methods have been extended to constrained optimization. Birgin and Martínez [8] developed the so-called spectral gradient (SPG) method for solving convex constrained problems. Dai and Fletcher [18] considered to use the two BB stepsizes alternately with an adaptive nonmonotone line search and proposed a projected BB (PABB) method to solve box-constrained quadratic programming. Projected BB methods have been successfully applied in many areas, for example, Han et al [35] proposed four projected BB algorithms for solving the nonnegative matrix factorization (NMF) problems [39,48]. Although the convergence of the BB method for unconstrained optimization problems has been studied extensively [17,20,49], there are few results on the rate of convergence of projected BB methods. Recently, Hager et al [33] showed the R-linear convergence of a BB-like method for minimizing the sum of two functions. Since the objective function is possibly nonsmooth, by using of the indicator function for Ω, the convergence results in [33] apply to problem (1). For more details on BB-like methods see [10,24] and references therein.

Optimal methods provide us an alternative way of solving problem (1). An algorithm is called optimal if it can achieve the worst-case complexity. It has been shown by Nemirovski and Yudin [44] that the lower complexity bound of first-order methods for smooth convex functions with Lipschitz continuous gradient is $O(1/k^2)$ where k is the iteration counter. In 1983, Nesterov [45] presented his seminal optimal first-order method for smooth convex problems. Several optimal schemes for both unconstrained and constrained problems can be found in Nesterov's book [46]. However, the aforementioned methods depend on the availability of the Lipschitz constant for ∇f. A scheme for estimating the Lipschitz constant was proposed by Nesterov [47] for minimizing composite functions that possibly nonsmooth. Beck and Teboulle [3] applied a similar idea to linear inverse problems arising in signal/image processing. In

order to make use of the information of a strong convexity constant, Gonzaga et al [28] extended Nesterov's methods [46] and proposed an optimal algorithm for solving problems in the form (1). Due to their surprising computational efficiency and interesting theoretical properties, optimal first-order methods have attracted much attention in recent literature, see [1,4,27,38,46,51] for example.

For our problem, most existing BB-like methods including SPG and PABB do not make use of the information of the Lipschitz constant L_f. From the theoretical point of view, many convergence results of optimal methods depend on the Lipschitz constant L_f [28,45–47]. This may degrade their practical numerical efficiency for a large L_f.

In this paper, based on the regularization techniques and PG strategies, we propose a quadratic regularization projected alternating BB (QRPABB) method for solving problem (1). At each iteration, the QRPABB method first computes a point by minimizing a quadratic regularization function of $f(x)$ over Ω, where the regularization weight is an estimation of the Lipschitz constant L_f that determined by a trust region strategy. Then it updates the iterate by running a PG step where the two BB stepsizes are used alternately. We employ a nonmonotone line search proposed by Zhang and Hager [53] to accelerate the convergence. We prove the convergence of the QRPABB method to a stationary point of (1) for general nonconvex functions. When the objective is convex, we show that there exists a constant $a > 0$ such that $f(x_k) - f(x_*) \leq \frac{a}{k+1}$ where x_* is a solution and k is the iteration counter. Moreover, if the objective is strongly convex, then the rate of convergence is R-linear.

This paper is organized as follows. In Section 2, we present the QRPABB method formally. In Section 3, we prove that the QRPABB method converges to a stationary point of (1) for general nonconvex functions. Moreover, we establish sublinear and R-linear convergence of the proposed method for convex and strongly convex objective functions, respectively. We present some numerical results in Section 4 to demonstrate the feasibility and efficiency of our method. Finally, in Section 5, we draw some conclusions.

2 Quadratic regularization projected alternating Barzilai–Borwein method

In this section, we present a quadratic regularization projected alternating Barzilai–Borwein (QRPABB) method for solving problem (1).

Our approach is based on the regularization techniques and PG strategies. At the k-th iteration, the QRPABB method computes a point z_k by minimizing the quadratic approximation of f around x_k, i.e.,

$$z_k = \arg\min_{x\in\Omega} \left\{ \phi(x) = f(x_k) + \nabla f(x_k)^T(x - x_k) + \frac{L_k}{2}\|x - x_k\|^2 \right\}, \qquad (2)$$

where the regularization weight $L_k > 0$ is an estimation of the Lipschitz constant L_f that determined by a trust region strategy [7,12–14], and then calculates x_{k+1} by a PG step incorporating the nonmonotone scheme.

Now we present our QRPABB method in Algorithm 1.

Algorithm 1

Step 1. Choose constants $\sigma, \sigma_1, \sigma_2, \rho \in (0,1)$, $\eta > 1$, $\alpha_{\max} > \alpha_{\min} > 0$. Initialize iteration counter $k = 0$, $L_0 > 0$, and initial guess $x_0 \in \Omega$.

Step 2. Compute z_k by (2). If $z_k = x_k$, stop; otherwise, define

$$r_k = \frac{f(z_k) - f(x_k) - \nabla f(x_k)^T(z_k - x_k)}{\frac{1}{2}L_k\|z_k - x_k\|^2}. \tag{3}$$

If $r_k > 1$, let $z_k = x_k$.

Step 3. Updating the regularization weight: Set

$$L_{k+1} = \begin{cases} \sigma_1 L_k, & \text{if } r_k \le \sigma_2; \\ L_k, & \text{if } r_k \in (\sigma_2, 1]; \\ \eta L_k, & \text{if } r_k > 1. \end{cases} \tag{4}$$

Step 4. Nonmonotone line search. Find the smallest nonnegative integer m_k such that

$$f(z_k + \rho^{m_k} d_k) \le f_k^r + \sigma\rho^{m_k}\nabla f(z_k)^T d_k, \tag{5}$$

where f_k^r is a reference value and

$$d_k = P(z_k - \alpha_k \nabla f(z_k)) - z_k, \tag{6}$$

with $\alpha_k \in [\alpha_{\min}, \alpha_{\max}]$. Set $\lambda_k = \rho^{m_k}$ and $x_{k+1} = z_k + \lambda_k d_k$

Step 5. Set $k = k + 1$ and go to Step 2.

Since the function $\phi(x)$ is strongly convex, (2) has a simple closed-form solution

$$z_k = P\left(x_k - \frac{1}{L_k}\nabla f(x_k)\right). \tag{7}$$

Noting that f is continuously differentiable with Lipschitz continuous gradient on Ω, we have $r_k \le 1$ for all $L_k \ge L_f$. Moreover, by the rule (4), we set $L_{k+1} = \eta L_k$ only when $L_k < L_f$. Therefore, we have

$$L_k \le \max\{L_0, \eta L_f\}, \ \forall\ k \ge 0. \tag{8}$$

See Lemma 3.3 of [7].

By the optimality conditions of constrained optimization, we know that $x \in \Omega$ is a stationary point of (1) if and only if

$$\|P(x - \kappa\nabla f(x)) - x\| = 0, \tag{9}$$

for any fixed $\kappa > 0$. Apparently, $x_k, z_k \in \Omega$ for all $k \ge 0$. Thus, if x_k is stationary, by (7), then $z_k = x_k$ is also stationary. On the other hand, if z_k is a stationary point of (1), then $d_k = 0$ which implies $x_{k+1} = z_k$ is stationary.

As we know, the BB stepsizes can improve the performance of PG methods significantly. We use the following two BB stepsizes for our method:

$$\alpha_k^{BB1} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}}, \tag{10}$$

and

$$\alpha_k^{BB2} = \frac{s_{k-1}^T y_{k-1}}{y_{k-1}^T y_{k-1}}, \tag{11}$$

where $s_{k-1} = x_k - z_{k-1}$ and $y_{k-1} = \nabla f(x_k) - \nabla f(z_{k-1})$. Dai and Fletcher [18] have shown by numerical experiments that use the two BB stepsizes alternately is better than use one throughout. Thus, we adopt their strategy to use

$$\alpha_k^{ABB} = \begin{cases} \alpha_k^{BB1}, \text{ for odd } k; \\ \alpha_k^{BB2}, \text{ for even } k. \end{cases} \tag{12}$$

We restrict α_k^{ABB} in an interval to avoid uphill direction

$$\alpha_k = \min\{\alpha_{\max}, \max\{\alpha_{\min}, \alpha_k^{ABB}\}\}. \tag{13}$$

Clearly, if $f_k^r = f(z_k)$ for all k, then (5) reduces to the well-known Armijo line search which enforces the function value to decrease at every iteration. However, BB-like methods are often more efficient with nonmonotone schemes. One popular such scheme is proposed by Grippo, Lampariello, and Lucidi (GLL) [29], where f_k^r is the largest objective of the last M iterations with M being a fixed integer. Although the GLL nonmonotone line search technique has been incorporated into many optimization algorithms, it has been observed several drawbacks. Dai [16] showed by an example that the iterates may not satisfy the inequality (5) for sufficiently large k, for any fixed M. Moreover, the numerical performance of the GLL nonmonotone line search is heavily depend on M in some cases, see [29,50] for example. We consider to use the nonmonotone scheme introduced by Zhang and Hager [53]. Let $f_0^r = f(z_0)$, $Q_0 = 1$. Define $Q_{k+1} = \gamma_k Q_k + 1$ and

$$f_{k+1}^r = \frac{\gamma_k Q_k f_k^r + f(z_{k+1})}{Q_{k+1}}, \tag{14}$$

where $\gamma_k \in [\gamma_{\min}, \gamma_{\max}]$ with $0 \le \gamma_{\min} \le \gamma_{\max} \le 1$. By Lemma 1.1 of [53], we know that $f(z_k) \le f_k^r$.

3 Convergence

3.1 Global convergence

The following property of projection is needed in our analysis.

Lemma 1 *[11] Let $z \in \Omega$, then for all $x \in \mathbb{R}^n$ we have*

$$\langle P(x) - x, z - P(x) \rangle \ge 0.$$

We can show a lower bound for the step length λ_k in a similar way as Lemma 4 in [36], see also Lemma 2.1 in [53].

Lemma 2 *The step length λ_k of Algorithm 1 satisfies*

$$\lambda_k \geq \min\left\{1, \frac{2\rho(1-\sigma)}{\alpha_{\max} L_f}\right\} := \bar{\lambda}. \tag{15}$$

Proof If $\lambda_k = 1$, we need no proof. Consider the case that the inequality (5) fails at least once, then

$$\begin{aligned} f(z_k + \frac{\lambda_k}{\rho} d_k) &> f_k^r + \sigma \frac{\lambda_k}{\rho} \nabla f(x_k)^T d_k, \\ &\geq f(z_k) + \sigma \frac{\lambda_k}{\rho} \nabla f(x_k)^T d_k. \end{aligned} \tag{16}$$

Using Lipschitz continuity of f, we have

$$f(z_k + \frac{\lambda_k}{\rho} d_k) \leq f(z_k) + \frac{\lambda_k}{\rho} \nabla f(x_k)^T d_k + \frac{L_f}{2} \frac{\lambda_k^2}{\rho^2} \|d_k\|^2. \tag{17}$$

It follows from (16) and (17) that

$$(\sigma - 1) \nabla f(x_k)^T d_k \leq \frac{\lambda_k L_f}{2\rho} \|d_k\|^2. \tag{18}$$

By Lemma 1, we have

$$\nabla f(z_k)^T d_k \leq -\frac{1}{\alpha_k} \|d_k\|^2 \leq -\frac{1}{\alpha_{\max}} \|d_k\|^2, \tag{19}$$

which together with (18) implies that

$$\lambda_k \geq \frac{2\rho(1-\sigma)}{\alpha_{\max} L_f}.$$

This completes the proof. □

Next lemma shows that $f(z_k) \leq f(x_k)$ holds throughout the iterative process.

Lemma 3 *If $r_k \leq 1$, then*

$$f(z_k) \leq f(x_k) - \frac{L_k}{2} \|z_k - x_k\|^2. \tag{20}$$

Proof Notice that $x_k \in \Omega$, by Lemma 1 and (7), we have

$$\nabla f(x_k)^T (z_k - x_k) \leq -L_k \|z_k - x_k\|^2.$$

Since $r_k \leq 1$, by the definition of r_k, we obtain

$$\begin{aligned} f(z_k) \leq \phi(z_k) &= f(x_k) + \nabla f(x_k)^T (z_k - x_k) + \frac{L_k}{2} \|z_k - x_k\|^2 \\ &\leq f(x_k) - \frac{L_k}{2} \|z_k - x_k\|^2. \end{aligned}$$

□

Now we prove the convergence of Algorithm 1 to a stationary point of (1).

Theorem 1 *Any accumulation point of* $\{z_k\}$ *generated by Algorithm 1 with* $\gamma_{\max} < 1$ *is a stationary point of* (1).

Proof By Lemma 2, the definition of f_k^r, and (19), we have

$$\begin{aligned} f_{k+1}^r &= \frac{\gamma_k Q_k f_k^r + f(z_{k+1})}{Q_{k+1}} \\ &\leq \frac{\gamma_k Q_k f_k^r + f(x_{k+1})}{Q_{k+1}} \\ &\leq \frac{\gamma_k Q_k f_k^r + f_k^r + \sigma\lambda_k \nabla f(z_k)^T d_k}{Q_{k+1}} \\ &\leq f_k^r - \frac{\sigma\bar{\lambda}\|d_k\|^2}{\alpha_{\max} Q_{k+1}}. \end{aligned} \tag{21}$$

Since $\gamma_{\max} < 1$, we have

$$Q_{k+1} = 1 + \sum_{i=1}^{k}\prod_{j=1}^{i}\gamma_{k-j} \leq 1 + \sum_{i=1}^{k}\gamma_{\max}^{i+1} \leq \sum_{i=0}^{\infty}\gamma_{\max}^{i} \leq \frac{1}{1-\gamma_{\max}}. \tag{22}$$

Combining (21) and (22), we obtain

$$f_{k+1}^r \leq f_k^r - \frac{\sigma\bar{\lambda}(1-\gamma_{\max})\|d_k\|^2}{\alpha_{\max}}. \tag{23}$$

Therefore, the sequence $\{f_k^r\}$ is monotonically decreasing. Recall that $f(z_k) \leq f_k^r$ and f is bounded below, then f_k^r is bounded below as well. Taking limits in both sides of (23) to get

$$\lim_{k\to\infty}\|d_k\| = 0. \tag{24}$$

Let $\bar{z} \in \Omega$ be an accumulation point of $\{z_k\}$. Suppose that a subsequence $\{z_{k_j}\}$ converges to $\bar{z}$. Since $\alpha_{k_j} \in [\alpha_{\min}, \alpha_{\max}]$ for all j, by taking a subsequence if necessary, we have $\alpha_{k_j} \to \bar{\alpha} > 0$. By the continuity of ∇f and $\|\cdot\|$, (6), and (24), we have

$$\|P(\bar{z} - \bar{\alpha}\nabla f(\bar{z})) - \bar{z}\| = 0.$$

Noting that $\{z_k\} \subseteq \Omega$ and Ω is closed, thus, by (9), $\bar{z} \in \Omega$ is stationary. □

From (24), we conclude that any accumulation point of $\{x_k\}$ is also a constrained stationary point.

3.2 Rate of convergence

In this subsection, we will establish sublinear and R-linear convergence of Algorithm 1 using the techniques in [33].

For a given $x_0 \in \Omega$, define the level set by

$$\mathcal{L}(x_0) = \{x | f(x) \leq f(x_0),\ x \in \Omega\}.$$

By Lemma 3, the condition (5), and (21), one has

$$f(z_{k+1}) \leq f(x_{k+1}) \leq f_k^r \leq f_{k-1}^r \leq f(z_0) \leq f(x_0).$$

Therefore, the two sequences $\{x_k\}$ and $\{z_k\}$ are contained in $\mathcal{L}(x_0)$.

In what follows, we assume that the level set $\mathcal{L}(x_0)$ is bounded, f attains a minimum on Ω at point x_* and the associated objective function value $f_* = f(x_*)$.

In order to prove the rate of convergence of Algorithm 1, we define an auxiliary sequence as follows. Let $C_0 = f(x_0)$ and

$$C_{k+1} = \frac{\gamma_k Q_k C_k + f(x_{k+1})}{Q_{k+1}}. \tag{25}$$

Then, by Lemma 3 and the definition of f_k^r, we have $f(z_k) \leq f_k^r \leq C_k$.

Lemma 4 *If f is convex on Ω and $\gamma_{\max} < 1$, then $\lim_{k\to\infty} C_k = f_*$.*

Proof It follows from the definitions of Q_k and C_k that

$$\begin{aligned} C_k - f_* &= \frac{\gamma_{k-1} Q_{k-1}(C_{k-1} - f_*) + f(x_k) - f_*}{Q_k} \\ &= \frac{\sum_{i=0}^{k}[(\prod_{j=i}^{k-1} \gamma_j)(f(x_i) - f_*)]}{Q_k} \\ &\leq \sum_{i=0}^{k}[(\prod_{j=i}^{k-1} \gamma_j)(f(x_i) - f_*)]. \end{aligned}$$

By Lemma 1.1 of [53] again, we know that $f(x_k) \leq C_k$. Then we obtain

$$f(x_k) - f_* \leq C_k - f_* \leq \sum_{i=0}^{k}[(\prod_{j=i}^{k-1} \gamma_j)(f(x_i) - f_*)].$$

Since f is convex, a stationary point is a global minimizer. From Theorem 1 we know that $f(x_k) - f_* \to 0$. Notice that $\gamma_{\max} < 1$, we can conclude C_k converges to f_*. □

Next theorem gives sublinear convergence of Algorithm 1.

Theorem 2 *Let $\{x_k\}$ be a sequence generated by Algorithm 1 with $\gamma_{\max} < 1$. If f is convex on Ω, then there exists a constant a such that*

$$f(x_k) - f_* \leq \frac{a}{k+1}, \ \forall \ k \geq 0.$$

Proof From Lipschitz continuity of f and the fact that $\bar{\lambda} \leq \lambda_k \leq 1$, we have

$$\begin{aligned} f(x_{k+1}) &= f(z_k + \lambda_k d_k) \\ &\leq f(z_k) + \lambda_k \nabla f(z_k)^T d_k + \frac{\lambda_k^2 L_f}{2}\|d_k\|^2 \\ &\leq (1-\lambda_k) f(z_k) + \lambda_k \psi(z_k + d_k) + \frac{\lambda_k L_f}{2}\|d_k\|^2, \end{aligned} \tag{26}$$

where

$$\psi(x) = f(z_k) + \nabla f(z_k)^T (x - z_k) + \frac{1}{2\alpha_k}\|x - z_k\|^2.$$

By the definitions of d_k, we know that $z_k + d_k$ is the unique minimizer of $\psi(x)$ on Ω. It follows from the convexity of f that

$$\begin{aligned} \psi(z_k + d_k) &= \min_{x\in\Omega} \left\{ f(z_k) + \nabla f(z_k)^T (x - z_k) + \frac{1}{2\alpha_k}\|x - z_k\|^2 \right\} \\ &\leq \min_{x\in\Omega} \left\{ f(x) + \frac{1}{2\alpha_k}\|x - z_k\|^2 \right\}. \end{aligned}$$

Let $x = (1-\delta)z_k + \delta x_*$, $\delta \in [0,1]$, we have

$$\begin{aligned} \min_{x\in\Omega} \ \left\{ f(x) + \frac{1}{2\alpha_k}\|x - z_k\|^2 \right\} &\leq f((1-\delta)z_k + \delta x_*) + \frac{1}{2\alpha_{\min}}\delta^2 \|z_k - x_*\|^2 \\ &\leq (1-\delta) f(z_k) + \delta f_* + \delta^2 \beta_k, \end{aligned}$$

where $\beta_k := \dfrac{1}{2\alpha_{\min}}\|z_k - x_*\|^2$. This together with (26) gives

$$\begin{aligned} f(x_{k+1}) &\leq (1 - \lambda_k\delta) f(z_k) + \lambda_k(\delta f_* + \delta^2 \beta_k) + \frac{\lambda_k L_f}{2}\|d_k\|^2 \\ &\leq (1 - \lambda_k\delta) C_k + \lambda_k(\delta f_* + \delta^2 \beta_k) + \frac{\lambda_k L_f}{2}\|d_k\|^2 \\ &= C_k + \lambda_k(\delta f_* - \delta C_k + \delta^2 \beta_k) + \frac{\lambda_k L_f}{2}\|d_k\|^2. \end{aligned} \tag{27}$$

Since $z_k, x_* \in \mathcal{L}(x_0)$ and the level set $\mathcal{L}(x_0)$ is bounded, we have

$$\beta_k = \frac{1}{2\alpha_{\min}}\|z_k - x_*\|^2 \leq \frac{1}{2\alpha_{\min}}(\text{diameter of } \mathcal{L})^2 := b_1. \tag{28}$$

It follows from (5) and (19) that

$$\lambda_k \|d_k\|^2 \leq \frac{\alpha_{\max}}{\sigma}(f_k^r - f(x_{k+1})) \leq \frac{\alpha_{\max}}{\sigma}(C_k - f(x_{k+1})). \tag{29}$$

Combining (27), (28), and (29), we have

$$f(x_{k+1}) \leq C_k + \lambda_k(\delta f_* - \delta C_k + \delta^2 b_1) + b_2(C_k - f(x_{k+1})), \tag{30}$$

where $b_2 = \frac{\alpha_{\max} L_f}{2\sigma}$. Let $h(\delta) = \delta f_* - \delta C_k + \delta^2 b_1$, which attains its minimum at

$$\delta_{\min} = \min\left\{1, \frac{C_k - f_*}{2b_1}\right\}.$$

From Lemma 4 we know that $\delta_{\min} < 1$ holds for sufficiently large k. By (30), when $\delta_{\min} < 1$, we have

$$\begin{aligned} f(x_{k+1}) &\leq C_k - \frac{\lambda_k (C_k - f_*)^2}{4b_1} + b_2(C_k - f(x_{k+1})) \\ &\leq C_k - b_3(C_k - f_*)^2 + b_2(C_k - f(x_{k+1})), \end{aligned} \tag{31}$$

where $b_3 = \dfrac{\bar{\lambda}}{4b_1}$ with $\bar{\lambda}$ given by (15). By subtracting f_* from each side of (31) and rearranging terms, we have

$$f(x_{k+1}) - f_* \leq C_k - f_* - b_4(C_k - f_*)^2, \tag{32}$$

where $b_4 = \frac{b_3}{1+b_2}$. Combining the definition of C_{k+1}, (22), and (32) gives

$$\begin{aligned} C_{k+1} - f_* &= \frac{\gamma_k Q_k (C_k - f_*) + f(x_{k+1}) - f_*}{Q_{k+1}} \\ &\leq \frac{\gamma_k Q_k (C_k - f_*) + C_k - f_* - b_4(C_k - f_*)^2}{Q_{k+1}} \\ &= C_k - f_* - \frac{b_4}{Q_{k+1}}(C_k - f_*)^2 \\ &\leq C_k - f_* - b_4(1 - \gamma_{\max})(C_k - f_*)^2, \ k > k_0. \end{aligned} \tag{33}$$

Let $r_k = C_k - f_*$, exploit the monotonicity of C_k, we have for $k > k_0$,

$$\frac{1}{r_{k+1}} \geq \frac{1}{r_k} + b_4(1 - \gamma_{\max}).$$

Applying the above inequality recursively gives

$$\frac{1}{r_k} \geq \frac{1}{r_{k_0}} + b_4(1 - \gamma_{\max})(k - k_0),$$

which implies that

$$r_k \leq \frac{r_{k_0}}{1 + b_4 r_{k_0}(1 - \gamma_{\max})(k - k_0)} \leq \frac{1}{b_4(1 - \gamma_{\max})(k - k_0)}, \ k > k_0.$$

For these k such that $k > 2k_0$, we have

$$r_k \leq \frac{2}{b_4(1 - \gamma_{\max})k} \leq \frac{2}{b_4(1 - \gamma_{\max})k} = \frac{b}{k} \leq \frac{2b}{k+1},$$

where $b = \dfrac{2}{b_4(1-\gamma_{\max})}$. Choose a finite $a > 2b$ for all $k \in [0, 2k_0]$, we obtain

$$f(x_k) - f_* \leq C_k - f_* = r_k \leq \frac{a}{k+1}.$$

We finish the proof. □

Now we are ready to prove the R-linear convergence of Algorithm 1.

Theorem 3 *Let $\{z_k\}$ be a sequence generated by Algorithm 1 with $\gamma_{\max} < 1$. If f is convex on Ω and there exists $\tau > 0$ such that*

$$f(z) \geq f(x_*) + \tau\|z - x_*\|^2, \ \forall\ z \in \Omega, \tag{34}$$

then we can find a constant $\theta \in (0,1)$ such that

$$f(x_k) - f_* \leq \theta^k (f(x_0) - f_*), \ \forall\ k \geq 0. \tag{35}$$

Proof We will show that there exists $\nu \in (0,1)$ such that

$$f(x_{k+1}) - f_* \leq \nu(C_k - f_*). \tag{36}$$

Let ω satisfies that

$$0 < \omega < \min\left\{\frac{\alpha_{\max}}{\bar{\lambda}\sigma}, \frac{1}{L_f}, \frac{\tau\alpha_{\min}}{L_f}\right\},$$

where $\bar{\lambda}$ is given by (15). We consider two cases.

Case 1. $\|d_k\|^2 \geq \omega(C_k - f_*)$. By Lemma 2 and the right inequality of (29), we have

$$\frac{\alpha_{\max}}{\sigma}(C_k - f(x_{k+1})) \geq \lambda_k\|d_k\|^2 \geq \bar{\lambda}\omega(C_k - f_*),$$

which implies that

$$f(x_{k+1}) - f_* \leq \left(1 - \frac{\bar{\lambda}\sigma\omega}{\alpha_{\max}}\right)(C_k - f_*).$$

Recalling that $\omega < \dfrac{\alpha_{\max}}{\bar{\lambda}\sigma}$, we get (36) by setting $\nu = 1 - \dfrac{\bar{\lambda}\sigma\omega}{\alpha_{\max}} < 1$.

Case 2. $\|d_k\|^2 < \omega(C_k - f_*)$. It follows from (34) that

$$\beta_k = \frac{1}{2\alpha_{\min}}\|z_k - x_*\|^2 \leq \frac{1}{2\tau\alpha_{\min}}(f(z_k) - f_*) \leq b_5(C_k - f_*), \tag{37}$$

where $b_5 = \dfrac{1}{2\tau\alpha_{\min}}$. Combining the first inequality of (27) and (37), we have

$$\begin{aligned} f(x_{k+1}) &\leq (1 - \lambda_k\delta)f(z_k) + \lambda_k\delta f_* + \lambda_k\left(b_5\delta^2 + \frac{\omega L_f}{2}\right)(C_k - f_*) \\ &\leq C_k + \lambda_k\left(b_5\delta^2 - \delta + \frac{\omega L_f}{2}\right)(C_k - f_*), \end{aligned} \tag{38}$$

Subtracting f_* from each side of (38) to obtain

$$f(x_{k+1}) - f_* \leq \left[1 + \lambda_k \left(b_5\delta^2 - \delta + \frac{\omega L_f}{2}\right)\right](C_k - f_*),\ \forall\ \delta \in [0,1]. \quad (39)$$

We need to show the minimum of $h(\delta) = 1 + \lambda_k \left(b_5\delta^2 - \delta + \frac{\omega L_f}{2}\right)$ over $[0,1]$ is less than 1. In fact, $h(\delta)$ has a unique minimizer

$$\delta_{\min} = \min\left\{1, \frac{1}{2b_5}\right\}.$$

When $\delta_{\min} = 1$, $b_5 \leq \frac{1}{2}$. Since $\omega < \frac{1}{L_f}$, we have

$$h(\delta_{\min}) = 1 + \lambda_k \left(b_5 - 1 + \frac{\omega L_f}{2}\right) \leq 1 - \bar{\lambda}\frac{1 - \omega L_f}{2} < 1. \quad (40)$$

When $\delta_{\min} < 1$, using $\omega < \frac{\tau\alpha_{\min}}{L_f}$, we deduce

$$h(\delta_{\min}) = 1 + \lambda_k \left(\frac{1}{4b_5} - \frac{1}{2b_5} + \frac{\omega L_f}{2}\right) \leq 1 - \bar{\lambda}\left(\frac{1}{4b_5} - \frac{\omega L_f}{2}\right) < 1. \quad (41)$$

Therefore, (36) holds with $\nu = h(\delta_{\min})$. Combining (22), the first equality in (33), and (36), we have

$$\begin{aligned} C_{k+1} - f_* &\leq \frac{\gamma_k Q_k (C_k - f_*) + \nu(C_k - f_*)}{Q_{k+1}} \\ &= \left(1 - \frac{1-\nu}{Q_{k+1}}\right)(C_k - f_*) \\ &\leq [1 - (1 - \gamma_{\max})(1 - \nu)](C_k - f_*), \end{aligned}$$

which implies that

$$f(x_k) - f_* \leq C_k - f_* \leq \theta^k (f(x_0) - f_*),$$

where $\theta = 1 - (1 - \gamma_{\max})(1 - \nu)$. We complete the proof. □

Corollary 1 *Let $\{z_k\}$ be a sequence generated by Algorithm 1 with $\gamma_{\max} < 1$. If f is convex on Ω, then*

$$f(z_k) - f_* \leq \frac{a}{k+1},\ \forall\ k \geq 0.$$

Moreover, if the inequality (22) holds, then

$$f(z_k) - f_* \leq \theta^k (f(x_0) - f_*),\ \forall\ k \geq 0.$$

Here, a and θ are defined in Theorem 1.

4 Numerical experiments

In this section, we present numerical experiments of our QRPABB method. We implement our method in MATLAB. All the runs were carried out on a 3.10 GHz Core i5 PC with 4 GB of RAM under Windows 7.

4.1 Box constrained quadratic programming problems

In this subsection, we compare our QRPABB method with other five methods on some randomly generated box-constrained quadratic programming problems. Particularly, the following algorithms are tested:

A1: Algorithm 2 in [28].
A2: Our QRPABB method.
A3: Scheme (4.9) proposed by Nesterov in [47] using restart strategy every 100 iterations.
A4: PABB proposed by Dai and Fletcher [18].
A5: Projected BFGS[1] [37].
A6: Algorithm SPG proposed by Birgin, Marínez, and Raydan in [8,9].

We test the problems described in [28]:

$$\begin{aligned} \min\ & f(x) = (g^*)^T(x - x^*) + \frac{1}{2}(x - x^*)^T Q(x - x^*) \\ s.t.\ & x \in \Omega = \{x \in \mathbb{R}^n | 0 \le x \le u\}, \end{aligned}$$

where g^* and x^* are random vectors satisfying the optimality condition $P(x^* - g^*) = x^*$, u is a random vector with values in $[0, 1]$, and Q is a random matrix with eigenvalues in $[\mu, L]$, $L > \mu = 1$.

We set $n = 1,000$ for our test problems. The parameters for our QRPABB method are set to

$$\sigma = 10^{-4},\ \sigma_1 = 0.9,\ \sigma_2 = 0.5,\ \rho = 0.25,\ \eta = 2.$$

The initial BB stepsize is chosen to be $\alpha_0^{BB} = 1$ while $\alpha_{\max}$ is set to 10^{30} and $\alpha_{\min} = 1/\alpha_{\max}$. We use $\gamma_k = 0.1$ for all $k \ge 0$. Although our analysis allows us to use different γ_k during the iterative process, the above choice performs well in our test.

We stop the iteration of the QRPABB method if

$$\|z_k - P(z_k - \nabla f(z_k))\| \le 10^{-6}.$$

Other algorithms also use the norm of the projected gradient as the stopping condition. We also use 3,000 as the maximal number of iterations allowed for each algorithm.

We report the objective values versus iteration numbers in Figure 1 with $L_0 = L_f$ (left) and an overestimation of the Lipschitz constant $L_0 > L_f$ (right),

[1] The MATLAB code is available at http://www4.ncsu.edu/~ctk/

respectively. Clearly, our QRPABB method is much faster than other five algorithms. Although the projected BFGS gives the smallest objective value, it takes approximately twice as many iterations as our QRPABB method to reach the same level of accuracy.

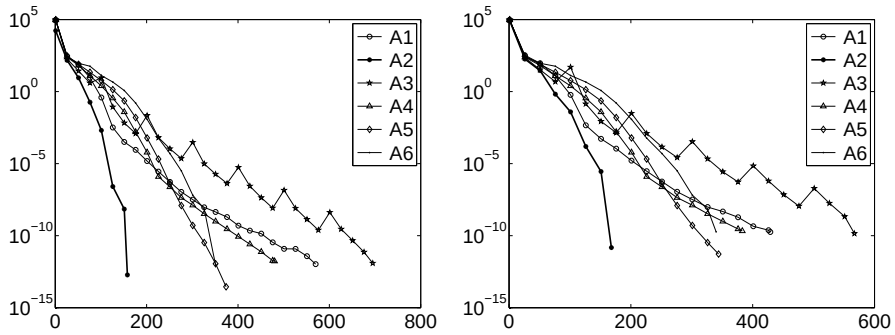

Fig. 1 Objective values versus iterations. Left: $L_0 = L_f$. Right: $L_0 > L_f$.

We evaluate the performance profiles [22] of the algorithms relative to the number of iterations and a laboriousness measure introduced in [28], where $ng + (nf - ng)/3$ with ng and nf being the numbers of gradient and function evaluations is used to illustrate the cost of an algorithm. Since Nesterov's method does not need to evaluating the function, we use $2ng + nf$ instead. It is worth noting that such a measure is based on the assumption that the oracle for evaluating the function spends one-third of the time used for computing both function and gradient.

A collection of 400 instances problems was generated with L_f from 10^2 to 10^5. For each test problem, L_0 is an overestimation for the Lipschitz constant, Q is a positive definite matrix, and strict complementarity holds at an optimal solution. Figure 2 shows that our QRPABB method has always had the least iterations and the best laboriousness for these well-behaved problems.

As we know, the absence of strict complementarity at optimal solutions will affect the performance of the algorithm. We consider a problem with strict complementarity at the optimal solution, a problem with half the optimal multipliers equal to zero, and a problem with null optimal multipliers. We compare algorithms A1, QRPABB, PABB, and projected BFGS for the three problems and present the results in left, center, and right of Figure 3, respectively. We observe that algorithm A1 and our QRPABB method were much less affected by the degeneracy than the other algorithms. Moreover, the QRPABB method is faster than algorithm A1.

Some researchers have observed from numerical results that the performance of the BB method was related to the relationship between the stepsizes and the eigenvalues of the Hessian, see [23] for example. We then consider a problem in which the Hessian eigenvalues are badly distributed: a large num-

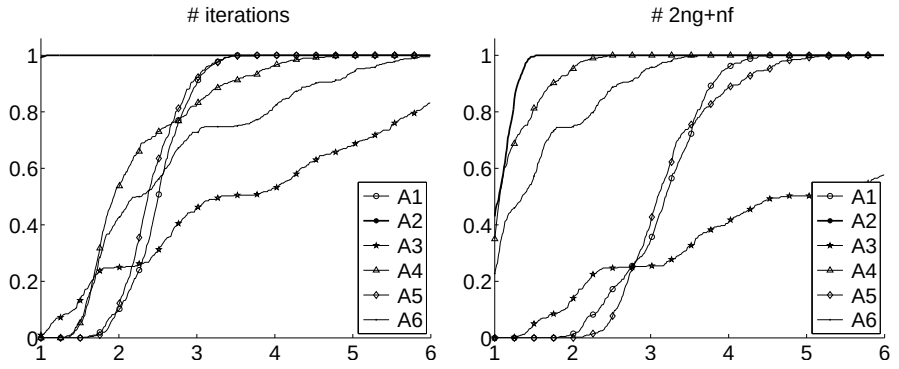

Fig. 2 Performance profile for number of iterations (left) and laboriousness measure (right).

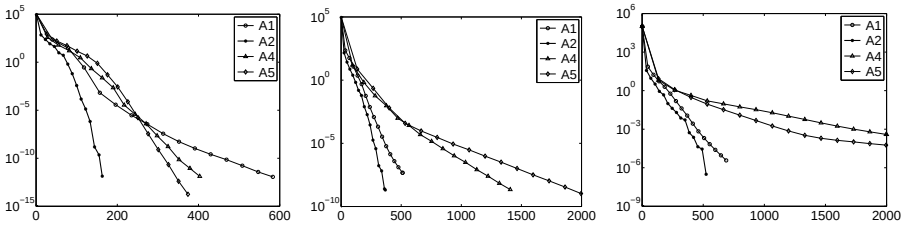

Fig. 3 Objective values versus iterations for examples with increasing degeneracy.

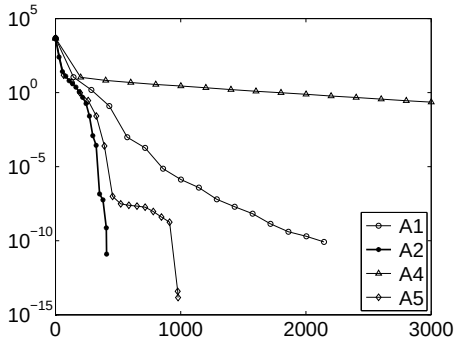

Fig. 4 Objective values versus iterations for a problem with badly distributed Hessian eigenvalues.

ber of eigenvalues in $[0, 1]$ and a small number in $[5000, 10000]$. We show the objective values versus iteration numbers in Figure 4. We observe that the PABB algorithm performs very badly, while the other methods were not much affected. Our QRPABB method is still the fastest one.

4.2 Nonnegative matrix factorization

Nonnegative matrix factorization (NMF) [39,48] has the form:

$$\min_{W\geq 0,\ H\geq 0} F(W,H) := \frac{1}{2}\|V-WH\|_F^2, \tag{42}$$

where $\|\cdot\|_F$ is the Frobenius norm, and $W \geq 0$ and $H \geq 0$ mean that all elements of W and H are nonnegative.

Recently, Vavasis [52] showed that problem (42) is NP-hard with respect to variables W and H. Fortunately, by resorting to the "block coordinate descent" method in bound constrained optimization [6], we can alternatively optimize one matrix factor with another fixed. In particular, the NMF problem (42) can be solved by the following alternating nonnegative least squares (ANLS) framework.

ANLS Framework for NMF

Step 1. Initialize $W^0 \in \mathbb{R}_+^{m\times r}$ and $H^0 \in \mathbb{R}_+^{r\times n}$. Set $k = 0$.
Step 2. Update the matrices in the following way until a convergence criterion is satisfied:

$$W^{k+1} = \arg\min_{W\geq 0} F(W,H^k), \tag{43}$$

$$H^{k+1} = \arg\min_{H\geq 0} F(W^{k+1},H). \tag{44}$$

Although the original problem (42) is non-convex, the subproblems (43) and (44) are convex problems for which optimal solutions can be found. However, the subproblems may have multiple optimal solutions because they are not strictly convex. Grippo and Sciandrone [30] proved that any limit point of the sequence $\{W_k, H_k\}$ generated by the ANLS framework is a stationary point of (42).

Now consider the subproblem (43). The gradient of $F(W,H^k)$ is given by

$$\nabla_W F(W,H^k) = (WH^k - V)(H^k)^T,$$

which is Lipschitz continuous with constant $L_W = \|H^k(H^k)^T\|_2$, see [31] for example. Therefore, we can apply the QRPABB method to solve the subproblem (43). Since $r << m,n$, the Lipschitz constant L_W is inexpensive to calculate. So we use L_W throughout the iterative process. All the parameters of our QRPABB method are set to same values as the former subsection except $\gamma_k = 0.85$. Notice that the role of matrices W and H is perfectly symmetric. Our QRPABB method can be directly translated into an update for H.

We compare the performance of the following algorithms:

(i) QRPABB method in the ANLS framework (QRPABB)
(ii) Nesterov's optimal method in the ANLS framework (NeNMF,[31])

(iii) Projected BB method (APBB2[2], [35])
(iv) Lin's projected gradient method (PG[3], [42])

Define the projected gradient of $F(W,H)$ with respect to W by

$$\nabla^P_W F(W,H) = \begin{cases} \nabla_W F(W,H)_{ij}, & (W)_{ij} > 0, \\ \min\{0, \nabla_W F(W,H)_{ij}\}, & (W)_{ij} = 0. \end{cases}$$

The KKT conditions of (42) can be written as

$$\nabla^P_H F(W,H) = 0, \ \nabla^P_W F(W,H) = 0,$$

where $\nabla^P_H F(W,H)$ is defined in the same way as $\nabla^P_W F(W,H)$.

We stop the algorithms if the approximate projected gradient norm satisfies

$$pgn := \|[\nabla^P_H F(W^k, H^{k-1}), \nabla^P_W F(W^k, H^k)^T]\|_F \le \epsilon \cdot pgn^0, \qquad (45)$$

where $\epsilon > 0$ is a tolerance and pgn^0 is the initial projected gradient norm.

For the subproblems (43) and (44), we stop the iterative procedure of the QRPABB method when

$$\|\nabla^P_W F(W^k, H^{k-1})\|_F \le \epsilon_W, \text{ and } \|\nabla^P_H F(W^k, H^k)\|_F \le \epsilon_H,$$

where

$$\epsilon_W = \epsilon_H = \max(10^{-3}, \epsilon) \cdot pgn^0. \qquad (46)$$

If QRPABB solves (43) without any iterations, we decrease the stopping tolerance by $\epsilon_W = 0.1\epsilon_W$. The same strategy is adopted to solve (44).

Firstly, we test the algorithms on random generated problems. Using MATLAB routines, we generate a random $m \times n$ matrix V. For each V, we generate 10 different random initial points and present the average results from using these initial values in Table 1. In this table, *iter* denotes the number of iterations needed when the termination criterion (45) was met. We denote the total number of sub-iterations for solving (43) and (44), the final value of projected gradient norm as defined in (45), the final value of $\|V - W^k H^k\|_F / \|V\|_F$, and the CPU time used at the termination of each algorithm, by *niter*, *pgn*, *residual*, and *time*, respectively.

From Table 1, we observe that for most test problems, QRPABB outperforms other three algorithms in terms of CPU time. However, the performance of the algorithms on computing the factorization of a $m \times n$ matrix with rank r differs from that of a $n \times m$ matrix with the same rank, which is obvious as the problem size grows.

We then test the algorithms on large size NMF problems to assess the affect of the size. Table 3 reports the average results using 10 different initial values. The notation "-T" means run the algorithm on the transpose matrix V^T. The numbers *witer* and *hiter* denote the total number of iterations for solving (43) and (44), respectively. We observe from this table that, when $m < n$, PG and

[2] http://homepages.umflint.edu/~lxhan/software.html
[3] http://www.csie.ntu.edu.tw/~cjlin/nmf/index.html

Table 1 Experimental results on synthetic datasets (dense matrices) with $\epsilon = 10^{-7}$. All methods were executed with the same initial values, and the average results using 10 different initial values are presented.

$(m\ n\ r)$	Alg	iter	niter	pgn	time	residual
(25,50,5)	PG	607.0	10941.2	3.29E-05	0.49	0.3889
	NeNMF	458.4	20450.4	2.89E-05	0.26	0.3890
	APBB2	534.3	7506.9	2.93E-05	0.28	0.3890
	QRPABB	547.5	4622.2	2.96E-05	0.19	0.3889
(50,25,5)	PG	220.0	4377.3	2.92E-05	0.20	0.3880
	NeNMF	151.4	5787.8	2.86E-05	0.08	0.3880
	APBB2	230.5	2407.1	2.47E-05	0.10	0.3880
	QRPABB	182.9	1328.5	2.14E-05	0.06	0.3880
(50,100,5)	PG	600.6	13321.0	8.47E-05	0.73	0.4361
	NeNMF	586.9	27549.0	8.37E-05	0.52	0.4361
	APBB2	659.3	9778.9	7.02E-05	0.46	0.4361
	QRPABB	608.4	5323.5	6.44E-05	0.29	0.4361
(100,50,5)	PG	643.6	10935.7	7.25E-05	0.61	0.4388
	NeNMF	399.7	18220.0	8.36E-05	0.36	0.4388
	APBB2	538.1	6775.6	6.60E-05	0.34	0.4388
	QRPABB	404.3	3345.0	7.62E-05	0.19	0.4388
(100,200,10)	PG	565.9	17411.2	7.91E-04	1.80	0.4402
	NeNMF	753.5	32871.8	7.62E-04	2.00	0.4402
	APBB2	603.8	10538.8	6.05E-04	0.97	0.4403
	QRPABB	504.3	5423.5	4.50E-04	0.59	0.4402
(200,100,10)	PG	1326.9	65375.0	7.36E-04	8.10	0.4393
	NeNMF	543.1	24968.9	8.04E-04	1.56	0.4393
	APBB2	685.8	10706.6	6.49E-04	1.11	0.4393
	QRPABB	660.5	6500.4	7.16E-04	0.79	0.4393
(100,300,20)	PG	771.4	16038.5	3.50E-03	4.04	0.4068
	NeNMF	905.8	36400.4	1.69E-03	5.72	0.4067
	APBB2	318.8	5945.7	2.10E-03	1.04	0.4069
	QRPABB	449.1	4937.4	2.88E-03	1.10	0.4069
(300,100,20)	PG	544.6	14225.3	3.54E-03	4.67	0.4062
	NeNMF	960.9	40425.1	3.55E-03	6.65	0.4060
	APBB2	461.0	7397.6	2.99E-03	1.70	0.4062
	QRPABB	490.3	4870.1	2.24E-03	1.36	0.4062
(300,500,25)	PG	558.9	40842.7	1.61E-02	21.23	0.4485
	NeNMF	933.9	33660.2	1.60E-02	11.14	0.4485
	APBB2	198.1	3723.8	1.08E-02	1.66	0.4487
	QRPABB	246.9	3368.9	1.41E-02	1.85	0.4487
(500,300,25)	PG	1476.3	51838.4	1.53E-02	31.62	0.4486
	NeNMF	1081.6	36055.2	1.59E-02	11.52	0.4487
	APBB2	245.3	4015.4	1.22E-02	2.11	0.4489
	QRPABB	249.8	3005.7	1.53E-02	1.91	0.4489
(500,1000,50)	PG	688.3	12946.3	1.03E-01	26.27	0.4460
	NeNMF	471.7	15567.8	1.16E-01	18.49	0.4460
	APBB2	50.4	1450.7	9.03E-02	2.03	0.4473
	QRPABB	491.0	5473.2	8.94E-02	11.37	0.4460
(1000,500,50)	PG	720.4	30681.5	1.01E-01	77.40	0.4469
	NeNMF	1009.1	31254.3	1.17E-01	34.62	0.4469
	APBB2	63.8	1475.8	1.03E-01	2.87	0.4482
	QRPABB	59.5	812.9	1.09E-01	1.83	0.4481

NeNMF are much faster in finding the factorization of V than V^T. If $m > n$, the performance of PG and NeNMF is better in factorizing V^T. The APBB2 exhibits similar property. However, our QRPABB method seems faster for factorizing V with $m > n$. One possible reason is the different stopping tolerances in (46) for solving the two subproblems since the tolerances depending on the norm of the initial projected gradient. Another factor is the algorithms may

generate different sequences due to different update strategies. Finally, since the NMF problem (42) is not convex, the algorithms may converge to different stationary points. We can see that the sub-iterations for solving the subproblems vary greatly in factorizing V and V^T. These factors also attribute to the differences in values of *residual*.

Table 2 Experimental results on synthetic datasets (dense matrices) with $\epsilon = 10^{-7}$. All methods were executed with the same initial values, and the average results using 10 different initial values are presented.

(m, n, r)	Alg	iter	witer	hiter	pgn	time	residual
(1000,2000,50)	PG	318.2	22708.8	5152.4	3.19E-01	97.87	0.4710
	NeNMF	364.0	6526.7	5239.2	3.29E-01	27.81	0.4710
	APBB2	85.4	1530.2	425.6	2.74E-01	6.47	0.4718
	QRPABB	295.5	2255.4	1263.6	2.74E-01	15.86	0.4710
	PG-T	692.9	10972.9	14271.4	2.82E-01	128.11	0.4710
	NeNMF-T	853.1	13661.1	14165.9	3.31E-01	68.60	0.4709
	APBB2-T	45.3	931.5	334.5	3.15E-01	6.12	0.4718
	QRPABB-T	30.3	310.2	117.8	2.37E-01	2.22	0.4719
(1000,5000,100)	PG	94.6	3443.6	522.0	2.02E+00	55.95	0.4615
	NeNMF	121.7	2079.3	2220.9	2.10E+00	57.28	0.4615
	APBB2	75.1	2197.6	453.3	1.56E+00	31.75	0.4615
	QRPABB	37.0	559.9	140.1	1.89E+00	10.24	0.4645
	PG-T	170.1	6249.2	4506.1	1.98E+00	337.07	0.4613
	NeNMF-T	700.3	11246.6	13362.2	2.13E+00	290.91	0.4607
	APBB2-T	101.9	1176.7	978.5	1.60E+00	51.07	0.4619
	QRPABB-T	15.5	185.7	57.8	1.39E+00	7.93	0.4644
(2000,1000,50)	PG	841.2	12797.0	16203.1	2.70E-01	148.09	0.4712
	NeNMF	827.4	13163.9	13784.0	3.32E-01	66.12	0.4712
	APBB2	44.8	930.6	336.6	3.02E-01	6.08	0.4720
	QRPABB	30.7	313.6	119.2	2.39E-01	2.23	0.4721
	PG-T	291.7	21537.4	4738.3	3.20E-01	91.75	0.4713
	NeNMF-T	457.4	8675.1	6405.6	3.30E-01	35.15	0.4712
	APBB2-T	86.7	1579.5	453.7	2.72E-01	6.71	0.4719
	QRPABB-T	309.1	2327.7	1303.2	2.48E-01	16.40	0.4712
(5000,1000,100)	PG	181.6	6522.8	4529.9	1.98E+00	353.39	0.4616
	NeNMF	670.6	10973.8	13158.3	2.13E+00	284.94	0.4611
	APBB2	88.5	1093.7	836.3	1.76E+00	46.78	0.4626
	QRPABB	15.3	184.2	57.0	1.49E+00	7.82	0.4648
	PG-T	93.7	3409.5	525.2	1.99E+00	55.85	0.4618
	NeNMF-T	122.8	2092.1	2238.4	2.09E+00	57.79	0.4618
	APBB2-T	76.7	2291.3	466.3	1.65E+00	32.82	0.4618
	QRPABB-T	56.7	889.8	216.7	1.69E+00	15.87	0.4631

Based on the above observation, when $m > n$, we will run PG, NeNMF, and APBB2 on the transpose matrix V^T. If $m < n$, our QRPABB method will be applied to V^T. In order to improve the accuracy of our QRPABB method, we will decrease the stopping tolerance with a factor 0.1 if QRPABB solves (43) or (44) less than 5 iterations. It is worth noting that for random problems, such a strategy seems no meaningful improvement. Moreover, the ANLS framework will be run at least 10 iterations for all the algorithms to decrease the objective.

We test the algorithms on the ORL[4] and CBCL[5] image databases. The CBCL image database contains 2,429 facial images each of which has 19×19 pixels. We obtain the $361 \times 2,429$ matrix V by representing each image as a row of V. The ORL image database consists of 400 facial images of 40 different people. Each face image has 92×112 pixels. Similar as the CBCL database, we obtain an $10,304 \times 400$ matrix. We report the average results of 10 different randomly generated initial iterates in Table 3 with $\epsilon = 10^{-7}$ in (45). The algorithms compute fairly good solutions with regard to the projected gradient norms. Our QRPABB method is faster than other three methods.

Table 3 Experimental results on the CBCL and ORL databases with $\epsilon = 10^{-7}$. All methods were executed with the same initial values, and the average results using 10 different initial values are presented.

$(m\ n\ r)$	Alg	iter	niter	pgn	time	residual
(361,2429,49)	PG	82.4	11427.3	2.03E-01	16.13	0.1947
	NeNMF	121.1	4290.3	2.07E-01	10.18	0.1946
	APBB2	97.8	4007.9	1.79E-01	6.90	0.1955
	QRPABB	67.7	1453.8	1.49E-01	4.41	0.1955
(10304,400,25)	PG	10.0	476.1	3.14E-01	5.91	0.2035
	NeNMF	12.0	517.3	3.10E-01	3.28	0.2070
	APBB2	14.5	329.5	2.59E-01	1.74	0.1883
	QRPABB	10.0	143.8	2.49E-01	1.17	0.1860

Finally, we test the algorithms on the Reuters-21578 [41] and TDT-2 [15] text corpus [6]. There are 21,578 documents in 135 categories contained in the Reuters-21578 corpus. We discard those documents with multiple category labels and obtained 8,293 documents in 65 categories. The corpus is represented by an $18,933 \times 8,293$-dimension matrix with 18,933 distinct terms. The TDT-2 corpus contains 11,201 on-topic documents, which are collected from ABC, CNN, VOA, NYT, PRI, and APW, classified into 96 semantic categories. We remove those documents appearing in two or more categories and keep the largest 30 categories which leave us with 9,394 documents. After preprocessing, this corpus is represented by a $36,771 \times 9,394$-dimension matrix. The QRPABB method again outperforms other methods in terms of CPU time. Although NeNMF obtains the smallest projected gradient norm, our method yields a solution with smaller residual which implies the reconstruction obtained by the QRPABB method is better.

5 Conclusions

We have presented a quadratic regularization projected alternating Barzilai–Borwein (QRPABB) method for solving constrained minimization problems

[4] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[5] http://cbcl.mit.edu/software-datasets/FaceData2.html.

[6] Both Reuters-21578 corpus and TDT-2 corpus in MATLAB format are available at http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html

Table 4 Experimental results on the Reuters-21578 and TDT2 datasets with $\epsilon = 10^{-7}$. All methods were executed with the same initial values, and the average results using 10 different initial values are presented.

$(m\ n\ r)$	Alg	iter	niter	pgn	time	residual
(18933,8293,50)	PG	10.0	448.5	8.47E+00	15.72	0.9395
	NeNMF	10.0	250.0	1.26E-01	5.77	0.9242
	APBB2	12.9	285.1	6.56E+00	8.58	0.9337
	QRPABB	10.0	118.5	5.52E+00	4.78	0.9236
(36771,9394,100)	PG	10.0	493.5	4.54E+00	47.57	0.9294
	NeNMF	10.0	271.0	1.12E-01	21.80	0.9095
	APBB2	10.0	318.7	4.51E+01	24.58	0.9329
	QRPABB	10.0	115.9	3.20E+00	17.95	0.9092

where the objective has Lipschitz continuous gradient. It converges to a constrained stationary point for general nonconvex objective functions. We established sublinear and R-linear convergence of the QRPABB method for convex and strongly convex objective functions, respectively. Our method requires low memory and is extremely easy to implement. Experimental comparisons with other BB-like methods and optimal first-order methods on box-constrained quadratic problems show that our QRPABB method is very promising. Moreover, the practical potential of the proposed method is demonstrated by applying it to nonnegative matrix factorization problems. The QRPABB method is expected to be efficient provided that projections are not complicated.

Acknowledgements The authors are very grateful to Professor Elizabeth W. Karas of Federal University of Paraná for providing us the codes of [28] and helpful comments on the paper. The authors also would like to thank Dr. Naiyang Guan of National University of Defense Technology for providing us the codes of [31] and Bo Jiang of Nanjing Normal University for his helpful comments on the paper. This work was supported by the National Natural Science Foundation of China (NNSFC) under Grant No. 61072144 and No. 61179040 and the Fundamental Research Funds for the Central Universities No. K50513100007.

References

1. Auslender, A., Teboulle, M.: Interior gradient and proximal methods for convex and conic optimization. SIAM J. Optim. **16**(3), 697–725 (2006)
2. Barzilai, J., Borwein, J.M.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**(1), 141–148 (1988)
3. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Imaging Sci. **2**(1), 183–202 (2009)
4. Becker, S.R., Candès, E.J., Grant, M.C.: Templates for convex cone problems with applications to sparse signal recovery. Math. Program. Comput. **3**(3), 165–218 (2011)
5. Bertsekas, D.P.: On the Goldstein-Levitin-Polyak gradient projection method. IEEE Trans. Automat. Control **21**(2), 174–184 (1976)
6. Bertsekas, D.P.: Nonlinear Programming, 2nd ed. Athena Scientific, Belmont (1999)
7. Bian, W., Chen, X.: Worst-case complexity of smoothing quadratic regularization methods for non-Lipschitzian optimization. SIAM J. Optim. **23**(3), 1718–1741 (2013)
8. Birgin, E.G., Martínez, J.E.M., Raydan, M.: Nonmonotone spectral projected gradient methods on convex sets. SIAM J. Optim. **10**(4), 1196–1211 (2000)
9. Birgin, E.G., Martínez, J.M., Raydan, M.: Algorithm 813: SPG–software for convex-constrained optimization. ACM Trans. Math. Software **27**(3), 340–349 (2001)

10. Birgin, E.G., Martínez, J.M., Raydan, M.: Spectral projected gradient methods: Review and perspectives. http://www.ime.usp.br/~egbirgin/publications/bmr5.pdf (2012)
11. Calamai, P.H., Moré, J.J.: Projected gradient methods for linearly constrained problems. Math. Program. **39**(1), 93–116 (1987)
12. Cartis, C., Gould, N.I.M., Toint, P.L.: Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. Math. Program. **127**(2), 245–295 (2011)
13. Cartis, C., Gould, N.I.M., Toint, P.L.: Adaptive cubic regularisation methods for unconstrained optimization. Part II: worst-case function- and derivative-evaluation complexity. Math. Program. **130**(2), 295–319 (2011)
14. Cartis, C., Gould, N.I., Toint, P.L.: On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. SIAM J. Optim. **21**(4), 1721–1739 (2011)
15. Cieri, C., Graff, D., Liberman, M., Martey, N., Strassel, S.: The TDT-2 text and speech corpus. In: Proceedings of the DARPA Broadcast News Workshop, pp. 57–60 (1999)
16. Dai, Y.H.: On the nonmonotone line search. J. Optim. Theory Appl. **112**(2), 315–330 (2002)
17. Dai, Y.H., Fletcher, R.: On the asymptotic behaviour of some new gradient methods. Math. Program. **103**(3), 541–559 (2005)
18. Dai, Y.H., Fletcher, R.: Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. Numer. Math. **100**(1), 21–47 (2005)
19. Dai, Y.H., Hager, W.W., Schittkowski, K., Zhang, H.: The cyclic Barzilai-Borwein method for unconstrained optimization. IMA J. Numer. Anal. **26**(3), 604–627 (2006)
20. 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)
21. Dai, Y.H., Zhang, H.: Adaptive two-point stepsize gradient algorithm. Numer. Algor. **27**(4), 377–385 (2001)
22. Dolan, E.D., Moré, J.J.: Benchmarking optimization software with performance profiles. Math. Program. **91**(2), 201–213 (2002)
23. Fletcher, R.: Low storage methods for unconstrained optimization. Lectures in Applied Mathematics (AMS) **26**, 165–179 (1990)
24. Fletcher, R.: On the Barzilai-Borwein method. In: Optimization and Control with Applications (2005)
25. Gafni, E.M., Bertsekas, D.P.: Two-metric projection methods for constrained optimization. SIAM J. Control Optim. **22**(6), 936–964 (1984)
26. Goldstein, A.A.: Convex programming in Hilbert space. Bulletin of the American Mathematical Society **70**(5), 709–710 (1964)
27. Gonzaga, C.C., Karas, E.W.: Fine tuning Nesterov's steepest descent algorithm for differentiable convex programming. Math. Program. **138**(1-2), 141–166 (2013)
28. Gonzaga, C.C., Karas, E.W., Rossetto, D.R.: An optimal algorithm for constrained differentiable convex optimization. SIAM J. Optim. **23**(4), 1939–1955 (2013)
29. Grippo, L., Lampariello, F., Lucidi, S.: A nonmonotone line search technique for Newton's method. SIAM J. Numer. Anal. **23**(4), 707–716 (1986)
30. Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. Operat. Res. Lett. **26**(3), 127–136 (2000)
31. Guan, N., Tao, D., Luo, Z., Yuan, B.: NeNMF: An optimal gradient method for nonnegative matrix factorization. IEEE Trans. Signal Process. **60**(6), 2882–2898 (2012)
32. Hager, W.W., Park, S.: The gradient projection method with exact line search. J. Glob. Optim. **30**(1), 103–118 (2004)
33. Hager, W.W., Phan, D.T., Zhang, H.: Gradient-based methods for sparse recovery. SIAM J. Imaging Sci. **4**(1), 146–165 (2011)
34. Hager, W.W., Zhang, H.: A new active set algorithm for box constrained optimization. SIAM J. Optim. **17**(2), 526–557 (2006)
35. Han, L., Neumann, M., Prasad, U.: Alternating projected Barzilai-Borwein methods for nonnegative matrix factorization. Electron Trans. Numer. Anal. **36**(6), 54–82 (2009)
36. Huang, Y., Liu, H., Zhou, S.: A Barzilai-Borwein type method for stochastic linear complementarity problems. Numer. Algor. (2013). doi: 10.1007/s11075-013-9803-y
37. Kelley, C.T.: Iterative methods for optimization. SIAM, Philadelphia (1999).

38. Lan, G., Lu, Z., Monteiro, R.D.: Primal-dual first-order methods with $O(1/\epsilon)$ iteration-complexity for cone programming. Math. Program. **126**(1), 1–29 (2011)
39. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**(6755), 788–791 (1999)
40. Levitin, E.S., Polyak, B.T.: Constrained minimization methods. USSR Computational mathematics and mathematical physics **6**(5), 1–50 (1966)
41. Lewis, D.D., Yang, Y., Rose, T.G., Li, F.: RCV1: A new benchmark collection for text categorization research. J. Mach Learn. Res. **5**, 361–397 (2004)
42. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Comput. **19**(10), 2756–2779 (2007)
43. Luo, Z.Q., Tseng, P.: On the linear convergence of descent methods for convex essentially smooth minimization. SIAM J. Control Optim. **30**(2), 408–425 (1992)
44. Nemirovsky, A.S., Yudin, D.B.: Problem complexity and method efficiency in optimization. John Wiley, New York (1983)
45. Nesterov, Y.: A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Soviet Mathematics Doklady. **27**, pp. 372–376 (1983)
46. Nesterov, Y.: Introductory lectures on convex optimization: A basic course. Springer, Boston (2004)
47. Nesterov, Y.: Gradient methods for minimizing composite functions. **140**(1), 125–161 (2013). Math. Program.
48. Paatero, P., Tapper, U.: Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5**(2), 111–126 (1994).
49. Raydan, M.: On the Barzilai and Borwein choice of steplength for the gradient method. IMA J. Numer. Anal. **13**(3), 321–326 (1993)
50. Raydan, M.: The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. SIAM J. Optim. **7**(1), 26–33 (1997)
51. Tseng, P.: On accelerated proximal gradient methods for convex-concave optimization. http://www.math.washington.edu/~tseng/papers/apgm.pdf (2009)
52. Vavasis, S.: On the complexity of nonnegative matrix factorization. SIAM J. Optim. **20**(3), 1364–1377 (2009).
53. Zhang, H., Hager, W.W.: A nonmonotone line search technique and its application to unconstrained optimization. SIAM J. Optim. **14**(4), 1043–1056 (2004)