

## TWO MODIFIED BARZILAI-BORWEIN PROJECTION METHODS WITH APPLICATIONS TO IMAGE RESTORATION\*

Peiting Gao<sup>1,2</sup>, Ruiping Zhang<sup>1,†</sup> and Jianchao Bai<sup>3,†</sup>

**Abstract** In this paper, we propose two modified Barzilai-Borwein projection methods for solving constrained monotone nonlinear equations. These methods are proved to satisfy the Quasi-Newton equation and sufficient descent property independent of any line search. Performance comparisons show that our proposed methods are more competitive and efficient for solving these equations. Our projection methods are also used to restore images to demonstrate their effectiveness.

**Keywords** Monotone nonlinear equations, projection method, quasi-newton equation, global convergence, image restoration.

**MSC(2010)** 65F10, 90C52, 65K05.

### 1. Introduction

Let  $R^n$  be the set of  $n$  dimensional real column vectors equipped with Euclidean norm  $\|\cdot\|$ . We consider the problem of finding  $x \in \Omega$  such that

$$F(x) = 0, \quad (1.1)$$

where  $\Omega$  is a non-empty closed convex set in  $R^n$  and  $F : R^n \rightarrow R^n$  is a continuous nonlinear mapping satisfying the following monotone inequality

$$\langle F(x) - F(y), x - y \rangle \geq 0, \quad \forall x, y \in R^n. \quad (1.2)$$

A lot of practical problems can be transformed into the form of constrained monotone nonlinear equation (1.1), for example, power flow equations [19, 34], chemical systems [27], power flow [20], complementarity problems [11], signal recovery [18] and image restoration [37]. Hence, developing numerical methods with convergence theories for (1.1) has attracted much attention. As a pioneer, Solodov and Svaiter [32] firstly proposed a linesearch-based projection method to solve (1.1). Based on [32], Wang et al. [35] introduced a modified projection method with

---

<sup>†</sup>The corresponding authors.

<sup>1</sup>Shanxi Provincial People's Hospital, No. 29 Shuangtasi Street, Yingze District, 030001 Taiyuan, Shanxi Province, China

<sup>2</sup>College of Artificial Intelligence, Taiyuan University of Technology, No. 209 University Street, Yuci District, 030600 Jinzhong, Shanxi Province, China

<sup>3</sup>School of Mathematics and Statistics, Northwestern Polytechnical University, No. 1 Dongxiang Road, Chang'an District, 710129 Xi'an, Shaanxi Province, China

\*This research was partially supported by the National Natural Science Foundation of China (62302331, 12471298), the Natural Science Foundation of Shanxi Province (202203021212255), and the Shaanxi Fundamental Science Research Project for Mathematics and Physics (23JSQ031).

Email: [gaopeitng3680@163.com](mailto:gaopeitng3680@163.com)(P. Gao), [zrp\\_7142@sxmu.edu.cn](mailto:zrp_7142@sxmu.edu.cn)(R. Zhang), [jianchaobai@nwpu.edu.cn](mailto:jianchaobai@nwpu.edu.cn)(J. Bai)

global convergence guarantees. By combining the Quasi-Newton method [10] with the projection method [33], Zhou-Li proposed a Quasi-Newton projection method [39] and its variant [40] to solve (1.1). These methods gained much attention of researchers due to the local super-linear convergence property. However, it is not suitable to apply them to solve large-scale constrained monotone nonlinear equation (1.1), since the massive storage of the corresponding Jacobian matrix or its approximation is needed at each iteration. Compared with above projection methods, the Conjugate Gradient (CG) method needs low storage requirement and is very suitable for dealing with large-scale constrained monotone nonlinear equation (1.1). Advanced CG-type methods include the three-term CG-based projection method with Dai-Liao conjugacy condition [1, 14, 17, 22–24, 26], the piecewise CG-based projection method [15], three-term CG-based projection method with Quasi-Newton equation [14, 18, 39], spectral CG-based projection method [2, 31], and modified CG-based projection method [28].

To further improve the performance of CG method, Barzilai and Borwein (BB) introduced the following classical yet useful stepsizes [4]:

$$\alpha_k^{BB1} = \frac{s_{k-1}^\top s_{k-1}}{s_{k-1}^\top y_{k-1}}, \quad \alpha_k^{BB2} = \frac{s_{k-1}^\top y_{k-1}}{y_{k-1}^\top y_{k-1}}, \quad (1.3)$$

where  $s_{k-1} = x_k - x_{k-1}$ ,  $y_{k-1} = g_k - g_{k-1}$  and  $g_k = F(x_k) := F_k$ . Obviously, according to (1.2) and the denominator of (1.3) being nonzero,  $s_{k-1}^\top y_{k-1} > 0$  holds and hence  $\alpha_k^{BB1} \geq \alpha_k^{BB2}$ . So,  $\alpha_k^{BB1}$  is called the long BB step length, while  $\alpha_k^{BB2}$  is called the short BB step length.

With the development of BB step size, more and more researchers had paid their attention to applying them for different optimization methods. In 2005, Dai et al. [7] studied two projected BB methods for large-scale box-constrained quadratic programming and proved their global convergence under Grippo-Lampariello-Lucidi nonmonotone linear search. Later, Dai et al. [8] proposed a cyclic BB method and proved its local linearly convergence. In 2019, Dai et al. [6] introduced the stabilized BB method to overcome the shortcoming of long steps and non-convergence. Dai et al. [9] also exploited a convex combination of the long BB step size and short BB step size in a family of spectral gradient method. In 2021, Qu et al. [30] proposed a projection method with alternate BB step-size for general variational inequalities and the global convergence was proved under some strong conditions. Recently, by combining nonmonotone line searches and gradient projection techniques, an accelerated BB method was proposed [21] for solving bound constrained optimization problems. Notice that, the above methods related to BB step length are dissatisfied with the so-called Quasi-Newton equation.

It is well-known that the Quasi-Newton equation plays a key role in improving the performance of methods. Gao and He [14] proposed a derivative-free three-term CG-based method satisfying with Quasi-Newton equation. Inspired by [14], Gao et al. [18] also proposed an effective three-term CG-based method satisfying with Quasi-Newton equation. However, to the best of our knowledge, three-term BB-type projection methods with Quasi-Newton equation have not been investigated so far. Motivated by this issue, we propose two BB projection methods with Quasi-Newton equation for solving (1.1). Contributions of this paper are summerized as follows:

- We design a piecewise search direction with Quasi-Newton equation to ensure the global convergence of the proposed BB projection methods.
- The BB step-size is introduced to improve the step size, which can seek suitable step length easily.

- We prove the boundedness of BB step-size and search direction by using the modifying the parameter construction  $\bar{y}_k$ .
- Under the modified step-size, the global convergence is proved with proper assumption and the convergence rate of this method is proved with an extra local error bound condition.
- The numerical experiments demonstrate that our method is effective and promising for monotone nonlinear equations and image restoration. Furthermore, in addition to comparing the signal-to-noise ratio (SNR) of the images, we also evaluated the peak signal-to-noise ratio (PSNR) to show the effectiveness of our methods.

The remainder of this paper is organized as follows. In Section 2, we present two piecewise BB projection methods for solving (1.1). In Section 3, the convergence analysis of our method is discussed. In Section 4, experiments are reported to show the efficiency of the proposed methods. In Section 5, we apply the proposed methods to restore images by comparing some existing methods. Finally, we conclude the paper in Section 6.

## 2. Development of our algorithm

Before showing our proposed BB projection methods with Quasi-Newton equation to solve (1.1), we recall the BB method for unconstrained optimization problem:

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

where  $f : R^n \rightarrow R$  is continuously differentiable and  $\nabla f(x)$  is simply denoted by  $F(x)$  in this paper. The BB method reads the following iteration

$$x_{k+1} = x_k + \alpha_k d_k, \quad k = 0, 1, \dots, \quad (2.2)$$

where  $\alpha_k$  is obtain by (1.3) and  $d_k = -F_k$ . Now, the search direction  $d_k$  is transformed into the following three-term form to satisfy the Quasi-Newton equation:

$$\begin{aligned} d_k &= -F_k + \beta_k s_{k-1} + \gamma_k y_{k-1} \\ &= -F_k + \frac{F_k^\top y_{k-1}}{F_k^\top y_{k-1}} \beta_k s_{k-1} + \frac{F_k^\top y_{k-1}}{F_k^\top y_{k-1}} \gamma_k y_{k-1} \\ &= - \left( I - \beta_k \frac{s_{k-1} y_{k-1}^\top}{y_{k-1}^\top F_k} - \gamma_k \frac{y_{k-1} y_{k-1}^\top}{y_{k-1}^\top F_k} \right) F_k \\ &= -Q_k F_k, \end{aligned} \quad (2.3)$$

where  $Q_k = I - \beta_k \frac{s_{k-1} y_{k-1}^\top}{y_{k-1}^\top F_k} - \gamma_k \frac{y_{k-1} y_{k-1}^\top}{y_{k-1}^\top F_k}$  denotes a relative matrix, and  $I$  denotes the identity matrix with proper dimensions.

Gao and He [14] introduced a derivative-free three-term conjugate gradient-based method satisfying with Quasi-Newton equation and the search direction is showed as

$$\begin{aligned} d_k &= -Q_k F_k \\ &= - \left( \frac{s_{k-1}^\top \omega_{k-1}}{\|s_{k-1}\|^2} I - \frac{s_{k-1} \omega_{k-1}^\top}{\|s_{k-1}\|^2} + \frac{s_{k-1} s_{k-1}^\top}{s_{k-1}^\top \omega_{k-1}} + \frac{\|\omega_{k-1}\|^2 s_{k-1} s_{k-1}^\top}{\|s_{k-1}\|^2 s_{k-1}^\top \omega_{k-1}} - t \frac{\omega_{k-1} s_{k-1}^\top}{\|s_{k-1}\|^2} \right) F_k \\ &= - \frac{s_{k-1}^\top \omega_{k-1}}{\|s_{k-1}\|^2} F_k + \left( \frac{F_k^\top \omega_{k-1}}{\|s_{k-1}\|^2} - \frac{F_k^\top s_{k-1}}{s_{k-1}^\top \omega_{k-1}} - \frac{\|\omega_{k-1}\|^2 F_k^\top s_{k-1}}{\|s_{k-1}\|^2 s_{k-1}^\top \omega_{k-1}} \right) s_{k-1} + t \frac{F_k^\top s_{k-1}}{\|s_{k-1}\|^2} \omega_{k-1}, \end{aligned}$$

where  $\omega_{k-1} = y_{k-1} + rs_{k-1}$ ,  $r > 0$  is a constant,  $t \in [0, 1)$  and the relative matrix satisfies the quasi-Newton equation.

In addition, Gao et al. [18] proposed an effective three-term conjugate gradient-based method satisfying with Quasi-Newton equation and the detailed search direction reads

$$\begin{aligned} d_k &= -Q_k F_k \\ &= -\left( \frac{s_{k-1}^\top \bar{y}_{k-1}}{2\|\bar{y}_{k-1}\|^2} I + \frac{s_{k-1} \bar{y}_{k-1}^\top}{2\|\bar{y}_{k-1}\|^2} + \frac{s_{k-1} s_{k-1}^\top}{2s_{k-1}^\top \bar{y}_{k-1}} - \frac{\bar{y}_{k-1} s_{k-1}^\top}{2\|\bar{y}_{k-1}\|^2} \right) F_k \\ &= -\frac{s_{k-1}^\top \bar{y}_{k-1}}{2\|\bar{y}_{k-1}\|^2} F_k - \left( \frac{F_k^\top \bar{y}_{k-1}}{2\|\bar{y}_{k-1}\|^2} + \frac{F_k^\top s_{k-1}}{2s_{k-1}^\top \bar{y}_{k-1}} \right) s_{k-1} + \frac{F_k^\top s_{k-1}}{2\|\bar{y}_{k-1}\|^2} \bar{y}_{k-1}, \end{aligned}$$

where  $\bar{y}_{k-1} = \omega_{k-1}$ , and the relative matrix satisfies the quasi-Newton equation.

Motivated by above articles, we also want to proposed a three-term projection method satisfying with quasi-Newton equation, which implies that the relative matrix  $Q_k$  of search direction (2.3) satisfy the Quasi-Newton equation. Equivalently,

$$\begin{aligned} Q_k y_{k-1} &= \left( I - \beta_k \frac{s_{k-1} y_{k-1}^\top}{y_{k-1}^\top F_k} - \gamma_k \frac{y_{k-1} y_{k-1}^\top}{y_{k-1}^\top F_k} \right) y_{k-1} \\ &= y_{k-1} - \beta_k \frac{\|y_{k-1}\|^2}{y_{k-1}^\top F_k} s_{k-1} - \gamma_k \frac{\|y_{k-1}\|^2}{y_{k-1}^\top F_k} y_{k-1} \\ &= \left( 1 - \gamma_k \frac{\|y_{k-1}\|^2}{y_{k-1}^\top F_k} \right) y_{k-1} + \left( -\beta_k \frac{\|y_{k-1}\|^2}{y_{k-1}^\top F_k} \right) s_{k-1} \\ &= s_k. \end{aligned} \tag{2.4}$$

From (2.4), we can derive the following equations by equaling the coefficients of the corresponding parameters which is motivated by the derivation of Quasi-Newton-type algorithm named Davidon-Fletcher-Powell (DFP, [38]).

$$1 - \gamma_k \frac{\|y_{k-1}\|^2}{y_{k-1}^\top F_k} = 0 \Rightarrow \gamma_k = \frac{y_{k-1}^\top F_k}{\|y_{k-1}\|^2},$$

and

$$-\beta_k \frac{\|y_{k-1}\|^2}{y_{k-1}^\top F_k} = 1 \Rightarrow \beta_k = -\frac{y_{k-1}^\top F_k}{\|y_{k-1}\|^2}.$$

As a result, we obtain the following search direction which satisfies the quasi-Newton equation

$$d_k = -F_k - \frac{y_{k-1}^\top F_k}{\|y_{k-1}\|^2} s_{k-1} + \frac{y_{k-1}^\top F_k}{\|y_{k-1}\|^2} y_{k-1}. \tag{2.5}$$

According to the novel search direction in (2.5),  $y_k$  in (2.5) replaced by  $\bar{y}_k$  and proving the sufficient descent, we propose the following updating formula:

$$d_k = \begin{cases} -F_k, & \text{if } -\frac{F_k^\top \bar{y}_{k-1} F_k^\top s_{k-1}}{\|\bar{y}_{k-1}\|^2} + \frac{(F_k^\top \bar{y}_{k-1})^2}{\|\bar{y}_{k-1}\|^2} > 0, \\ -F_k - \frac{F_k^\top \bar{y}_{k-1}}{\|\bar{y}_{k-1}\|^2} s_{k-1} + \frac{F_k^\top \bar{y}_{k-1}}{\|\bar{y}_{k-1}\|^2} \bar{y}_{k-1}, & \text{otherwise.} \end{cases} \tag{2.6}$$

In addition to search direction, our algorithm also needs to introduce the concept of projection mapping and its related properties.

**Definition 2.1.** Let  $\Omega \subseteq R^n$  is nonempty closed and convex set. Then for any  $x \in R^n$ , its projection onto  $\Omega$  can be defined as

$$P_\Omega[x] = \arg \min \{ \|x - y\| \mid y \in \Omega \}.$$

**Lemma 2.1.** Let  $\Omega \in R^n$  be a nonempty, closed and convex set. Then, the following inequalities hold [16]:

$$(x - P_\Omega(x))^\top (P_\Omega(x) - z) \geq 0, \quad \forall x \in R^n, z \in \Omega, \quad (2.7)$$

$$\|P_\Omega(x) - P_\Omega(y)\| \leq \|x - y\|, \quad \forall x, y \in R^n, \quad (2.8)$$

$$\|P_\Omega(x) - z\|^2 \leq \|x - z\|^2 - \|x - P_\Omega(x)\|^2, \quad \forall x \in R^n, z \in \Omega. \quad (2.9)$$

Based on the above preparations, we described our method as follows.

---

**Algorithm 2.1.**

---

**Step 0.** Initialize  $x_0 \in R^n$ ,  $\rho, \varepsilon, \sigma \in (0, 1)$ ,  $\tau_1 \in (0, 2)$ ,  $r > 0$ ,  $\alpha_0 = 1$ ,  $\xi > 0$ . Set  $k = 0$ .

**Step 1.** While  $\|F_k\| \leq \varepsilon$ , break. Otherwise, turn to Step 2.

**Step 2.** If  $k = 0$ , then  $d_k = -F_k$ . Otherwise,  $d_k$  is computed by (2.6).

**Step 3.** Set  $z_k = x_k + \alpha_k d_k$ , and compute  $\alpha_k = \alpha_k^{BB}(\alpha_k^{BB1} \text{ or } \alpha_k^{BB2})\xi\rho^m$  in which  $m$  is the smallest nonnegative integer such that

$$-F(z_k)^\top d_k \geq \sigma \alpha_k \|F(z_k)\| \|d_k\|^2. \quad (2.10)$$

**Step 4.** If  $z_k \in \Omega$  and  $\|F(z_k)\| \leq \varepsilon$ , then  $x_{k+1} = z_k$ , stop. Otherwise, update  $x_{k+1}$  by

$$x_{k+1} = P_\Omega[x_k - \tau_1 \xi_k F(z_k)],$$

where  $\xi_k = \frac{F(z_k)^\top (x_k - z_k)}{\|F(z_k)\|^2}$ . Then, set  $k := k + 1$ , go to step 1.

---

**Remark 2.1.** We obtain two methods with different step-sizes from Algorithm 2.1: Algorithm 2.1 with  $\alpha_k^{BB} = \alpha_k^{BB1}$  is called NBBP1, and Algorithm 2.1 with  $\alpha_k^{BB} = \alpha_k^{BB2}$  is called NBBP2. The search direction of  $d_k$  satisfies with the Quasi-Newton equation and the sufficient descent property regardless of any line search (see Lemma 3.2 below).

**Remark 2.2.** We clarify that there always exists a nonnegative integer  $m$  satisfying the inequality (2.10). In fact, if it does not hold for any nonnegative integer  $i$ , then

$$-F(x_k + \alpha_k^{BB} \xi \rho^i d_k)^\top d_k < \sigma \alpha_k^{BB} \xi \rho^i \|F(x_k + \alpha_k^{BB} \xi \rho^i d_k)\| \|d_k\|^2.$$

By the continuity and boundedness of  $F$  (see Remark 3.2 below),  $\rho \in (0, 1)$ , and by letting  $i \rightarrow \infty$ , we obtain

$$-F(x_k)^\top d_k \leq 0. \quad (2.11)$$

However, by Lemma 3.2 below, it holds that

$$-F(x_k)^\top d_k \geq \|F(x_k)\|^2 > 0,$$

which contradicts (2.11).

### 3. Convergence analysis

In this section, we analyze the convergence properties (global convergence and convergence rate) of our methods under the following hypotheses:

(B1) The solution set  $\Omega^*$  of (1.1) is nonempty and closed.

(B2) The mapping  $F$  is Lipschitz continuous on  $R^n$ , which means that there exists a constant  $L > 0$  such that

$$\|F(x) - F(y)\| \leq L\|x - y\|, \quad \forall x, y \in R^n.$$

(B3) For any  $x^* \in \Omega^*$ , there exist two positive constants  $b$  (sufficient small) and  $\delta$  such that

$$b \operatorname{dist}(x, \Omega^*) \leq \|F(x)\|, \quad \forall x \in N(x^*, \delta),$$

where  $\operatorname{dist}(x, \Omega^*)$  is the distance from  $x$  to the solution set  $\Omega^*$ , and  $N(x^*, \delta) := \{x \in R^n \mid \|x - x^*\| \leq \delta\}$ .

**Lemma 3.1.** *Suppose that the assumptions (B1)-(B2) hold, the sequence  $\{x_k\}$  and  $\{z_k\}$  are generated by Algorithm 2.1. Then, we have the step-sizes of  $\alpha_k^{BB}$  is bounded.*

**Proof.** From the construction of  $\alpha_k^{BB}$ , namely,  $\alpha_k^{BB1} = \frac{\|s_{k-1}\|^2}{s_{k-1}^\top \bar{y}_{k-1}}$  and  $\alpha_k^{BB2} = \frac{s_{k-1}^\top \bar{y}_{k-1}}{\|\bar{y}_{k-1}\|^2}$ , we can obtain from  $\bar{y}_{k-1}$  and (1.2) that, we have

$$r\|s_{k-1}\|^2 \leq s_{k-1}^\top \bar{y}_{k-1} = s_{k-1}^\top (F_k - F_{k-1} + r s_{k-1}) \leq (L + r)\|s_{k-1}\|^2,$$

and so  $\frac{1}{L+r} \leq \alpha_k^{BB1} \leq \frac{1}{r}$ .

Furthermore, by  $\|\bar{y}_{k-1}\| = \|F_k - F_{k-1} + r s_{k-1}\| \leq \|F_k - F_{k-1}\| + r\|s_{k-1}\|$ , we have

$$r^2\|s_{k-1}\|^2 \leq \|\bar{y}_{k-1}\|^2 \leq (L + r)^2\|s_{k-1}\|^2,$$

and so  $\frac{r}{(L+r)^2} \leq \alpha_k^{BB2} \leq \frac{L+r}{r^2}$ . The conclusion is proved.  $\square$

**Lemma 3.2.** *Suppose that the assumptions (B1)-(B2) hold, the sequence  $\{x_k\}$  and  $\{z_k\}$  are generated by Algorithm 2.1. Then, we have*

$$F_k^\top d_k \leq -\|F_k\|^2. \quad (3.1)$$

**Proof.** If  $-\frac{F_k^\top \bar{y}_{k-1} F_k^\top s_{k-1}}{\|\bar{y}_{k-1}\|^2} + \frac{(F_k^\top \bar{y}_{k-1})^2}{\|\bar{y}_{k-1}\|^2} \leq 0$ , then

$$F_k^\top d_k = F_k^\top \left( -F_k - \frac{F_k^\top \bar{y}_{k-1}}{\|\bar{y}_{k-1}\|^2} s_{k-1} + \frac{F_k^\top \bar{y}_{k-1}}{\|\bar{y}_{k-1}\|^2} \bar{y}_{k-1} \right) \leq -\|F_k\|^2,$$

else

$$F_k^\top d_k = -\|F_k\|^2.$$

The conclusion is proved.  $\square$

**Remark 3.1.** From Lemma 3.2, we further obtain

$$\|F_k\| \|d_k\| \geq -F_k^\top d_k \geq \|F_k\|^2,$$

and simplify it to have

$$\|d_k\| \geq \|F_k\|.$$

**Lemma 3.3.** Suppose that the assumptions (B1)-(B2) hold, the sequence  $\{x_k\}$  and  $\{z_k\}$  are generated by Algorithm 2.1. Then, we have

$$\|d_k\| \leq A\|F_k\|, \quad (3.2)$$

where  $A = \max\{1, 2 + \frac{1}{r}\}$ .

**Proof.** From the construction of  $\bar{y}_{k-1}$  and (1.2), we obtain

$$r\|s_{k-1}\|^2 \leq s_{k-1}^T \bar{y}_{k-1} = s_{k-1}^T (F_k - F_{k-1} + r s_{k-1}) \leq \|s_{k-1}\| \|\bar{y}_{k-1}\|, \quad (3.3)$$

equivalently,

$$\|s_{k-1}\| \leq \frac{\|\bar{y}_{k-1}\|}{r}. \quad (3.4)$$

By (2.6) and (3.4), we discuss the following two scenarios:

If  $-\frac{F_k^\top \bar{y}_{k-1} F_k^\top s_{k-1}}{\|\bar{y}_{k-1}\|^2} + \frac{(F_k^\top \bar{y}_{k-1})^2}{\|\bar{y}_{k-1}\|^2} > 0$ , we have

$$\|d_k\| \leq \|F_k\|.$$

Otherwise,  $-\frac{F_k^\top \bar{y}_{k-1} F_k^\top s_{k-1}}{\|\bar{y}_{k-1}\|^2} + \frac{(F_k^\top \bar{y}_{k-1})^2}{\|\bar{y}_{k-1}\|^2} \leq 0$ , we have

$$\begin{aligned} \|d_k\| &\leq \|F_k\| + \frac{|F_k^\top \bar{y}_{k-1}|}{\|\bar{y}_{k-1}\|^2} \|s_{k-1}\| + \frac{|F_k^\top \bar{y}_{k-1}|}{\|\bar{y}_{k-1}\|^2} \|\bar{y}_{k-1}\| \\ &\leq \|F_k\| + \frac{\|F_k\| \|\bar{y}_{k-1}\|}{\|\bar{y}_{k-1}\|^2} \|s_{k-1}\| + \frac{\|F_k\| \|\bar{y}_{k-1}\|^2}{\|\bar{y}_{k-1}\|^2} \\ &\leq \|F_k\| + \frac{\|F_k\|}{r} + \|F_k\| \\ &= \left(2 + \frac{1}{r}\right) \|F_k\|. \end{aligned}$$

Let  $A = \max\{1, 2 + \frac{1}{r}\}$ , the conclusion is proved.  $\square$

**Lemma 3.4.** Suppose that the assumptions (B1)-(B2) hold. Then, the sequences  $\{x_k\}$  and  $\{z_k\}$  generated by Algorithm 2.1 satisfy

$$\|x_{k+1} - x^*\|^2 \leq \|x_k - x^*\|^2 - \tau_1(2 - \tau_1)\sigma^2 \|x_k - z_k\|^4, \quad \forall x^* \in \Omega^*, \quad (3.5)$$

and hence both of them are bounded. Furthermore, we have

$$\lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0 \quad \text{and} \quad \lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0.$$

**Proof.** By the monotonicity of  $F$ , it follows that

$$(F(z_k) - F(x^*))^\top (x_k - x^*) - (F(z_k) - F(x^*))^\top (x_k - z_k) = (F(z_k) - F(x^*))^\top (z_k - x^*) \geq 0,$$

which, by (2.10) and  $F(x^*) = 0$ , gives

$$(F(z_k) - F(x^*))^\top (x_k - z_k) = -\alpha_k F^\top(z_k) d_k \geq \sigma \alpha_k^2 \|F(z_k)\| \|d_k\|^2 \geq 0. \quad (3.6)$$

For any  $x^* \in \Omega^*$ , from (2.8) and (3.6), we have

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 &= \|P_\Omega[x_k - \tau_1 \xi_k F(z_k)] - P_\Omega(x^*)\|^2 \\
&\leq \|x_k - \tau_1 \xi_k F(z_k) - x^*\|^2 \\
&= \|x_k - x^*\|^2 - 2\tau_1 \xi_k (x_k - x^*)^\top F(z_k) + \tau_1^2 \xi_k^2 \|F(z_k)\|^2 \\
&\leq \|x_k - x^*\|^2 - 2\tau_1 \frac{((x_k - z_k)^\top F(z_k))^2}{\|F(z_k)\|^2} + \tau_1^2 \frac{((x_k - z_k)^\top F(z_k))^2}{\|F(z_k)\|^2} \\
&= \|x_k - x^*\|^2 - \tau_1(2 - \tau_1) \frac{((x_k - z_k)^\top F(z_k))^2}{\|F(z_k)\|^2} \\
&\leq \|x_k - x^*\|^2 - \tau_1(2 - \tau_1) \sigma^2 \alpha_k^4 \|d_k\|^4 \\
&= \|x_k - x^*\|^2 - \tau_1(2 - \tau_1) \sigma^2 \|x_k - z_k\|^4.
\end{aligned}$$

By  $0 < \tau < 2$ , it can be easily obtained that the sequence  $\{\|x_k - x^*\|\}$  is decreasing and hence  $\{x_k\}$  is bounded. By the Cauchy-Schwarz inequality and (2.10), we have

$$\|F(z_k)\| \|x_k - z_k\| \geq F(z_k)^\top (x_k - z_k) \geq \sigma \alpha_k^2 \|F(z_k)\| \|d_k\|^2 = \sigma \|F(z_k)\| \|x_k - z_k\|^2,$$

which suggests

$$\sigma \|x_k - z_k\| \leq 1,$$

and so the sequence  $\{z_k\}$  is also bounded from the boundedness of  $\{x_k\}$ .

Combine the decreasing property of  $\|x - x^*\|$  with (3.5), to obtain

$$\sum_{k=0}^{\infty} \tau_1(2 - \tau_1) \sigma^2 \|x_k - z_k\|^4 \leq \sum_{k=0}^{\infty} (\|x_k - x^*\|^2 - \|x_{k+1} - x^*\|^2) < \infty,$$

which implies that

$$\lim_{k \rightarrow \infty} \|x_k - z_k\| = 0, \Leftrightarrow \lim_{k \rightarrow \infty} \alpha_k \|d_k\| = 0.$$

Now, combine (2.9) and the definition of  $\xi_k$  to have

$$\|x_{k+1} - x_k\| = \|P_\Omega[x_k - \tau_1 \xi_k F(z_k)] - x_k\| \leq \|\tau_1 \xi_k F(z_k)\| \leq \tau_1 \|x_k - z_k\|. \quad (3.7)$$

Consequently,  $\lim_{k \rightarrow \infty} \|x_{k+1} - x_k\| = 0$ . The conclusion is proved.  $\square$

**Remark 3.2.** From Lemma 3.4 and the continuity of  $F$ , the sequence  $\{F(z_k)\}$  and  $\{F(x_k)\}$  are also bounded, i.e.,  $\exists M > 0$ ,  $\|F(x_k)\| \leq M$  and  $\|F(z_k)\| \leq M$ .

**Lemma 3.5.** Suppose that the assumptions (B1)-(B2) hold. Then, we have

$$\alpha_k \geq \min \left\{ \xi, \frac{\rho}{(L + \sigma M) A^2} \right\}.$$

**Proof.** Clearly, if  $\alpha_k = \xi$ , then (2.10) is satisfied.

Otherwise,  $\alpha_k \neq \xi$ , then  $\alpha_k = \rho^{-1} \alpha_k$  does not satisfy the line search (2.10), namely,

$$-F(x_k + \rho^{-1} \alpha_k d_k)^\top d_k < \sigma \rho^{-1} \alpha_k \|F(x_k + \rho^{-1} \alpha_k d_k)\| \|d_k\|^2.$$

Combining it with (B2) and Lemma 3.2, we have

$$\begin{aligned}
\|F_k\|^2 &\leq -F_k^\top d_k \\
&= (F(x_k + \rho^{-1} \alpha_k d_k)^\top d_k - F_k^\top d_k) - F(x_k + \rho^{-1} \alpha_k d_k)^\top d_k \\
&< L \rho^{-1} \alpha_k \|d_k\|^2 + \sigma \rho^{-1} \alpha_k \|d_k\|^2 M \\
&= (L + \sigma M) \rho^{-1} \alpha_k \|d_k\|^2,
\end{aligned} \quad (3.8)$$

which implies

$$\alpha_k > \frac{\rho \|F_k\|^2}{(L + \sigma M) \|d_k\|^2}.$$

Combine it with Lemma 3.3, we have

$$\alpha_k > \frac{\rho}{(L + \sigma M) A^2}. \quad (3.9)$$

The proof is completed.  $\square$

**Theorem 3.1.** *Suppose that assumptions (B1)-(B2) hold. Then, we have*

$$\lim_{k \rightarrow \infty} \|F_k\| = 0. \quad (3.10)$$

**Proof.** If the conclusion (3.10) is not true, there is a constant  $\eta > 0$  such that

$$\|F_k\| \geq \eta, \forall k \geq 0. \quad (3.11)$$

By (3.11) and Remark 3.1, we obtain

$$\|d_k\| \geq \|F_k\| \geq \eta. \quad (3.12)$$

Combining Lemma 3.4 with (3.12), we have

$$\lim_{k \rightarrow \infty} \alpha_k = 0,$$

which contradicts with the conclusion of Lemma 3.5. The proof is completed.  $\square$

**Theorem 3.2.** *Suppose that assumptions (B1)-(B3) hold, and let  $\{x_k\}$  be the sequence generated by Algorithm 2.1. Then the sequence  $\{\text{dist}(x_k, \Omega^*)\}$   $Q$ -linearly convergent to 0, i.e.,*

$$\limsup_{k \rightarrow \infty} \frac{\text{dist}(x_{k+1}, \Omega^*)}{\text{dist}(x_k, \Omega^*)} < 1. \quad (3.13)$$

Therefore, it converges  $R$ -linearly to 0, i.e.,

$$\limsup_{k \rightarrow \infty} [\text{dist}(x_k, \Omega^*)]^{\frac{1}{k}} < 1. \quad (3.14)$$

**Proof.** Let  $\nu_k$  be the projection of  $x_k$  onto  $\Omega^*$ . Because of the proof of Lemma 3.4, we know

$$\begin{aligned} \text{dist}(x_{k+1}, \Omega^*)^2 &\leq \text{dist}(x_k, \Omega^*)^2 - \tau_1(2 - \tau_1) \frac{((x_k - z_k)^\top F(z_k))^2}{\|F(z_k)\|^2} \\ &= \text{dist}(x_k, \Omega^*)^2 - \tau_1(2 - \tau_1) \frac{(-\alpha_k d_k^\top F(z_k))^2}{\|F(z_k)\|^2} \\ &= \text{dist}(x_k, \Omega^*)^2 - \tau_1(2 - \tau_1) \frac{\alpha_k^2 \|d_k\|^2 \|F(z_k)\|^2 e^2}{\|F(z_k)\|^2} \\ &= \text{dist}(x_k, \Omega^*)^2 - \tau_1(2 - \tau_1) \alpha_k^2 \|d_k\|^2 e^2 \\ &\leq \text{dist}(x_k, \Omega^*)^2 - \tau_1(2 - \tau_1) \alpha_k^2 \|F_k\|^2 e^2 \\ &\leq \text{dist}(x_k, \Omega^*)^2 - \tau_1(2 - \tau_1) \alpha_k^2 b^2 e^2 \text{dist}(x_k, \Omega^*)^2 \\ &= (1 - \tau_1(2 - \tau_1) \alpha_k^2 b^2 e^2) \text{dist}(x_k, \Omega^*)^2, \end{aligned} \quad (3.15)$$

where  $e = \cos \angle(F(z_k), x_k - z_k), e^2 \in [0, 1]$ . Furthermore, by (3.9) and (3.15), we have

$$\frac{\text{dist}(x_{k+1}, \Omega^*)}{\text{dist}(x_k, \Omega^*)} \leq (1 - \tau_1(2 - \tau_1)\alpha_k^2 b^2 e^2)^{\frac{1}{2}} < \left(1 - \tau_1(2 - \tau_1) \frac{\rho^2}{(L + \sigma M)^2 A^4} b^2 e^2\right)^{\frac{1}{2}},$$

then,

$$\limsup_{k \rightarrow \infty} \frac{\text{dist}(x_{k+1}, \Omega^*)}{\text{dist}(x_k, \Omega^*)} < \left(1 - \tau_1(2 - \tau_1) \frac{\rho^2}{(L + \sigma M)^2 A^4} b^2 e^2\right)^{\frac{1}{2}} < 1,$$

and so the inequality (3.13) holds.

By above analysis, it is not difficult to obtain that

$$\text{dist}(x_k, \Omega^*) < \left(1 - \tau_1(2 - \tau_1) \frac{\rho^2}{(L + \sigma M)^2 A^4} b^2 e^2\right)^{\frac{k}{2}} \text{dist}(x_0, \Omega^*),$$

thus,

$$\limsup_{k \rightarrow \infty} [\text{dist}(x_k, \Omega^*)]^{\frac{1}{k}} < \left(1 - \tau_1(2 - \tau_1) \frac{\rho^2}{(L + \sigma M)^2 A^4} b^2 e^2\right)^{\frac{1}{2}} [\text{dist}(x_0, \Omega^*)]^{\frac{1}{k}} < 1,$$

and the inequality (3.14) is proved which means that the second conclusion holds.  $\square$

## 4. Numerical experiments

In this section, the methods NBBP1, NBBP2, Liu [25] and Algorithm 2.1 (named NGCG) [18] are used to solve the following test problems by using the fixedly initial points and randomly initial points from (0,1). The parameters for the compared method are taken from corresponding references. We choose the parameters of our methods as  $\rho = 0.85$ ,  $\sigma = 10^{-4}$ ,  $\tau_1 = 1.2$ ,  $r = 0.1$  and  $\xi = 1$  in this paper. The terminate conditions of four methods are the same as that in [25] and the result is showed NA when the CPU time exceeds ten seconds. The coding and operating environment are Matlab R2017a and run on a PC with 2.30 GHZ CPU processor and RAM 8.00 GB for four methods.

**Problem 4.1.** Consider the problem in [14], where  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$  with

$$\begin{aligned} f_1(x) &= 2x_1 - x_2 + \exp(x_1) - 1, \\ f_i(x) &= -x_{i-1} + 2x_i - x_{i+1} + \exp(x_i) - 1, \quad i = 2, 3, \dots, n-1, \\ f_n(x) &= 2x_n - x_{n-1} + \exp(x_n) - 1, \end{aligned}$$

and  $\Omega = R_+^n$ . The following  $\Omega$  is the same as here.

**Problem 4.2.** This problem is from [15]. The mapping  $F$  is taken as  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_i(x) = \exp(x_i) - 1, \quad i = 1, 2, 3, \dots, n.$$

**Problem 4.3.** The problem comes from [3]. The mapping  $F$  is taken as  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_i(x) = (x_i)^2 + x_i - 2, \quad i = 1, 2, \dots, n.$$

**Problem 4.4.** The problem originates from [37]. The mapping  $F$  is taken as  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_i(x) = \left(\frac{i}{n}\right) \exp(x_i) - 1 \quad i = 1, 2, \dots, n.$$

**Problem 4.5.** The problem is taken from [37]. The mapping  $F$  is taken as  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$f_i(x) = 2x_i - \sin(x_i) \quad i = 1, 2, \dots, n.$$

**Problem 4.6.** The problem is extracted from [37]. The mapping  $F$  is taken as  $F(x) = (f_1(x), f_2(x), \dots, f_n(x))^T$ , where

$$\begin{aligned} f_1(x) &= \exp(x_1) - 1, \\ f_i(x) &= \exp(x_i) + x_i - 1, \quad i = 2, \dots, n. \end{aligned}$$

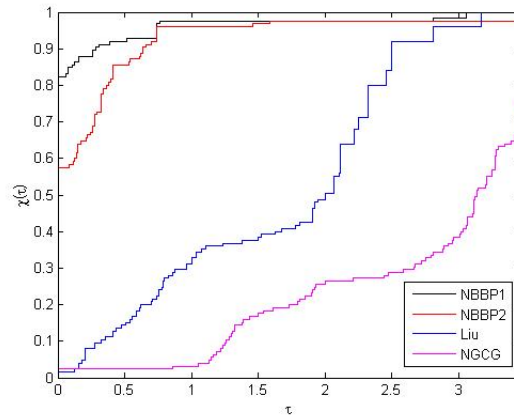
**Table 1.** Experimental results of Problem 4.1 for four algorithms.

|    |        | NBBP1              | NBBP2               | Liu                 | NGCG                  | Ablation    |
|----|--------|--------------------|---------------------|---------------------|-----------------------|-------------|
| IP | DIM    | NI/NF/CPU/FV       | NI/NF/CPU/FV        | NI/NF/CPU/FV        | NI/NF/CPU/FV          | NI/NF/CPU   |
| X1 | 5000   | 18/21/0.01/9.31279 | 26/31/0.03/4.63626  | 32/94/0.04/7.42943  | 119/360/0.18/7.98252  | 48/477/0.06 |
|    | 10000  | 19/22/0.03/4.56623 | 23/28/0.05/4.57439  | 33/97/0.07/6.57900  | 140/628/0.28/9.55819  | 61/595/0.15 |
|    | 50000  | 19/22/0.08/7.15111 | 24/32/0.12/9.01623  | 40/114/0.26/9.08689 | 159/745/1.28/8.60341  | 48/486/0.50 |
|    | 80000  | 19/22/0.15/7.77992 | 22/29/0.20/8.76323  | 41/117/0.39/6.97489 | 158/726/1.85/7.43442  | 47/466/0.68 |
|    | 100000 | 19/22/0.18/8.01945 | 23/30/0.21/4.68480  | 39/113/0.46/7.31736 | 154/689/2.62/9.22239  | 47/467/1.18 |
| X2 | 5000   | 19/21/0.01/5.19822 | 23/27/0.02/6.87146  | 31/91/0.03/8.35219  | 114/409/0.12/8.66803  | 44/434/0.06 |
|    | 10000  | 19/21/0.02/6.24236 | 25/30/0.03/4.96690  | 32/94/0.07/8.99039  | 121/386/0.21/9.24258  | 44/434/0.11 |
|    | 50000  | 19/21/0.10/9.18597 | 21/24/0.12/5.07736  | 39/111/0.31/8.07298 | 132/545/0.97/9.33151  | 69/660/0.62 |
|    | 80000  | 19/21/0.15/9.76869 | 21/24/0.17/5.19059  | 37/107/0.38/7.68913 | 144/531/1.51/7.16998  | 45/445/0.64 |
|    | 100000 | 19/21/0.22/9.98122 | 21/24/0.23/5.22992  | 37/107/0.48/7.27034 | 147/538/1.84/9.40226  | 45/445/0.78 |
| X3 | 5000   | 20/36/0.02/6.19587 | 55/62/0.04/8.59363  | 25/136/0.05/9.53205 | 177/934/0.22/9.27197  | 60/657/0.08 |
|    | 10000  | 20/42/0.04/5.95404 | 60/72/0.10/8.88160  | 23/103/0.06/8.79467 | 174/939/0.40/7.55590  | 55/611/0.14 |
|    | 50000  | 25/60/0.26/4.58684 | 75/120/0.61/8.05478 | 4/119/0.15/0        | 3/50/0.08/0           | 51/586/0.56 |
|    | 80000  | 25/68/0.45/8.58154 | 77/133/1.01/7.63430 | 4/119/0.27/0        | 3/51/0.13/0           | 51/574/0.80 |
|    | 100000 | 28/97/0.63/4.21950 | 68/139/1.31/7.97262 | 4/119/0.30/0        | 4/78/0.19/0           | 49/553/0.99 |
| X4 | 5000   | 11/12/0.02/9.25199 | 13/14/0.03/6.18790  | 16/46/0.04/7.57054  | 61/149/0.06/9.74203   | 55/496/0.09 |
|    | 10000  | 10/11/0.03/8.26390 | 12/13/0.04/5.89311  | 15/43/0.05/7.31198  | 64/145/0.09/9.72406   | 52/466/0.16 |
|    | 50000  | 8/9/0.06/8.01246   | 10/11/0.07/5.22219  | 13/35/0.10/8.56986  | 37/36/0.16/8.76948    | 47/416/0.37 |
|    | 80000  | 8/9/0.08/5.03683   | 9/10/0.09/6.94918   | 12/32/0.15/8.36380  | 54/47/0.33/9.46852    | 46/406/0.55 |
|    | 100000 | 7/8/0.09/9.19890   | 9/10/0.10/5.56771   | 12/32/0.17/6.94226  | 38/35/0.31/9.25774    | 45/396/0.67 |
| X5 | 5000   | 18/31/0.02/8.95854 | 22/33/0.03/9.08024  | 36/106/0.04/8.08689 | 126/491/0.13/8.41998  | 43/432/0.06 |
|    | 10000  | 18/26/0.02/4.29426 | 21/29/0.04/5.23545  | 33/97/0.06/8.86457  | 177/795/0.36/8.86186  | 46/469/0.13 |
|    | 50000  | 19/27/0.11/5.40319 | 30/50/0.17/6.16445  | 23/67/0.18/6.19675  | 225/1240/2.06/8.61570 | 45/453/0.43 |
|    | 80000  | 19/27/0.18/8.73978 | 23/38/0.22/5.72233  | 23/67/0.24/5.45643  | 210/1537/3.35/7.25575 | 46/463/0.76 |
|    | 100000 | 20/28/0.25/8.66920 | 22/37/0.26/7.97284  | 23/67/0.28/8.02639  | 161/964/3.07/9.74702  | 47/473/0.83 |

Comparative results in terms of NI/NF/CPU/FV are shown in Tables 1-6, where “NI” denotes the number of iterations, “NF” stands for the number of function value iterations, “CPU” stands for the CPU time, “FV( $10^{-6}$ )” stands for the terminate function value of each

**Table 2.** Experimental results of Problem 4.2 for four algorithms.

| IP | DIM    | NBBP1                 | NBBP2                 | Liu                | NGCG                | Ablation        |
|----|--------|-----------------------|-----------------------|--------------------|---------------------|-----------------|
|    |        | NI/NF/CPU/FV          | NI/NF/CPU/FV          | NI/NF/CPU/FV       | NI/NF/CPU/FV        | NI/NF/CPU       |
| X1 | 5000   | 5/16/0.02/0           | 3/6/0.01/0            | 17/22/0.03/3.64711 | 33/2/0.03/7.91061   | <b>2/7/0.01</b> |
|    | 10000  | 5/16/0.03/0           | 3/6/0.02/0            | 17/22/0.04/3.64711 | 33/2/0.04/7.91061   | <b>2/7/0.02</b> |
|    | 50000  | 5/16/0.05/0           | 3/6/0.03/0            | 17/22/0.07/3.64711 | 33/2/0.07/7.91061   | <b>2/7/0.03</b> |
|    | 80000  | 5/16/0.07/0           | 3/6/0.04/0            | 17/22/0.09/3.64711 | 33/2/0.10/7.91061   | <b>2/7/0.04</b> |
|    | 100000 | 5/16/0.09/0           | 3/6/0.05/0            | 17/22/0.11/3.64711 | 33/2/0.14/7.91061   | <b>2/7/0.04</b> |
| X2 | 5000   | 3/5/0.01/0            | 3/5/0.01/0            | 17/23/0.02/6.95808 | 33/2/0.03/9.62609   | 3/11/0.01       |
|    | 10000  | 3/5/0.01/0            | 3/5/0.02/0            | 17/23/0.04/6.96587 | 33/2/0.04/9.62585   | 3/11/0.02       |
|    | 50000  | 3/5/0.02/0            | 3/5/0.03/0            | 17/23/0.07/6.97211 | 33/2/0.09/9.62565   | 3/11/0.03       |
|    | 80000  | 3/5/0.03/0            | 3/5/0.04/0            | 17/23/0.10/6.97269 | 33/2/0.13/9.62563   | 3/11/0.04       |
|    | 100000 | 3/5/0.04/0            | 3/5/0.05/0            | 17/23/0.12/6.97289 | 33/2/0.17/9.62563   | 3/11/0.04       |
| X3 | 5000   | 2/3/0.01/2.22045e-10  | 2/3/0.01/2.22045e-10  | 11/15/0.02/3.48297 | 29/2/0.02/8.90589   | 2/4/0.01        |
|    | 10000  | 2/3/0.01/2.22045e-10  | 2/3/0.02/2.22045e-10  | 11/15/0.03/3.48297 | 29/2/0.03/8.90589   | 2/4/0.01        |
|    | 50000  | 2/3/0.02/2.22045e-10  | 2/3/0.03/2.22045e-10  | 11/15/0.05/3.48297 | 29/2/0.07/8.90589   | 2/4/0.02        |
|    | 80000  | 2/3/0.03/2.22045e-10  | 2/3/0.04/2.22045e-10  | 11/15/0.08/3.48297 | 29/2/0.10/8.90589   | 2/4/0.03        |
|    | 100000 | 2/3/0.04/2.22045e-10  | 2/3/0.05/2.22045e-10  | 11/15/0.09/3.48297 | 29/2/0.12/8.90589   | 2/4/0.03        |
| X4 | 5000   | 3/13/0.01/4.96507e-10 | 3/13/0.01/4.96507e-10 | 13/18/0.02/5.20222 | 26/2/0.02/9.41767   | 4/15/0.01       |
|    | 10000  | 3/13/0.02/4.96507e-10 | 3/13/0.03/4.96507e-10 | 13/18/0.04/5.20222 | 26/2/0.04/9.41767   | 4/15/0.02       |
|    | 50000  | 3/13/0.04/4.96507e-10 | 3/13/0.05/4.96507e-10 | 13/18/0.06/5.20222 | 26/2/0.07/9.41767   | 4/15/0.03       |
|    | 80000  | 3/13/0.05/4.96507e-10 | 3/13/0.06/4.96507e-10 | 13/18/0.07/5.20222 | 26/2/0.10/9.41767   | 4/15/0.05       |
|    | 100000 | 3/13/0.06/4.96507e-10 | 3/13/0.07/4.96507e-10 | 13/18/0.09/5.20222 | 26/2/0.11/9.41767   | 4/15/0.06       |
| X5 | 5000   | 11/32/0.02/0          | 12/38/0.03/0          | 19/24/0.04/7.36891 | 42/80/0.05/8.75568  | <b>1/3/0.01</b> |
|    | 10000  | 12/40/0.03/0          | 13/42/0.04/0          | 20/26/0.05/2.15277 | 42/109/0.06/9.62486 | <b>1/3/0.01</b> |
|    | 50000  | 18/89/0.19/0          | 18/76/0.18/0          | 20/26/0.08/3.93016 | 45/110/0.20/8.07098 | <b>1/3/0.02</b> |
|    | 80000  | 19/93/0.35/0          | 18/76/0.30/0          | 20/26/0.12/4.86724 | 41/154/0.39/9.05571 | <b>1/3/0.02</b> |
|    | 100000 | 24/144/0.49/0         | 22/115/0.47/0         | 20/26/0.16/5.40317 | 47/143/0.51/7.42211 | <b>1/3/0.02</b> |

**Figure 1.** Performance profiles of four algorithms (NBBP1, NBBP2, Liu and NGCG) in terms of iterative number.

method, “Dim” represents the problem dimensions, and “IP” represents the initial points. Table 1 shows the reported results about Problem 4.1, where the initial points are taken as follows:  $x_1 = (\frac{1}{7}, \frac{1}{7}, \dots, \frac{1}{7})^\top$ ,  $x_2 = (0.1, 0.1, \dots, 0.1)^\top$ ,  $x_3 = (1, 1, \dots, 1)^\top$ ,  $x_4 = (\frac{1}{n}, \frac{1}{n}, \dots, \frac{1}{n})^\top$ ,  $x_5 = (\frac{1}{3}, \frac{1}{3}, \dots, \frac{1}{3})^\top$ . Table 2 lists the numerical results about Problem 4.2, where the initial points

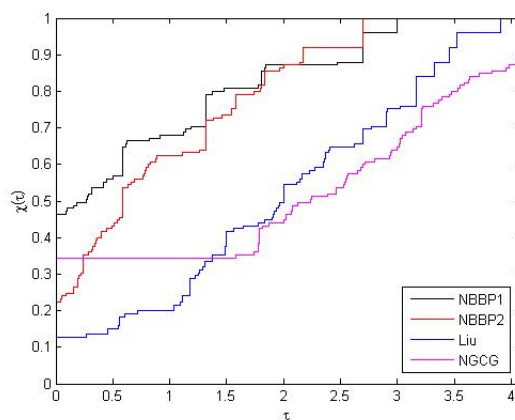
**Table 3.** Experimental results of Problem 4.3 for four algorithms.

|        | NBBP1                 | NBBP2                 | Liu                 | NGCG         |
|--------|-----------------------|-----------------------|---------------------|--------------|
| DIM    | NI/NF/CPU/FV          | NI/NF/CPU/FV          | NI/NF/CPU/FV        | NI/NF/CPU/FV |
| 5000   | 8/17/0.01/6.21492     | 8/14/0.01/6.54042     | 78/158/0.03/9.09627 | NA           |
|        | 10/20/0.01/4.64830    | 8/16/0.01/6.86809     | 78/158/0.04/8.79852 | NA           |
|        | 10/17/0.01/2.06199    | 8/15/0.01/3.88685     | 78/158/0.06/8.93414 | NA           |
|        | 8/16/0.01/6.55624     | 10/20/0.01/2.03916    | 78/158/0.05/8.99188 | NA           |
|        | 10/20/0.01/2.50972    | 10/19/0.01/3.88973    | 78/158/0.03/8.69663 | NA           |
| 10000  | 10/17/0.01/6.94980e-1 | 10/19/0.01/5.77155    | 80/162/0.06/8.88374 | NA           |
|        | 8/17/0.01/9.66004     | 9/18/0.01/4.64568e-1  | 80/162/0.06/8.74839 | NA           |
|        | 10/20/0.01/6.27483e-1 | 8/15/0.01/6.91873     | 80/162/0.06/8.82644 | NA           |
|        | 8/16/0.01/7.20756     | 8/16/0.01/5.71288     | 80/162/0.06/8.87768 | NA           |
|        | 10/20/0.01/3.53755    | 10/17/0.01/2.14764    | 80/162/0.06/8.76051 | NA           |
| 50000  | 10/19/0.03/2.41851    | 10/17/0.03/3.96476    | 84/170/0.21/9.43225 | NA           |
|        | 10/20/0.03/1.13645    | 9/20/0.03/2.92764     | 84/170/0.18/9.37335 | NA           |
|        | 10/20/0.03/7.48513    | 9/20/0.03/3.01407     | 84/170/0.17/9.41641 | NA           |
|        | 10/20/0.03/1.21450    | 10/21/0.03/2.02319    | 84/170/0.18/9.48097 | NA           |
|        | 10/19/0.03/2.42916    | 10/19/0.03/7.43870e-1 | 84/170/0.18/9.39495 | NA           |
| 80000  | 10/20/0.06/9.55293    | 9/20/0.06/6.66906     | 85/172/0.32/9.98720 | NA           |
|        | 10/20/0.05/1.28926    | 10/19/0.06/2.03168    | 86/174/0.29/8.39177 | NA           |
|        | 10/20/0.06/9.91531    | 9/20/0.06/8.66070     | 85/172/0.28/9.92904 | NA           |
|        | 10/20/0.06/1.02263    | 10/17/0.06/7.10540    | 86/174/0.26/8.35469 | NA           |
|        | 10/20/0.05/9.89054    | 10/21/0.05/3.49063    | 85/172/0.27/9.87883 | NA           |
| 100000 | 10/20/0.08/1.10937    | 9/20/0.08/5.26203     | 86/174/0.48/9.30713 | NA           |
|        | 10/20/0.07/1.19773    | 10/21/0.07/2.69449    | 86/174/0.37/9.32851 | NA           |
|        | 10/20/0.07/1.61981    | 10/19/0.06/1.84909    | 86/174/0.51/9.31622 | NA           |
|        | 10/20/0.07/1.11744    | 10/17/0.07/6.85561    | 86/174/0.41/9.26915 | NA           |
|        | 10/20/0.07/1.48836    | 10/21/0.07/2.99990e-1 | 86/174/0.35/9.28354 | NA           |

are taken as follows:  $x_1 = (1, \frac{1}{2^2}, \dots, \frac{1}{n^2})^\top$ ,  $x_2 = (1, \frac{1}{2}, \dots, \frac{1}{n})^\top$ ,  $x_3 = (\frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^n})^\top$ ,  $x_4 = (\frac{1}{3}, \frac{1}{3^2}, \dots, \frac{1}{3^n})^\top$ ,  $x_5 = (1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0)^\top$ . Table 3 shows the numerical results about Problem 4.3, and the initial points are randomly generated from  $(0, 1)$ . Table 4 shows the numerical results about Problem 4.4, and we choose the initial points as  $x_1 = (\frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^n})^\top$ ,  $x_2 = (0.2, 0.2, \dots, 0.2)^\top$ ,  $x_3 = (\frac{1}{7}, \frac{1}{7}, \dots, \frac{1}{7})^\top$ ,  $x_4 = (\frac{1}{3}, \frac{1}{3^2}, \dots, \frac{1}{3^n})^\top$ ,  $x_5 = (0.4, 0.4, \dots, 0.4)^\top$ . Table 5 shows the numerical results about Problem 4.5, and we choose the initial points as  $x_1 = (\frac{1}{n}, \frac{2}{n}, \dots, 1)^\top$ ,  $x_2 = (1, \frac{1}{2^2}, \dots, \frac{1}{n^2})^\top$ ,  $x_3 = (1, \frac{1}{2}, \dots, \frac{1}{n})^\top$ ,  $x_4 = (1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0)^\top$ ,  $x_5 = (\frac{1}{3}, \frac{1}{3^2}, \dots, \frac{1}{3^n})^\top$ . Table 6 reports the numerical results about Problem 4.6, and we choose the initial points as  $x_1 = (\frac{1}{n}, \frac{2}{n}, \dots, 1)^\top$ ,  $x_2 = (\frac{1}{2}, \frac{1}{2^2}, \dots, \frac{1}{2^n})^\top$ ,  $x_3 = (1, \frac{1}{2}, \dots, \frac{1}{n})^\top$ ,  $x_4 = (1 - \frac{1}{n}, 1 - \frac{2}{n}, \dots, 0)^\top$ ,  $x_5 = (\frac{1}{3}, \frac{1}{3^2}, \dots, \frac{1}{3^n})^\top$ . Generally speaking, the proposed methods NBBP1 is more competitive and effective than other methods in terms of NI/NF/CPU/FV from Tables 1-6. From Table 1, we notice that our method NBBP1 is insensitive to the increase of dimension, while our method NBBP2 is sensitive to the initial points. From Table 4, we observe that the two methods are sensitive to the initial points. From Tables 2-3 and 5-6, there are no significant differences between NBBP1 and NBBP2. Due to the random initial point of the data in Table

**Table 4.** Experimental results of Problem 4.4 for four algorithms.

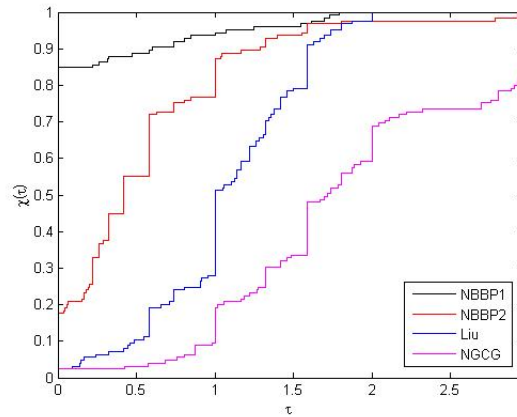
| IP | DIM    | NBBP1               | NBBP2                   | Liu                  | NGCG                | Ablation      |
|----|--------|---------------------|-------------------------|----------------------|---------------------|---------------|
|    |        | NI/NF/CPU/FV        | NI/NF/CPU/FV            | NI/NF/CPU/FV         | NI/NF/CPU/FV        | NI/NF/CPU     |
| X1 | 5000   | 19/32/0.02/7.13692  | 18/31/0.03/6.14878      | 23/30/0.04/5.85151   | 47/122/0.08/8.31957 | 25/166/0.04   |
|    | 10000  | 20/33/0.04/3.61428  | 18/31/0.05/8.36218      | 25/27/0.06/4.94250   | 51/170/0.10/7.77636 | 41/418/0.16   |
|    | 50000  | 24/41/0.20/4.49707  | 23/36/0.23/4.98480      | 65/222/0.55/8.70957  | 60/279/0.50/7.88447 | 55/614/0.86   |
|    | 80000  | 24/36/0.32/7.01152  | 24/37/0.33/4.49104      | 69/246/0.78/9.53334  | 56/298/0.77/8.85964 | 76/1148/2.32  |
|    | 100000 | 25/39/0.40/7.05120  | 23/35/0.41/3.67454      | 71/263/1.04/9.88020  | 56/267/0.91/8.12017 | 71/971/2.47   |
| X2 | 5000   | 15/25/0.01/1.04243  | 19/44/0.02/2.88184      | 23/30/0.03/3.49405   | 57/210/0.07/8.51545 | 24/151/0.04   |
|    | 10000  | 16/25/0.04/2.45125  | 16/31/0.05/4.78902      | 29/41/0.06/6.08480   | 60/202/0.11/9.03987 | 34/311/0.12   |
|    | 50000  | 34/139/0.30/5.87481 | 20/50/0.25/4.53056      | 47/148/0.34/5.17946  | 61/292/0.52/7.88777 | 74/1125/1.55  |
|    | 80000  | 30/122/0.57/2.07207 | 21/56/0.41/9.65711      | 72/272/0.85/7.73434  | 61/330/0.81/8.51759 | 74/1049/2.14  |
|    | 100000 | 27/98/0.81/9.57882  | 25/79/0.65/2.17478      | 65/247/0.95/5.72630  | 62/372/1.10/7.14399 | 74/1181/3.06  |
| X3 | 5000   | 20/53/0.02/1.57552  | 26/104/0.05/3.75991     | 23/30/0.03/3.71234   | 54/174/0.06/9.06052 | 29/135/0.05   |
|    | 10000  | 22/66/0.04/3.06480  | 32/120/0.14/3.70946     | 24/30/0.05/9.78023   | 55/209/0.12/8.59356 | 30/177/0.10   |
|    | 50000  | 32/98/0.30/3.83902  | 34/170/0.62/2.09889     | 49/143/0.32/7.05311  | 58/294/0.55/8.34642 | 40/294/0.48   |
|    | 80000  | 27/90/0.59/3.76125  | 42/248/1.44/7.19617     | 104/465/1.46/6.41246 | 59/308/0.88/7.94951 | 60/716/1.53   |
|    | 100000 | 26/101/0.78/1.26572 | 40/218/1.60/6.37696e-10 | 91/379/1.68/6.84301  | 64/370/1.25/9.69125 | 61/717/1.91   |
| X4 | 5000   | 20/75/0.02/3.57848  | 20/55/0.03/1.54500      | 23/30/0.03/3.73057   | 48/122/0.05/8.91891 | 36/269/0.06   |
|    | 10000  | 28/129/0.10/1.72536 | 23/61/0.06/3.00312      | 26/36/0.09/6.66348   | 51/171/0.12/8.41490 | 42/416/0.16   |
|    | 50000  | 29/126/0.28/3.39013 | 26/67/0.24/2.62319      | 52/143/0.33/9.81833  | 62/283/0.58/7.99468 | 71/868/1.21   |
|    | 80000  | 30/140/0.84/6.70450 | 25/54/0.40/6.93404      | 108/441/1.27/5.66337 | 56/298/0.81/8.56468 | 80/1055/2.13  |
|    | 100000 | 34/173/0.97/9.70749 | 26/76/0.64/2.87580      | 83/356/1.24/7.67959  | 54/267/0.86/7.24582 | 77/1029/2.61  |
| X5 | 5000   | 15/31/0.02/8.80220  | 18/45/0.03/1.13688      | 22/29/0.04/9.39474   | 56/236/0.07/7.21012 | 46/520/0.09   |
|    | 10000  | 21/46/0.05/3.48871  | 17/38/0.04/1.08494      | 23/30/0.06/4.40376   | 59/245/0.14/9.72310 | 70/1036/0.35  |
|    | 50000  | 19/41/0.19/4.75273  | 29/113/0.50/1.87411     | 27/29/0.21/5.47220   | 63/340/0.57/8.46435 | 116/2252/2.79 |
|    | 80000  | 16/34/0.22/1.79724  | 26/98/0.66/5.85904      | 27/28/0.26/9.34062   | 64/346/0.88/9.34587 | 140/2996/5.88 |
|    | 100000 | 18/45/0.30/2.22854  | 28/110/0.90/4.32746     | 31/33/0.33/4.68878   | 65/368/1.11/8.82989 | 156/3388/8.10 |

**Figure 2.** Performance profiles of four algorithms (NBBP1, NBBP2, Liu and NGCG) in terms of the iterative number of function value.

3, its initial point is uncertain. Conducting ablation experiments cannot guarantee consistency with the initial point of the new algorithms. Such experimental results are meaningless. We did not conduct the ablation experiment for the third test function. And so we only add Ablation experiments for Tables 1-2 and 4-6. We name the ablation experiment as Ablation in Tables 1-2

**Table 5.** Experimental results of Problem 4.5 for four algorithms.

| IP | DIM    | NBBP1        | NBBP2        | Liu                | NGCG                | Ablation        |
|----|--------|--------------|--------------|--------------------|---------------------|-----------------|
|    |        | NI/NF/CPU/FV | NI/NF/CPU/FV | NI/NF/CPU/FV       | NI/NF/CPU/FV        | NI/NF/CPU       |
| X1 | 5000   | 4/10/0.01/0  | 4/10/0.01/0  | 19/26/0.02/3.91676 | 38/65/0.04/9.84627  | 5/10/0.02       |
|    | 10000  | 4/10/0.02/0  | 4/10/0.03/0  | 19/26/0.04/5.53821 | 38/90/0.07/8.92078  | 5/10/0.02       |
|    | 50000  | 4/10/0.03/0  | 4/10/0.04/0  | 20/28/0.08/4.25273 | 43/110/0.21/9.92452 | 5/10/0.03       |
|    | 80000  | 4/10/0.05/0  | 4/10/0.06/0  | 20/28/0.11/5.37925 | 39/155/0.37/7.88977 | 5/10/0.05       |
|    | 100000 | 4/10/0.06/0  | 4/10/0.07/0  | 20/28/0.14/6.01416 | 37/116/0.39/9.68974 | 5/10/0.06       |
| X2 | 5000   | 3/13/0.01/0  | 3/13/0.01/0  | 13/18/0.02/4.22394 | 29/2/0.02/9.64591   | <b>2/3/0.01</b> |
|    | 10000  | 3/13/0.02/0  | 3/13/0.03/0  | 13/18/0.04/4.22394 | 29/2/0.04/9.64591   | <b>2/3/0.02</b> |
|    | 50000  | 3/13/0.03/0  | 3/13/0.04/0  | 13/18/0.05/4.22394 | 29/2/0.06/9.64591   | <b>2/3/0.03</b> |
|    | 80000  | 3/13/0.04/0  | 3/13/0.05/0  | 13/18/0.07/4.22394 | 29/2/0.09/9.64591   | <b>2/3/0.03</b> |
|    | 100000 | 3/13/0.06/0  | 3/13/0.07/0  | 13/18/0.09/4.22394 | 29/2/0.11/9.64591   | <b>2/3/0.03</b> |
| X3 | 5000   | 2/5/0.01/0   | 2/5/0.01/0   | 14/20/0.02/6.08034 | 31/2/0.02/7.29092   | 2/3/0.01        |
|    | 10000  | 2/5/0.01/0   | 2/5/0.02/0   | 14/20/0.03/6.08034 | 31/2/0.03/7.29114   | 2/3/0.01        |
|    | 50000  | 2/5/0.02/0   | 2/5/0.03/0   | 14/20/0.06/6.08034 | 31/2/0.08/7.29133   | 2/3/0.02        |
|    | 80000  | 2/5/0.03/0   | 2/5/0.04/0   | 14/20/0.08/6.08034 | 31/2/0.11/7.29134   | 2/3/0.03        |
|    | 100000 | 2/5/0.03/0   | 2/5/0.04/0   | 14/20/0.09/6.08034 | 31/2/0.14/7.29135   | 2/3/0.03        |
| X4 | 5000   | 4/19/0.01/0  | 4/19/0.01/0  | 19/26/0.02/3.91413 | 37/65/0.03/8.05620  | 5/10/0.02       |
|    | 10000  | 4/19/0.02/0  | 4/19/0.03/0  | 19/26/0.04/5.53635 | 42/65/0.06/7.22824  | 5/10/0.02       |
|    | 50000  | 4/19/0.03/0  | 4/19/0.04/0  | 20/28/0.08/4.25241 | 43/110/0.23/8.96972 | 5/10/0.03       |
|    | 80000  | 4/19/0.05/0  | 4/19/0.06/0  | 20/28/0.12/5.37899 | 37/152/0.38/9.11463 | 5/10/0.05       |
|    | 100000 | 4/19/0.06/0  | 4/19/0.07/0  | 20/28/0.15/6.01393 | 39/125/0.46/9.17488 | 5/10/0.06       |
| X5 | 5000   | 1/3/0.01/0   | 1/3/0.01/0   | 9/13/0.02/7.94263  | 23/2/0.02/9.21872   | 1/2/0.01        |
|    | 10000  | 1/3/0.01/0   | 1/3/0.02/0   | 9/13/0.03/7.94263  | 23/2/0.03/9.21872   | 1/2/0.01        |
|    | 50000  | 1/3/0.02/0   | 1/3/0.03/0   | 9/13/0.04/7.94263  | 23/2/0.06/9.21872   | 1/2/0.02        |
|    | 80000  | 1/3/0.03/0   | 1/3/0.04/0   | 9/13/0.06/7.94263  | 23/2/0.08/9.21872   | 1/2/0.02        |
|    | 100000 | 1/3/0.03/0   | 1/3/0.04/0   | 9/13/0.07/7.94263  | 23/2/0.09/9.21872   | 1/2/0.03        |

**Figure 3.** Performance profiles of four algorithms (NBBP1, NBBP2, Liu and NGCG) in terms of the CPU time.

and 4-6. The results are listed as NI/NF/CPU. By comparison Ours and Ablation, we mark the bad results bold. Briefly, the accelerate structure is valid.

According to Tables 1-6 (without Table 3), the performance profile proposed by Dolan and

**Table 6.** Experimental results of Problem 4.6 for four algorithms.

| IP | DIM    | NBBP1                | NBBP2             | Liu                | NGCG                | Ablation          |
|----|--------|----------------------|-------------------|--------------------|---------------------|-------------------|
|    |        | NI/NF/CPU/FV         | NI/NF/CPU/FV      | NI/NF/CPU/FV       | NI/NF/CPU/FV        | NI/NF/CPU         |
| X1 | 5000   | 5/19/0.01/0          | 5/19/0.02/0       | 19/39/0.03/9.58542 | 37/146/0.05/6.31840 | 11/94/0.02        |
|    | 10000  | 5/19/0.02/0          | 5/19/0.03/0       | 20/41/0.04/6.41353 | 42/176/0.08/7.37796 | 15/152/0.04       |
|    | 50000  | 5/19/0.03/0          | 5/19/0.05/0       | 21/43/0.11/6.18195 | 42/199/0.27/6.79131 | 40/732/0.51       |
|    | 80000  | 5/19/0.04/0          | 5/19/0.06/0       | 21/43/0.16/7.81955 | 41/167/0.43/9.68020 | 50/1053/1.18      |
|    | 100000 | 5/19/0.06/0          | 5/19/0.07/0       | 21/43/0.20/8.74250 | 46/271/0.64/6.37846 | 71/1637/2.09      |
| X2 | 5000   | 3/7/0.01/9.93014     | 5/25/0.02/2.17558 | 14/28/0.03/7.78275 | 94/65/0.04/7.73270  | 4/20/0.01         |
|    | 10000  | 3/7/0.02/9.93014     | 5/25/0.03/2.17558 | 14/28/0.04/7.78275 | 94/65/0.06/7.73270  | 4/20/0.02         |
|    | 50000  | 3/7/0.03/9.93014     | 5/25/0.05/2.17558 | 14/28/0.06/7.78275 | 94/65/0.21/7.73270  | 4/20/0.04         |
|    | 80000  | 3/7/0.04/9.93014     | 5/25/0.07/2.17558 | 14/28/0.09/7.78275 | 94/65/0.33/7.73270  | 4/20/0.05         |
|    | 100000 | 3/7/0.05/9.93014     | 5/25/0.09/2.17558 | 14/28/0.11/7.78275 | 94/65/0.46/7.73270  | 4/20/0.05         |
| X3 | 5000   | 4/11/0.01/0          | 5/13/0.02/0       | 15/31/0.03/7.55707 | 46/38/0.04/5.68668  | <b>3</b> /17/0.01 |
|    | 10000  | 4/11/0.02/0          | 5/13/0.03/0       | 15/31/0.04/7.55707 | 46/38/0.05/5.65471  | <b>3</b> /17/0.02 |
|    | 50000  | 4/11/0.04/0          | 5/13/0.05/0       | 15/31/0.08/7.55707 | 46/38/0.12/5.67823  | <b>3</b> /17/0.04 |
|    | 80000  | 4/11/0.06/0          | 5/13/0.07/0       | 15/31/0.12/7.55707 | 45/38/0.20/9.86938  | <b>3</b> /17/0.06 |
|    | 100000 | 4/11/0.07/0          | 5/13/0.08/0       | 15/31/0.14/7.55707 | 46/38/0.23/5.47407  | <b>3</b> /17/0.07 |
| X4 | 5000   | 5/19/0.01/0          | 5/19/0.02/0       | 19/39/0.03/9.58498 | 44/122/0.04/6.91136 | 11/75/0.02        |
|    | 10000  | 5/19/0.02/0          | 5/19/0.03/0       | 20/41/0.04/6.41805 | 44/195/0.08/7.09913 | 14/144/0.04       |
|    | 50000  | 5/19/0.04/0          | 5/19/0.05/0       | 21/43/0.11/6.18192 | 43/178/0.27/7.12946 | 38/737/0.49       |
|    | 80000  | 5/19/0.06/0          | 5/19/0.07/0       | 21/43/0.17/7.81953 | 51/189/0.54/7.11552 | 62/1410/1.54      |
|    | 100000 | 5/19/0.08/0          | 5/19/0.09/0       | 21/43/0.20/8.74248 | 39/229/0.60/8.24633 | 43/917/1.18       |
| X5 | 5000   | 3/7/0.01/8.88178e-10 | 4/9/0.01/0        | 15/30/0.02/9.20077 | 51/2/0.03/8.35344   | 6/36/0.02         |
|    | 10000  | 3/7/0.01/8.88178e-10 | 4/9/0.02/0        | 15/30/0.03/9.20077 | 51/2/0.04/8.35344   | 6/36/0.02         |
|    | 50000  | 3/7/0.03/8.88178e-10 | 4/9/0.04/0        | 15/30/0.07/9.20077 | 51/2/0.11/8.35344   | 6/36/0.05         |
|    | 80000  | 3/7/0.04/8.88178e-10 | 4/9/0.05/0        | 15/30/0.10/9.20077 | 51/2/0.16/8.35344   | 6/36/0.07         |
|    | 100000 | 3/7/0.05/8.88178e-10 | 4/9/0.06/0        | 15/30/0.12/9.20077 | 51/2/0.19/8.35344   | 6/36/0.08         |

**Table 7.** Experimental results of image restoration ( $512 \times 512$ ) for four methods.

| Image | NBBP1                | NBBP2                  | Liu                    | NGCG                   |
|-------|----------------------|------------------------|------------------------|------------------------|
|       | NI/Tcup/SNR/PSNR     | NI/Tcup/SNR/PSNR       | NI/Tcup/SNR/PSNR       | NI/Tcup/SNR/PSNR       |
| 1     | 41/16.95/24.33/37.63 | 38/19.94/24.32/37.63   | 79/33.84/24.29/37.59   | 246/117.16/23.93/37.24 |
| 2     | 41/16.81/23.95/35.20 | 69/41/23.84/35.09      | 118/52.27/23.79/35.04  | 433/203.88/23.10/34.34 |
| 3     | 66/26.78/23.81/36.21 | 68/39.39/23.83/36.23   | 179/74.91/23.56/35.97  | 320/150.20/23.53/35.93 |
| 4     | 62/30.56/24.11/37.38 | 32/18.56/24.60/37.87   | 156/76.67/23.74/37.02  | 273/126.95/23.75/37.02 |
| 5     | 87/41.59/21.10/36.35 | 138/100.17/20.96/36.21 | 213/106.48/20.87/36.12 | 365/178.05/20.80/36.05 |

More` [12] is employed in this study to further clear show the performance of four methods. The set of  $n_p$  problems and solvers are respectively denoted by  $P$  and  $S$ . Let  $t_{p,s}$  be CPU (NI and NF) required to solve problem  $p \in P$  by solver  $s \in S$  and define the performance ratio as

$$r_{p,s} = t_{p,s} / \min_{s \in S} t_{p,s}.$$

The performance profile for each solver  $s$  is defined by

$$\chi(\tau) = \frac{1}{n_p} \text{size} \{p \in P : \log_2 r_{p,s} \leq \tau\},$$

**Table 8.** Experimental results of image restoration ( $256 \times 256$ ) for four methods.

|       | NBBP1                 | NBBP2                 | Liu                   | NGCG                  |
|-------|-----------------------|-----------------------|-----------------------|-----------------------|
| Image | NI/Tcup/SNR/PSNR      | NI/Tcup/SNR/PSNR      | NI/Tcup/SNR/PSNR      | NI/Tcup/SNR/PSNR      |
| 1     | 35/3.61/22.56/32.02   | 148/19.95/22.60/32.06 | 63/7.03/22.49/31.95   | 439/38.70/22.45/31.91 |
| 2     | 54/5.47/19.90/30.99   | 182/31.20/19.93/31.02 | 65/6.80/19.67/30.76   | 546/49.53/19.91/30.99 |
| 3     | 109/10.42/21.91/32.00 | 62/7.97/22.13/32.21   | 175/17.45/21.92/32.01 | 420/38.92/21.73/31.85 |
| 4     | 84/8.45/21.54/31.58   | 181/24.94/21.18/31.22 | 150/15.38/21.47/31.51 | 494/46.56/20.72/30.76 |
| 5     | 74/10.53/24.71/32.98  | 92/16.42/24.70/32.97  | 156/22.00/24.63/32.91 | 527/49.23/23.53/31.80 |



**Figure 4.** From top to bottom: The original images, the blurred images, the restored images by NBBP1, NBBP2, Liu and NGCG, respectively.

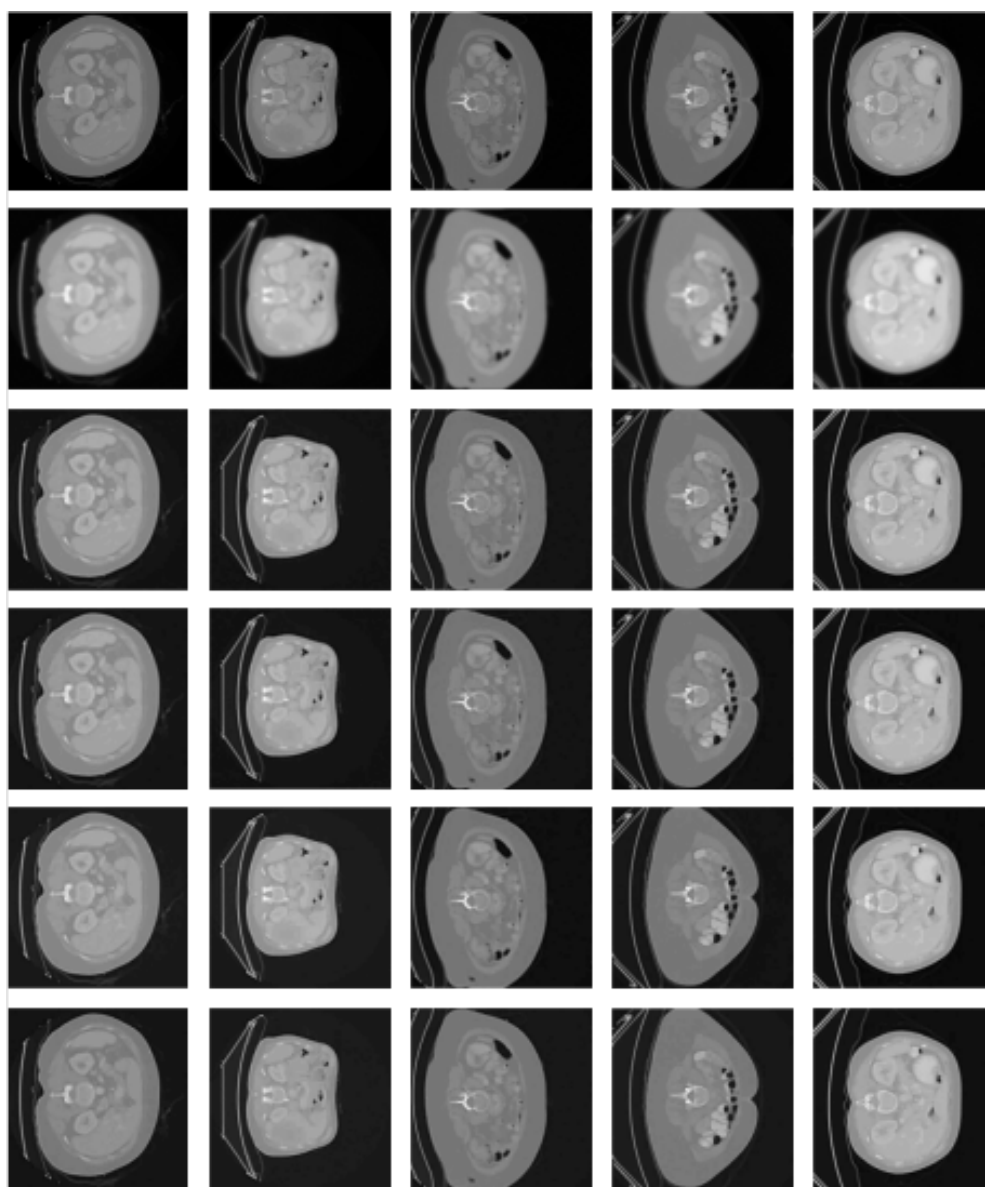


**Figure 5.** From top to bottom: The original images, the blurred images, the restored images by NBBP1, NBBP2, Liu and NGCG, respectively.

where  $\text{size}\{\mathcal{A}\}$  is the number of element  $\mathcal{A}$ . Then,  $\chi(\tau)$  is the probability for solvers  $s \in S$  that  $\log_2 r_{p,s}$  is within a factor  $\tau \in R_+$  of the best possible ratio. Moreover,  $\chi(\tau)$  is the distribution function for the performance ratio.

The performance curves about the four methods in terms of “NI”, “NF” and “CPU” are plotted in Figures 1-3, and obviously, the top curve in each subfigure means that the corresponding method is a winner. It can be seen from Figures 1-3 that the proposed method NBBP1 is more competitive and effective than other methods in terms of any aspect. From Figure 1, we observe that NBBP1 solves nearly 82 percent of the all examples with fewer iterative number than the other methods. From Figure 2, it is obvious that NBBP1 solves approximately 48 percent of the all examples with the least iterative number of function value among the methods. From Figure

3, our method NBBP1 solves 85 percent of the all examples with the least CPU time among the comparison methods.

## 5. Applications to image restoration

### 5.1. General description

In this subsection, to evaluate the performance of the above-mentioned methods in image restoration, we simply introduce the process of transforming an image restoration model into a constrained monotone nonlinear equations. The mathematical model can be written as

$$\min_{x \in R^n} \tau \|x\|_1 + \frac{1}{2} \|b - Ax\|_2^2, \quad (5.1)$$

where  $b \in R^m$  is observed value and  $\tau > 0$  is a parameter,  $A \in R^{m \times n}$  denotes a partial discrete Walsh-Hadamard transform (DWT) matrix and  $m$  rows are randomly chosen from the  $n \times n$  DWT matrix, and  $\|\cdot\|_1$  denotes the so-called  $l_1$ -norm (the sum of absolute values of all elements).

By the technique proposed by Figueiredo, Nowak and Wright [13], Problem (5.1) can be reformulated as a convex quadratic program problem. By splitting  $x \in R^n$  into positive and negative parts, we firstly have

$$x = u - v, \quad u \geq 0, \quad v \geq 0, \quad (5.2)$$

s.t.  $u_i = \max\{0, x_i\}$  and  $v_i = \max\{-x_i, 0\}$  with  $i \in \{1, 2, \dots, n\}$ . Secondly, by denoting  $e_n = (1, 1, \dots, 1)^T \in R^n$ , we can rewrite  $l_1$ -norm as

$$\|x\|_1 = e_n^T u + e_n^T v. \quad (5.3)$$

Combine (5.2) and (5.3) to rewrite Problem (5.1) as the following bound-constrained quadratic program:

$$\min_{u \geq 0, v \geq 0} \frac{1}{2} \|A(u - v) - b\|_2^2 + \tau e_n^T u + \tau e_n^T v,$$

which is equivalent to

$$\min_{z \geq 0} \frac{1}{2} z^T H z + c^T z, \quad (5.4)$$

where

$$z = \begin{pmatrix} u \\ v \end{pmatrix}, \quad c = \begin{pmatrix} \tau e_n - A^T b \\ \tau e_n + A^T b \end{pmatrix}, \quad H = \begin{pmatrix} A^T A & -A^T A \\ -A^T A & A^T A \end{pmatrix}.$$

Obviously, the problem (5.4) is a convex quadratic program problem since the matrix  $H$  is positive semidefinite. According to its first-order optimality condition, it is not difficult to obtain that  $z$  is a minimizer of (5.4) if and only if  $z$  is a solution of the following equations:

$$F(z) = \min\{z, Hz + c\} = 0, \quad z \geq 0, \quad (5.5)$$

where  $F : R^{2n} \rightarrow R^{2n}$  is Lipschitz continuous and monotone (see [29, Lemma 3] and [36, Lemma 2.2]) and the “min” is defined as componentwise minimum.

## 5.2. Experiments on image restoration

Now, we verify the efficiency of our two methods NBBP1 and NBBP2 for restoring blurred images by comparing with Liu and NGCG. All test images come from Test Image Collection. The quality of the restored images by signal-to-noise ratio (SNR) [5] and peak signal noise ratio (PSNR) [5] in dB:

$$\text{SNR} = 20 \times \log_{10} \frac{\|x\|}{\|\bar{x} - x\|},$$

and

$$\text{PSNR} = 10 \times \log_{10} \frac{(N-1)^2}{\frac{1}{MN} \sum_{i,j} (\bar{x}_{i,j} - x_{i,j})^2},$$

where  $x$  denotes the original image,  $x_{i,j}$  denotes the value of  $x$  at the point  $(i, j)$ ,  $\bar{x}$  is the restored image,  $\bar{x}_{i,j}$  denotes the value of  $\bar{x}$  at the point  $(i, j)$ ,  $M$  and  $N$  are the size of the image. The definition of SNR and PSNR implies that the larger is SNR and PSNR, the better is quality of the restored image. In our experiments, we set  $\tau = 0.35$  and the remaining parameters for four methods are chosen as the same as the above section. All methods start their iterations with  $x_0 = A^\top b$  and  $b$  is the blurred image obtained by the original images adding Gauss blur with blur kernel ( $\varpi = 1$ ). The parameter of four methods and the stopping criterion are the same as the above section. The original, blurred and restored images by four methods are showed in Figures 4 and 5. The detailed results are showed in Tables 7-8, where Tcup is the CPU time, SNR and PSNR respectively denoted the restored image SNR value and PSNR value. From Tables 7-8, the restored image quality for the proposed method outperform the compared methods in terms of any aspect. For proposed methods, the NBBP1 method is more competitive than the NBBP2 method.

## 6. Concluding remarks

In this paper, by constructing a new search direction, we present the two modified BB projection methods with Quasi-Newton equation for solving constrained monotone nonlinear equations. We prove the convergence analysis of the proposed methods. Numerical examples and application examples are investigated to show the effectiveness and competitiveness of the proposed methods. In practice, investigating different updates of the involved parameters  $\beta_k$  and  $\gamma_k$  will lead to a new search direction satisfying the Quasi-Newton equation and hence will results in new BB projection methods. We think this will be studied in the future since it is a hybrid direction that is different from the existing methods.

## Acknowledgements

This research was partially supported by the National Natural Science Foundation of China (62302331, 12471298), the Natural Science Foundation of Shanxi Province (202203021212255), and the Shaanxi Fundamental Science Research Project for Mathematics and Physics (23JSQ031).

The datasets generated during the current study are available from the corresponding author on reasonable request. The authors have not disclosed any competing interests.

## References

- [1] M. Abdullahi, A. B. Abubakar, A. Sulaiman and P. Chotpitayasunon, *An efficient projection algorithm for solving convex constrained monotone operator equations and sparse signal reconstruction problems*, The Journal of Analysis, 32, 2024, 2813–2832.

- [2] M. Abdullahi, K. J. Pan, A. Abubakar and A. S. Halilu, *An Inertial Spectral Conjugate Gradient Method for Monotone Nonlinear Equations with Applications*, Mathematical Methods in the Applied Sciences, 2025. DOI: [org/10.1002/mma.10906](https://doi.org/10.1002/mma.10906).
- [3] A. Abubakar, P. Kuman and A. Awwal, *Global convergence via descent modified three-term conjugate gradient projection algorithm with applications to signal recovery*, Results in Applied Mathematics, 2019, 4, 100069.
- [4] J. Barzilai and J. Borwein, *Two-point step size gradient methods*, IMA Journal of Numerical Analysis, 1988, 8, 141–148.
- [5] A. Bovik, *Handbook of Image and Video Processing*, Academic Press, 2000, New York.
- [6] O. Burdakov, Y. Dai and N. Huang, *Stabilized Barzilai-Borwein method*, Journal of Computational Mathematics, 2019, 37, 916–936.
- [7] Y. Dai and R. Fletcher, *Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming*, Numerische Mathematik, 2005, 100, 21–47.
- [8] Y. Dai, W. Hager, K. Schittkowski and H. Zhang, *The cyclic Barzilai-Borwein method for unconstrained optimization*, IMA Journal of Numerical Analysis, 2006, 26, 604–627.
- [9] Y. Dai, Y. Huang and X. Liu, *A family of spectral gradient methods for optimization*, Computational and Applied Mathematics, 2019, 37, 916–936.
- [10] J. Dennis and J. Moré, *Quasi-Newton method, motivation and theory*, SIAM Review, 1977, 19, 46–89.
- [11] S. Dirkse and M. Ferris, *MCPLIB: A collection of nonlinear mixed complementarity problems*, Optimization Methods and Software, 1995, 5, 319–345.
- [12] E. Dolan and J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 2002, 91, 201–213.
- [13] M. Figueiredo, R. Nowak and S. Wright, *Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problem*, IEEE Journal Selected Topics Signal Processing, 2007, 4, 586–597.
- [14] P. Gao and C. He, *A derivative-free projection algorithm involving spectral quotient for solving nonlinear monotone equations*, Optimization, 2018, 67, 1631–1648.
- [15] P. Gao and C. He, *An adaptive family of projection methods for constrained monotone nonlinear equations with applications*, Applied Mathematics and Computation, 359, 2019, 1–16.
- [16] P. Gao and C. He, *An efficient three-term conjugate gradient method for nonlinear monotone equations with convex constraints*, Calcolo, 2018, 55, 53.
- [17] P. Gao, T. Wang, X. Liu and Y. Wu, *An efficient three-term conjugate gradient-based algorithm involving spectral quotient for solving convex constrained monotone nonlinear equations with applications*, Computational and Applied Mathematics, 2022, 41, 1–19.
- [18] P. Gao, W. Zheng, T. Wang, Y. Li and F. Li, *Signal recovery with constrained monotone nonlinear equations through an effective three-term conjugate gradient method*, Journal of Applied Analysis and Computation, 2023, 13, 2006–2025.
- [19] B. Ghaddar, J. Marecek and M. Mevissen, *Optimal power flow as a polynomial optimization problem*, IEEE Transactions on Power Systems, 2015, 31, 539–546.

- [20] M. El-Hawary, *Optimal Power Flow: Solution Techniques, Requirement, and Challenges*, IEEE Service Center, Piscataway, 1996.
- [21] Y. Huang, Y. Dai, X. Liu and H. Zhang, *On the acceleration of the Barzilai-Borwein method*, Computational Optimization and Applications, 2022, 81, 717–740.
- [22] J. B. Jian, Q. X. Huang, J. H. Yin and G. D. Ma, *A family of inertial three-term CGPMs for large-scale nonlinear Pseudo-Monotone equations with convex constraints*, Numerical Linear Algebra with Applications, 2025, 32, e2589.
- [23] J. B. Jian, Z. W. Ren, J. H. Yin and X. Z. Jiang, *A family of hybrid acceleration DFPMs and applications for Pseudo-Monotone nonlinear equations with convex constraints*, Journal of Optimization Theory and Applications, 2026, 209, 26.
- [24] X. Z. Jiang, Y. C. Dong and J. J. Bao, *A memoryless Broyden-Fletcher-Goldfarb-Shanno-type projection method with self-adjusting line search for convexly constrained monotone equations and its applications*, Numerical Algorithm, 2026. DOI: [org/10.1007/s11075-026-02340-9](https://doi.org/10.1007/s11075-026-02340-9).
- [25] J. Liu, B. Tang, N. Zhang, J. Xiong, P. Gao and X. Dong, *A subspace derivative-free projection method for convex constrained nonlinear equations*, Japan Journal of Industrial and Applied Mathematics, 2025, 42, 197–221.
- [26] G. D. Ma, J. C. Jin, J. B. Jian, J. H. Yin and D. L. Han, *A modified inertial three-term conjugate gradient projection method for constrained nonlinear equations with applications in compressed sensing*, Numerical Algorithms, 2023, 92, 1621–1653.
- [27] K. Meintjes and A. Morgan, *A methodology for solving chemical equilibrium systems*, Applied Mathematics and Computation, 1987, 22, 333–361.
- [28] E. Nermeh, M. Abdullahi, A. S. Halilu and H. Abdullahi, *Modification of a conjugate gradient approach for convex constrained nonlinear monotone equations with applications in signal recovery and image restoration*, Communications in Nonlinear Science and Numerical Simulation, 2024, 136, 108079.
- [29] J. Pang, *Inexact Newton methods for the nonlinear complementarity problem*, Mathematical Programming, 1986, 36, 54–71.
- [30] Y. Qu, W. Tong, H. He and Y. Liou, *On projection alternating BB methods for variational inequalities*, Optimization, 2021, 70, 827–846.
- [31] S. B. Salihu, A. S. Halilu, M. Abdullahi, K. Ahmed, P. Mehta and S. Murtala, *An improved spectral conjugate gradient projection method for monotone nonlinear equations with application*, Journal of Applied Mathematics and Computing, 2024, 70, 3879–3915.
- [32] M. Solodov and B. Svaiter, *A new projection method for variational inequality problems*, SIAM Journal on Control and Optimization, 1999, 37, 765–776.
- [33] M. Solodov and B. Svaiter, *Reformulation: Nonsmooth, piecewise smooth, semismooth and smoothing methods*, in: Fukushima, M., Qi, L. (eds.) *A globally convergent inexact Newton method for systems of monotone equations*, Dordrecht: Kluwer Academic Publishers, 1998, 355–369.
- [34] R. Spangler and R. Shoults. *Power generation, operation, and control [book review]*, IEEE Power Energy Magazine, 2014, 12, 90–93.

- [35] C. Wang, Y. Wang and C. Xu, *A projection method for a system of nonlinear monotone equations with convex constraints*, Mathematical Methods of Operations Research, 2007, 66, 33–46.
- [36] Y. Xiao, Q. Wang and Q. Hu, *Non-smooth equations based method for  $l_1$ -norm problems with applications to compressed sensing*, Nonlinear Analysis, 2011, 74, 3570–3577.
- [37] J. Yin, J. Jian and X. Jiang, *A generalized hybrid CGPM-based algorithm for solving large-scale convex constrained equations with applications to image restoration*, Journal of Computational and Applied Mathematics, 2021, 391, 113423.
- [38] Y. X. Yuan and W. Y. Sun, *Optimization Theory and Method*, Science Press, Beijing, 1997.
- [39] W. Zhou and D. Li, *Limited memory BFGS method for nonlinear monotone equations*, Journal of Computational Mathematics, 2007, 25, 89–96.
- [40] W. Zhou and D. Li, *A globally convergent BFGS method for nonlinear monotone equations without any merit functions*, Mathematics of Computation, 2008, 77, 2231–2240.

Received July 2025; Accepted August 2026; Available online October 2026.