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

A new prediction-correction primal-dual hybrid gradient algorithm for solving convex minimization problems with linear constraints.

Fahimeh Alipour · M. R. Eslahchi · Masoud Hajarian

Received: date / Accepted: date

Abstract The primal-dual hybrid gradient (PDHG) algorithm has been applied for solving linearly constrained convex problems. However, it was shown that without some additional assumptions, convergence may fail. In this work, we propose a new competitive prediction-correction primal-dual hybrid gradient algorithm to solve this kind of problem. Under some conditions, we prove the global convergence for the proposed algorithm with the rate of $O(1/N)$ in a nonergodic sense. Comparative performance analysis of our proposed approach with other related methods on some matrix completion and wavelet-based image inpainting test problems shows the outperformance of our approach, in terms of iteration number and CPU time.

Keywords Prediction-correction method · Convex minimization · linear constraints

1 Introduction

We consider the convex minimization problem with linear constraints

$$\min\{f(x) \mid Ax = b\}, \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuous convex (but not necessarily smooth) function; $A \in \mathbb{R}^{m\times n}$, and $b \in \mathbb{R}^m$ are given matrix and vector, respectively and we assume the solution set of (1) is nonempty. The convex problems with linear constraints have wide applications in many areas such as compressed sensing, image processing, data mining, in particular, basis pursuit [26,9], TV-based models [20,25,30], and matrix completion [4,24].

Fahimeh Alipour
E-mail: f.alipour@modares.ac.ir
Department of Applied Mathematics, Faculty of Mathematical Sciences, Tarbiat Modares University, P.O. Box 14115-134, Tehran, Iran

Mohammad Reza Eslahchi
E-mail: eslahchi@modares.ac.ir
Department of Applied Mathematics, Faculty of Mathematical Sciences, Tarbiat Modares University, P.O. Box 14115-134, Tehran, Iran

Masoud Hajarian
E-mail: m_hajarian@sbu.ac.ir
Department of Applied Mathematics, Faculty of Mathematical Sciences, Shahid Beheshti University, Tehran, Iran

The Lagrangian function corresponding to the problem (1) is defined as

$$\mathcal{L}(x, y) = f(x) - y^T(Ax - b),$$

where $y \in \mathbb{R}^m$ is a Lagrangian multiplier. One of the classical approaches for solving (1) is the augmented Lagrangian method (ALM) or the method of multipliers, introduced by Hestenes [19] and Powell [27]. ALM has an essential role from both theoretical and algorithmic points of view, and the main idea of it, is minimizing the penalized Lagrangian function $\mathcal{L}(x, y^k) + \frac{\tau}{2}\|Ax - b\|^2$, where $\tau > 0$ is the penalty parameter and then applying a gradient ascent step on the dual variable. The ALM is defined by the following steps

$$\begin{cases} x^{k+1} = \arg\min_x \{\mathcal{L}(x, y^k) + \frac{\tau}{2}\|Ax - b\|^2\}, & \text{(2a)} \\ y^{k+1} = y^k - \tau(Ax^{k+1} - b). & \text{(2b)} \end{cases}$$

In [29], ALM was signified as the proximal point algorithm (PPA) applied to the dual problem of (1) and its global convergence was investigated under certain assumptions. For the first time, the PPA algorithm was introduced by Martinet [21], and further developed by Rockafellar [29]. Thereafter, many works have been published to modify the classical PPA for convex optimization problems, equilibrium problems, and nonlinear operator problems (for example see [14,16,31,33, 35] and references therein). The proximal operator of the function $f(x)$ is defined by

$$\text{prox}_{\delta f}(p) = \arg\min_x \{f(x) + \frac{1}{2\delta}\|x - p\|^2\}, \tag{3}$$

in which $\delta > 0$ is a constant. In this work, we presume that the proximal operator of the function $f(x)$ is simple to evaluate in the sense that it has a closed-form representation or can be efficiently solved. This property is confirmed by several practical interesting applications such as imaging restoration, matrix completion, and sparse signal reconstruction. For instance, when $f(x)$ denotes the l_1-norm of vector x, the closed-form solution of (3) is $x^* = T_\delta(p)$, where $T_\delta(p) = \text{sign}(p)\max(|p| - \delta, 0)$ is the soft-shrinkage operator; and when $f(X)$ is the nuclear norm of matrix X, the closed-form solution is as follows:

$$\mathcal{D}_\delta(P) = \arg\min_X \{\|X\|_* + \frac{1}{2\delta}\|X - P\|_F^2\},$$

in which $\mathcal{D}_\delta$ is the soft-thresholding operator [2] defined by:

$$\mathcal{D}_\delta(X) := U\mathcal{D}_\delta(\Sigma)V^T, \quad \mathcal{D}_\delta(\Sigma) = \text{diag}(\max\{\sigma_i - \delta, 0\}),$$

where $X = U\Sigma V^T$; the singular value decomposition (SVD) of X (for more examples, see [26] and references therein). It has been used of such special structures in the primal-dual hybrid gradient (PDHG) algorithm, which is very helpful in the algorithmic design of PDHG.

The PDHG approach [37] that was first applied for saddle point form of (1), is as follows

$$\begin{cases} x^{k+1} = \arg\min_x \{\mathcal{L}(x, y^k) + \frac{r}{2}\|x - x^k\|^2\}, & \text{(4a)} \\ y^{k+1} = y^k - \frac{1}{s}(Ax^{k+1} - b), & \text{(4b)} \end{cases}$$

where $r, s > 0$. We can easily see that PDHG is a combination of the classical proximal point algorithm with the Lagrangian function. Notice that the subproblem (4a) is easy to compute at each iteration under the assumption (3) because it estimates the proximal operator of $f(.)$. The PDHG has been proved to be efficient for saddle point problems that have arisen in a lot of image processing applications [1,5,10,36] and is effective if the parameters $r, s > 0$ and satisfy $rs > \|A^T A\|$ [37]. However, it was shown that (4a)-(4b) is may not be convergent if no additional assumptions are applied on the model (see [17] and a counterexample therein). Recently, many works have been extended to achieve convergent PDHG algorithms. Additional suppositions, such as strong convexity for the objective function and applying some conditions to restricting step sizes, could be used to settle this issue. Another option to obtain a convergent PDHG type is to modify the algorithm with some determined steps (for several PDHG variants in imaging processing, see [1,5,6,8,11,13,15,18]).

It is remarkable that assumption (3) has been widely utilized in PPA for solving the problem (1). He et al. designed a customized proximal point algorithm for (1) in [16]. They demonstrated that using a proper measure in the PPA, the outcoming subproblem estimates (3).

A linearized augmented Lagrangian multiplier (LALM) approach has been presented in [34], based on the idea of adding a proximal term to the quadratic penalty term of the augmented Lagrangian (2a), (for a relationship between these algorithms see [7,10]). We remark that the convergence requirement for these primal-dual algorithms is $rs > \|A^T A\|$. On the other hand, as stated in [11], the PDHG's effectiveness strongly depends on the parameters used, and how to choose appropriate step sizes is a critical issue. Algorithmically, convergence would be slow when the values of r and s are large. Because in this situation, the proximal term of (4a) has a substantial weight in the objective function and step sizes could be smaller. As a result, for guaranteeing the convergence, it is preferred to choose as small values of r and s as possible. In fact, the values of r and s are recommended to be selected larger than but very close to $\|A^T A\|$, as experimentally applied in many cases (we refer the readers to [3,11,36] for more numerical examples).

In order to enlarge the step sizes, Ma et al. [23] showed that the convergence condition of PDHG variants can be relaxed by $rs > a\ \|A^T A\|$ $(0 < a < 1)$. They combined the first PDHG with a correction process and introduced a prediction-correction-based PDHG (PC-PDHG) algorithm as follows:

Input: Choose $r, s > 0$ satisfying the $rs > \frac{1}{4}\|A^T A\|$.

Initialization: $(x^0, y^0) \in \Omega$.

Prediction:

$$\begin{cases} \tilde{x}^k = \arg\min_x \left\{\mathcal{L}(x, y^k) + \dfrac{r}{2}\|x - x^k\|^2\right\}, & \text{(5a)} \\ \tilde{y}^k = y^k - \dfrac{1}{s}(A\tilde{x}^k - b). & \text{(5b)} \end{cases}$$

Correction:

$$\begin{cases} x^{k+1} = \tilde{x}^k - \dfrac{1}{2r}A^T\left(y^k - \tilde{y}^k\right), & \text{(5c)} \\ y^{k+1} = y^k - \dfrac{1}{s}\left(A\left(x^{k+1} + \dfrac{1}{2}(\tilde{x}^k - x^k)\right) - b\right). & \text{(5d)} \end{cases}$$

Here, we present a new PDHG variant approach that works much superior to existing algorithms with appropriate selecting of its parameters. We can adjust the parameters of the proposed method so that more iterations would not need comparing of other similar algorithms. The proposed scheme first makes a prediction using the PDHG and then in the correction procedure uses two matrix-vector computations to generate a new iteration. The method utilizes more previous iterative information than PDHG using information from the most recent iterations. The proposed method is presented below.

New prediction-correction-based primal-dual hybrid gradient (NPC-PDHG) method:

Input: Choose $r, s > 0$ and α, β, w_1 and w_2 satisfying the following conditions:

$$1 + \alpha w_1 > 0, \tag{6a}$$
$$\frac{w_1 + w_2}{2} + \alpha w_1(2 - w_2) = 0, \tag{6b}$$
$$(2rs)(1 + \alpha w_1)(1 + \beta w_2 - w_2) > (1 - w_1)(1 - w_2)\big(2\alpha w_1 + \frac{1}{2}\big)\|AA^T\|. \tag{6c}$$

$$1 - \alpha w_1 > 0, \tag{7a}$$
$$\frac{w_1 + w_2}{2} - \alpha w_1(2 - w_2) = 0, \tag{7b}$$
$$(2rs)(1 - \alpha w_1)(1 - \beta w_2 + w_2) > \Big[w_2(1 - w_1)(1 - \alpha w_1) + \frac{(1 + w_1)^2}{2}\Big]\|AA^T\|. \tag{7c}$$

Initialization: $(x^0, y^0) \in \Omega$.

Prediction:

$$\begin{cases} \tilde{x}^k = \arg\min_x \{\mathcal{L}(x, y^k) + \frac{r}{2}\|x - x^k\|^2\}, & (8a) \\ \tilde{y}^k = y^k - \frac{1}{s}(A\tilde{x}^k - b). & (8b) \end{cases}$$

Correction:

$$\begin{cases} x^{k+1} = \tilde{x}^k - \dfrac{(1 - \omega_1)}{2r} A^T\big(y^k - \tilde{y}^k\big) - \alpha\omega_1(x^k - \tilde{x}^k), & (8c) \\ y^{k+1} = y^k - \dfrac{(1 - \omega_2)}{s}\Big(A\Big(x^{k+1} + \frac{1}{2}(\tilde{x}^k - x^k)\Big) - b\Big) - \beta\omega_2(y^k - \tilde{y}^k). & (8d) \end{cases}$$

With a special choice of parameters, our scheme covers some existing methods. For example, if we set $w_1 = w_2 = 0$ the NPC-PDHG method is reduced to PC-PDHG of [23].

The rest of this paper is laid out as follows. Preliminaries are provided in Section 2. In Section 3, we show that the NPC-PDHG method is globally convergent with the $O(1/N)$ rate, in a nonergodic sense. In Section 4, we extend our method to a more general convex problem. Section 5 then presents the computational experiments. Finally, in Section 6, we wrap up our study.

2 Preliminaries

In this section, we define the optimality conditions of (1) as a variational inequality (VI). This VI understanding will pave the way for our upcoming convergence analysis. Solving (1) is equal to identifying a saddle point of $\mathcal{L}$, based on our assumption of (1). Let (x^*, y^*) be a saddle point of $\mathcal{L}$. We've got

$$\underset{y\in\mathbb{R}^m}{\mathcal{L}}(x^*,y) \leq \mathcal{L}(x^*,y^*) \leq \underset{x\in\mathbb{R}^n}{\mathcal{L}}(x,y^*). \tag{9}$$

Then, using variational characterization, we could read off the optimality conditions directly. Finding a saddle point of (9) is equivalent to solving the following mixed variational inequality:

$$f(x) - f(x^*) + (v - v^*)^T J(v^*) \geq 0, \quad \forall v \in \Omega, \tag{10a}$$

with

$$v := \begin{pmatrix} x \\ y \end{pmatrix}, \quad J(v) := \begin{pmatrix} -A^T y \\ Ax - b \end{pmatrix}, \tag{10b}$$

and

$$\Omega = \mathbb{R}^n \times \mathbb{R}^m, \tag{10c}$$

Here, we denote (10a)-(10c) by $VI(\Omega, J)$. The notations $v^k := (x^k, y^k)^T$ and $\tilde{v}^k := (\tilde{x}^k, \tilde{y}^k)^T$ are thus obvious from the context.

It's worth noting that if

$$(J(v) - J(\tilde{v}))^T (v - \tilde{v}) \geq 0, \quad \forall v, \tilde{v} \in \Omega, \tag{11}$$

then the mapping $J(.)$ is said to be *monotone* with respect to Ω. As a result, it's simple to see that $J(v)$ in (10b) is monotone. The solution set of $VI(\Omega, J)$, denoted by Ω^*, is also nonempty under the nonempty assumption on the solution set of (1).

Now we will go through how a variational inequality can be used to characterize the optimality condition of an optimization model.

Proposition 1 *[23] Let $f(x), h(x) : \mathbb{R}^n \to \mathbb{R}$ be convex functions. In addition, $h(x)$ is differentiable. We assume that the solution set of the minimization problem* $\min\{f(x) + h(x)\}$ *is nonempty. Then,*

$$x^* = \arg\min\{f(x) + h(x)\},$$

if and only if

$$f(x) - f(x^*) + (x - x^*)^T \nabla h(x^*) \geq 0, \quad \forall x \in \mathbb{R}^n.$$

In the following lemma, we obtain the conditions for positive definiteness of a matrix with a special structure that repetitively appears in the next section for our convergence analysis.

Lemma 1 *Let the matrix Z has the following structure:*

$$Z = \begin{pmatrix} aI & bA^T \\ cA & dI + eAA^T \end{pmatrix},$$

where the parameters a, b, c, d and e are real scalars and satisfying

$$a > 0, \tag{12a}$$

$$b = c, \tag{12b}$$

$$ad > (bc - ae)\|AA^T\|, \tag{12c}$$

then the matrix Z is symmetric positive definite.

Proof We know Z is positive definite if and only if Z_{11} and $Z_{22} - Z_{21} Z_{11}^{-1} Z_{12}$ (that is Schur complement of Z_{11}), are positive definite matrices, thus from conditions (12a)-(12c) Z is straightly symmetric positive definite. $\square$

3 Convergence analysis

In this section, we establish some lemmas to prove the global convergence of the sequence $\{v^k\}$ generated by (8a)-(8d). For this purpose, we begin with the difference between the point $\tilde{v}^k$ and a solution point of $VI(\Omega, J)$ quantified in the following lemma.

Lemma 2 *Let $\{v^k\}$ and $\{\tilde{v}^k\}$ be generated by the NPC-PDHG algorithm. Then*

$$\tilde{v}^k \in \Omega, \quad f(x) - f(\tilde{x}^k) + (v - \tilde{v}^k)^T J(\tilde{v}^k) \geq (v - \tilde{v}^k)^T Q(v^k - \tilde{v}^k), \qquad \forall v \in \Omega, \tag{13}$$

in which Q is defined by

$$Q = \begin{bmatrix} rI & A^T \\ 0 & sI \end{bmatrix}. \tag{14}$$

Proof According to Proposition 1, the first-order optimality condition for the $x-$subproblem (8a) can be expressed as the following

$$f(x) - f(\tilde{x}^k) + (x - \tilde{x}^k)^T \{-A^T y^k + r(\tilde{x}^k - x^k)\} \geq 0, \quad \forall x \in \mathbb{R}^n. \tag{15}$$

In addition, the (8b) can be written as

$$A\tilde{x}^k - b + s(\tilde{y}^k - y^k) = 0.$$

It can be further expressed as

$$\tilde{y}^k \in \mathbb{R}^m, \quad (y - \tilde{y}^k)^T \{A\tilde{x}^k - b + s(\tilde{y}^k - y^k)\} \geq 0, \quad \forall y \in \mathbb{R}^m. \tag{16}$$

From (15) and (16), we have

$$\begin{aligned} &\tilde{v}^k \in \Omega, \quad f(x) - f(\tilde{x}^k) \\ &+ \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T \tilde{y}^k \\ A\tilde{x}^k - b \end{pmatrix} - \begin{pmatrix} rI & A^T \\ 0 & sI \end{pmatrix} \begin{pmatrix} x^k - \tilde{x}^k \\ y^k - \tilde{y}^k \end{pmatrix} \right\} \geq 0, \quad \forall v \in \Omega. \end{aligned}$$

The proof is complete. □

Lemma 3 *Let $\{v^k\}$ and $\{\tilde{v}^k\}$ be generated by the NPC-PDHG algorithm. Then*

$$v^{k+1} = v^k - M(v^k - \tilde{v}^k), \tag{17}$$

where M is defined by:

$$M = \begin{bmatrix} (1 + \alpha\omega_1)I & \frac{(1-\omega_1)}{2r} A^T \\ \frac{(1-\omega_2)}{s}(\alpha\omega_1 - \frac{1}{2})A & (1 - \omega_2 + \beta\omega_2)I - \frac{(1-\omega_1)(1-\omega_2)}{2rs} AA^T \end{bmatrix}. \tag{18}$$

Proof It follows from (8c) that

$$x^{k+1} = x^k - (1 + \alpha w_1)(x^k - \tilde{x}^k) - \Big(\frac{1 - w_1}{2r}\Big) A^T (y^k - \tilde{y}^k). \tag{19}$$

For the term of (8d), we have

$$\begin{aligned} y^{k+1} &= y^k + \Big(\frac{1 - w_2}{2s}\Big) A(x^k - \tilde{x}^k) - \beta w_2 (y^k - \tilde{y}^k) - \Big(\frac{1 - w_2}{s}\Big)(Ax^{k+1} - b) \\ &= y^k + \Big(\frac{1 - w_2}{2s}\Big) A(x^k - \tilde{x}^k) - \beta w_2 (y^k - \tilde{y}^k) \\ &\quad - \Big(\frac{1 - w_2}{s}\Big) \Big[(A\tilde{x}^k - b) - A(\tilde{x}^k - x^{k+1})\Big]. \end{aligned} \tag{20}$$

Note that from (8b), we get

$$A\tilde{x}^k - b = s(y^k - \tilde{y}^k), \tag{21}$$

and from (8c), we obtain

$$\tilde{x}^k - x^{k+1} = \Big(\frac{1-w_1}{2r}\Big)A^T(y^k - \tilde{y}^k) - \alpha w_1(x^k - \tilde{x}^k), \tag{22}$$

Substituting (21) and (22) into (20), we get

$$\begin{aligned} y^{k+1} = y^k &+ \Big(\frac{1-w_2}{s}\Big)\Big(\frac{1}{2} - \alpha w_1\Big)A\ (x^k - \tilde{x}^k) \\ &- \left[(1 - w_2 + \beta w_2)I - \frac{(1-w_1)(1-w_2)}{2rs}AA^T\right](y^k - \tilde{y}^k). \end{aligned} \tag{23}$$

From (19) and (23), we have

$$\begin{aligned} &\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \end{pmatrix} - \\ &\begin{pmatrix} (1+\alpha\omega_1)I & \frac{(1-\omega_1)}{2r}A^T \\ \frac{(1-\omega_2)}{s}(\alpha\omega_1 - \frac{1}{2})A & (1-\omega_2+\beta\omega_2)I - \frac{(1-\omega_1)(1-\omega_2)}{2rs}AA^T \end{pmatrix} \begin{pmatrix} x^k - \tilde{x}^k \\ y^k - \tilde{y}^k \end{pmatrix}. \end{aligned}$$

The assertation (17) is proved. □

For the matrices Q and M defined in (14) and (18), if there exists a positive definite matrix H such that $Q = HM$, then by using (17), we can rewrite the right-hand side of (13) as

$$(v - \tilde{v}^k)^T Q(v^k - \tilde{v}^k) = (v - \tilde{v}^k)^T H(v^k - v^{k+1}). \tag{24}$$

In the following, we show that the matrix

$$H = QM^{-1} \tag{25}$$

is a symmetric positive definite matrix and also easily seen that $Q = HM$.

Lemma 4 *For any $r, s > 0$ and α, β, w_1 and w_2 satisfying (6a)-(6c), the matrix H defined in (25) is positive definite.*

Proof For the matrix Q defined by (14), we have

$$Q^{-1} = \begin{pmatrix} \frac{1}{r}I & -\frac{1}{rs}A^T \\ 0 & \frac{1}{s}I \end{pmatrix}.$$

Thus, it follows from (18) and (25) that

$$H^{-1} = MQ^{-1} = \begin{pmatrix} \frac{(1+\alpha w_1)}{r}I, & -\frac{1}{rs}\big(\frac{1}{2} + \frac{w_1}{2} + \alpha w_1\big)A^T \\ \frac{(1-w_2)(\alpha w_1 - \frac{1}{2})}{rs}A, & \frac{(1-w_2+\beta w_2)}{s}I - \frac{w_1(1-w_2)(\alpha - \frac{1}{2})}{rs^2}AA^T \end{pmatrix}.$$

According to Lemma 1 and structure of H^{-1} above, it is clear that conditions (6a)-(6c) are equivalent to conditions (12a)-(12c) and thus H^{-1} is symmetric positive definite. The positive definiteness of H is followed directly. □

Notice that we do not need to calculate the matrix H directly. Substituting (24) into (13) and using the monotonicity of J, (11), we get

$$\tilde{v}^k \in \Omega, \quad f(x) - f(\tilde{x}^k) + (v - \tilde{v}^k)^T J(v) \geq (v - \tilde{v}^k)^T H(v^k - v^{k+1}), \quad \forall v \in \Omega. \tag{26}$$

In the following lemma we obtain an equivalent statement for the right-hand side of (26) in terms of $\|v - v^k\|_H$ and $\|v - v^{k+1}\|_H$, where the norm $\|.\|_Z^2$ for the symmetric positive definite matrix Z is defined as $\|v\|_Z^2 = v^T Z v$.

Lemma 5 *Let $\{v^k\}$ and $\{\tilde{v}^k\}$ be generated by the NPC-PDHG algorithm. Then, we have*

$$(v - \tilde{v}^k)^T H(v^k - v^{k+1}) = \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) + \frac{1}{2}\|v^k - \tilde{v}^k\|_G^2, \quad \forall v \in \Omega, \tag{27}$$

and the matrix G is defined as follows:

$$G = Q^T + Q - M^T H M.$$

Proof Applying the identity

$$(a - b)^T H(c - d) = \frac{1}{2}\big(\|a - d\|_H^2 - \|a - c\|_H^2\big) + \frac{1}{2}\big(\|c - b\|_H^2 - \|d - b\|_H^2\big).$$

to the right term of (26) with $a = v$, $b = \tilde{v}^k$, $c = v^k$ and $d = v^{k+1}$, we get

$$\begin{aligned}(v - \tilde{v}^k)^T H(v^k - v^{k+1}) = &\frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) \\ &+ \frac{1}{2}\big(\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2\big).\end{aligned} \tag{28}$$

For the last term of (28), by using (17) and (25), we can write

$$\begin{aligned}&\|v^k - \tilde{v}^k\|_H^2 - \|v^{k+1} - \tilde{v}^k\|_H^2 \\ &= \|v^k - \tilde{v}^k\|_H^2 - \|(v^k - \tilde{v}^k) - (v^k - v^{k+1})\|_H^2 \\ &= \|v^k - \tilde{v}^k\|_H^2 - \|(v^k - \tilde{v}^k) - M(v^k - \tilde{v}^k)\|_H^2 \\ &= 2(v^k - \tilde{v}^k)^T HM(v^k - \tilde{v}^k) - (v^k - \tilde{v}^k)^T M^T HM(v^k - \tilde{v}^k) \\ &= (v^k - \tilde{v}^k)^T\big(Q^T + Q - M^T HM\big)(v^k - \tilde{v}^k).\end{aligned}$$

The allegation is proved. □

Now, we form the matrix G. From (25), we have

$$\begin{aligned}G &= (Q^T + Q) - M^T Q \\ &= \begin{pmatrix} 2rI & A^T \\ A & 2sI \end{pmatrix} - \begin{pmatrix} (1 + \alpha\omega_1)I & \frac{(1-\omega_2)}{s}(\alpha\omega_1 - \frac{1}{2})A^T \\ \frac{(1-\omega_1)}{2r}A & (1 - \omega_2 + \beta\omega_2)I - \frac{(1-\omega_1)(1-\omega_2)}{2rs}AA^T \end{pmatrix}\begin{pmatrix} rI & A^T \\ 0 & sI \end{pmatrix} \\ &= \begin{pmatrix} r(1 - \alpha\omega_1)I & \big[(1 - \omega_2)(\frac{1}{2} - \alpha\omega_1) - \alpha\omega_1\big]A^T \\ \frac{(1+\omega_1)}{2}A & s(1 + \omega_2 - \beta\omega_2)I - \frac{\omega_2(1-\omega_1)}{2r}AA^T \end{pmatrix}.\end{aligned}$$

It is obvious that the matrix G has the same structure of Z in Lemma 1 and hence when $r, s > 0$ and the conditions (7a)-(7c) are satisfied, the matrix G is positive definite. The following theorem proves the contraction property of the method.

Theorem 1 *Assume the conditions (6a)-(6c) and (7a)-(7c) hold and $r, s > 0$. Then, for any $v^* = (x^*, y^*)^T \in \Omega$, the sequence $\{v^k\}$ generated by NPC-PDHG algorithm, satisfies the following inequality:*

$$\|v^{k+1} - v^*\|_H^2 \leq \|v^k - v^*\|_H^2 - \|v^k - \tilde{v}^k\|_G^2.$$

Proof By combining (26) and (27), we get

$$\begin{aligned} \tilde{v}^k \in \Omega, \quad & f(x) - f(\tilde{x}^k) + (v - \tilde{v}^k)^T J(v) \\ & \geq \frac{1}{2}\big(\|v - v^{k+1}\|_H^2 - \|v - v^k\|_H^2\big) + \frac{1}{2}\|v^k - \tilde{v}^k\|_G^2, \quad \forall v \in \Omega. \end{aligned} \tag{29}$$

Setting $v = v^*$ in (29), we obtain

$$\|v^k - v^*\|_H^2 - \|v^{k+1} - v^*\|_H^2 \geq \|v^k - \tilde{v}^k\|_G^2 + 2\{f(\tilde{x}^k) - f(x^*) + (\tilde{v}^k - v^*)^T J(v^*)\}.$$

Since $\tilde{v}^k \in \Omega$, because of the optimality of v^* (10a), we get

$$f(\tilde{x}^k) - f(x^*) + (\tilde{v}^k - v^*)^T J(v^*) \geq 0,$$

and then

$$\|v^k - v^*\|_H^2 - \|v^{k+1} - v^*\|_H^2 \geq \|v^k - \tilde{v}^k\|_G^2.$$

So the proof is complete. □

Now, according to the above results, the algorithm's global convergence can be demonstrated.

Theorem 2 *The sequence $\{v^k\}$ generated by the NPC-PDHG algorithm, converges to a solution point in Ω^*.*

Proof From Theorem 1 we obtain that $\{v^k\}$ is bounded and

$$\lim_{k\to\infty} \|v^k - \tilde{v}^k\| = 0. \tag{30}$$

Therefore, the sequence $\tilde{v}^k$ is also bounded, and there is at least one cluster point in it. Let v^∞ be a cluster point of $\tilde{v}^k$ and $\tilde{v}^{k_j}$ be the subsequence converging to v^∞. From (13), we have

$$\tilde{v}^k \in \Omega, \quad f(x) - f(\tilde{x}^{k_j}) + (v - \tilde{v}^{k_j})^T J(\tilde{x}^{k_j}) \geq (v - \tilde{v}^{k_j})^T Q(v^{k_j} - \tilde{v}^{k_j}), \quad \forall v \in \Omega.$$

Utilizing the continuity of J and f, we get that v^∞ is a solution of $VI(\Omega, J)$.

On the other hand, the subsequence $\{v^{k_j}\}$ also converges to v^∞, by using $\lim_{j\to\infty} \tilde{v}^{k_j} = v^\infty$ and (30). For any $k > k_j$, we have

$$\|v^k - v^\infty\|_H \leq \|v^{k_j} - v^\infty\|_H.$$

Consequently, the sequence $\{v^k\}$ converges to v^∞, that is a solution of $VI(\Omega, J)$. The proof is complete. □

Now, we can conclude the worst-case convergence with rate $O(1/N)$ for the NPC-PDHG, measured by iteration complexity. This result is given in the following theorem that its proof is the same as in [23], thus omitted.

Theorem 3 *Let $\{v^k\}$ and $\{\tilde{v}^k\}$ be generated by NPC-PDHG. Then there exists a constant c_0 such that*

$$\|v^N - v^{N+1}\|_H^2 \leq \frac{1}{(N+1)c_0}\|v^0 - v^*\|_H^2, \quad \forall v^* \in \Omega^*,$$

where H is defined in (25).

Note that Ω^* is closed and convex. Theorem 3 indicates that for any given $\epsilon > 0$, to ensure $\|v^k - v^{k+1}\|_H^2 \leq \epsilon$, the NPC-PDHG algorithm needs at most $[d^2/c_0\epsilon]$ iterations, where $d := inf\{\|v - v^*\|_H \mid v^* \in \Omega^*\}$. Recall that v^{k+1} is a solution of $VI(\Omega, J)$ if $\|v^k - v^{k+1}\|_H^2 = 0$. Thus, a worst-case $O(1/N)$ convergence rate in a nonergodic sense is established for the scheme (8a)-(8d).

4 Extension to separable convex problem

In this section, we demonstrate that, without any major modifications, our algorithm can be naturally extended to the separable convex problem as follows

$$\min_{x\in\mathbb{R}^n}\{f(x) + g(-Ax)\}, \tag{31}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $g : \mathbb{R}^m \to \mathbb{R}$ are both closed convex functions with non-empty domains.

Note that the original PDHG is proposed for solving saddle point problems [5,10,15]. Here, we also reformulate the problem (31) into a saddle point form by Fenchel duality [28]:

$$\min_{x\in\mathbb{R}^n}\max_y\{f(x) - y^T Ax - g^*(y)\}, \tag{32}$$

where $g^*(.)$ is the conjugate function of $g(.)$. Finding a saddle point of problem (32) can be written as the following monotone variational inequality: find (x^*, y^*) such that

$$f(x) + g^*(y) - f(x^*) - g^*(y^*) + (v - v^*)^T J(v^*) \geq 0, \quad \forall v \in \Omega,$$

with

$$v := \begin{pmatrix} x \\ y \end{pmatrix}, \qquad J(v) := \begin{pmatrix} -A^T y \\ Ax \end{pmatrix}. \tag{33}$$

Using (11), the operator $J(.)$ is also monotone.

In a similar way, the primal-dual method presented in [37] for (32) is as follows

$$\begin{cases} x^{k+1} = \arg\min_x \big\{f(x) - (y^k)^T Ax + \frac{r}{2}\|x - x^k\|^2\big\}, & \text{(34a)} \\ y^{k+1} = \arg\max_y \big\{ -g^*(y) - y^T Ax^{k+1} - \frac{s}{2}\|y - y^k\|^2\big\}. & \text{(34b)} \end{cases}$$

We can now offer the new prediction-correction-based primal-dual hybrid gradient method by setting the PDHG as a prediction step. It is as follows:

New prediction-correction-based primal-dual hybrid gradient (NPC-PDHG) method for (32):

Input: Choose $r, s > 0$ and α, β, w_1 and w_2 satisfying the conditions (6a)-(6c) and (7a)-(7c).

Initialization: $(x^0, y^0) \in \Omega$.

Prediction:

$$\begin{cases} \tilde{x}^k = \arg\min_x \{f(x) - (y^k)^T Ax + \frac{r}{2}\|x - x^k\|^2\}, & (35a) \\ \tilde{y}^k = \arg\max_y \{-g^*(y) - y^T A\tilde{x}^k - \frac{s}{2}\|y - y^k\|^2\}. & (35b) \end{cases}$$

Correction:

$$\begin{cases} x^{k+1} = \tilde{x}^k - \frac{(1-\omega_1)}{2r} A^T\big(y^k - \tilde{y}^k\big) - \alpha\omega_1(x^k - \tilde{x}^k), & (35c) \\ y^{k+1} = y^k - \frac{(1-\omega_2)}{s} A\Big(x^{k+1} + \frac{1}{2}(\tilde{x}^k - x^k)\Big) - \beta\omega_2(y^k - \tilde{y}^k). & (35d) \end{cases}$$

The convergence analysis of the NPC-PDHG for the general model (32) is similar to the earlier analysis for (1). The method (35) as a variational inequality in a prediction and correction type is now briefly given.

The first-order optimality condition for the x-subproblem (35a), according to Proposition 1, can be written as

$$f(x) - f(\tilde{x}^k) + (x - \tilde{x}^k)^T\{-A^T y^k + r(\tilde{x}^k - x^k)\} \ge 0, \quad \forall x \in \mathbb{R}^n.$$

And the (35b) can be expressed as

$$g^*(y) - g^*(\tilde{y}^k) + (y - \tilde{y}^k)^T\{A\tilde{x}^k + s(\tilde{y}^k - y^k)\} \ge 0, \quad \forall y \in \mathbb{R}^m.$$

From the above two relations we get

$$\begin{aligned} &\tilde{u}^k \in \Omega, \quad f(x) + g^*(y) - f(\tilde{x}^k) - g^*(\tilde{y}^k) \\ &+ \begin{pmatrix} x - \tilde{x}^k \\ y - \tilde{y}^k \end{pmatrix}^T \left\{ \begin{pmatrix} -A^T\tilde{y}^k \\ A\tilde{x}^k \end{pmatrix} - \begin{pmatrix} rI & A^T \\ 0 & sI \end{pmatrix} \begin{pmatrix} x^k - \tilde{x}^k \\ y^k - \tilde{y}^k \end{pmatrix} \right\} \ge 0, \quad \forall v \in \Omega. \end{aligned}$$

Using the notations of $J(v)$ and Q (see (33) and (14)), we obtain that $\{v^k\}$ and $\{\tilde{v}^k\}$ generated by (35a)-(35d) are satisfied in:

$$\begin{aligned} &f(x) + g^*(y) - f(\tilde{x}^k) - g^*(\tilde{y}^k) + (v - \tilde{v}^k)^T J(\tilde{u}^k) \\ &\ge (v - \tilde{v}^k)^T Q(v^k - \tilde{v}^k), \quad \forall v \in \Omega, \end{aligned}$$

where Q is defined in (14).

Also, we can easily see that for $\{v^k\}$ and $\{\tilde{v}^k\}$ generated by (35a)-(35d), we have

$$v^{k+1} = v^k - M(v^k - \tilde{v}^k),$$

where M is defined in (18).

Thus, all of the convergence analysis performed in Section 3 can be applied to the algorithm in the same way. As a consequent, the sequence $\{v^k\}$ provided by (35a)-(35d) converges to a solution of (32).

5 Numerical experiments

This section is devoted to numerically evaluating the effectiveness of the NPC-PDHG method. We implemented the method on two important problems. All experiments described in this section have been performed in MATLAB R2020b environment on a windows 10 laptop with an Intel 2.40-GHz processor and 4-GB RAM.

5.1 Matrix completion problem

Matrix completion (MC) is a problem of considerable practical interest: the recovery of low-rank matrix X from its incomplete observed entries and has many applications in computer vision, statistics, signal processing, and machine learning. From a mathematical point of view, its convex formulation is as follows

$$\begin{array}{ll} \min & \|X\|_* \\ s.t. & X_{ij} = M_{ij},\ (i,j) \in \Omega, \end{array}$$

where $\|X\|_*$ is the nuclear norm of X, which is defined as $\|X\|_* = \mathrm{trace}(\sqrt{X^T X}) = \sum_{i=1}^{\min\{m,n\}} \sigma_i(X)$, for the matrix X of size $m \times n$; M denotes the unknown matrix with p available sampled entries; Ω is a set of p pairs of indices. By considering $A(X) = X_\Omega$, where X_Ω represents the known entries of X, the MC model can be regarded as a specific example of (1). Since A is a projection operator, we get $\|AA^T\| = 1$.

We use three schemes to test our method in the following; the LALM [34], the customized proximal point algorithm (C-PPA) [16] and the PC-PDHG [23]. As previously stated, all of these methods are based on estimating the proximal operator and do not need the solution of difficult systems. As a result, they have the same ease of implementation.

In our experiments, we only consider random matrices the case $m = n$ and generate two random matrices $M_1 \in \mathbb{R}^{n\times ra}$ and $M_2 \in \mathbb{R}^{n\times ra}$ from i.i.d. standard Gaussian distribution and then set the low-rank matrix $M = M_1 M_2^T$. We sample uniformly at random p entries from M and collect them in Ω. We denote by $\mathrm{OS} = p/dr$ the oversampling ratio, where $dr = ra(2n - ra)$ is the "true dimensionality" or degree of freedom of a $n \times n$ matrix of rank ra. We define the relative error of an approximation X by

$$RE = \frac{\|X_\Omega - M_\Omega\|_F}{\|M_\Omega\|_F}.$$

For all algorithms, we set the relative error tolerance to $Tol = 10^{-5}$. The initial point in all of the tests is $(x^0, y^0) = (0, 0)$.

For all the tests, we tried different values of parameters involved in the algorithms and then we use $r = 0.008$, $s = 1.01/(1.84\ r)$, $w_1 = -0.1$, $w_2 = 0.1$, $\alpha = 0$ and $\beta = 3.7$ for the NPC-PDHG algorithm and for other algorithms we set the same parameters that are reported to work well in their main articles: [34], [16] and [23]. Namely we take $\tau = 1$, $\beta = 2.5/n$ for LALM, $r = 0.008$, $s = 1.01/2r$ for PC-PDHG and $s = 1.01/r$, $\gamma = 1.6$ for C-PPA.

Tables 1-3 show the comparisons of above methods with $\mathrm{OS} = 5, 10, 15$, respectively. From the numerical results of the tables, we found that the proposed NPC-PDHG outperforms other algorithms in terms of iteration number and computational time. The plot of relative errors as versus iteration and CPU time for four algorithms with $n = 1000$, $ra = 10$ and $\mathrm{OS} = 5$, is shown in Fig. 1 (in the logarithmic scale). For the other choices of n, the figure is similar to this case.

Table 1 Comparison results of LALM, C-PPA, PC-PDHG, and NPC-PDHG (OS=5)

Problems		LALM [34]		C-PPA [16]		PC-PDHG [23]		NPC-PDHG	
n	ra	Iter.	RE	Iter.	RE	Iter.	RE	Iter.	RE
100	10	17	5.69e-06	21	5.57e-06	14	7.97e-06	7	8.60e-06
200	10	33	7.36e-06	36	8.05e-06	20	8.67e-06	13	7.81e-06
500	10	81	9.39e-06	62	9.15e-06	43	8.73e-06	29	8.78e-06
1000	10	188	9.67e-06	104	9.06e-06	75	9.26e-06	50	9.51e-06
1000	50	32	9.17e-06	52	9.16e-06	28	8.67e-06	19	9.62e-06
1000	100	14	9.45e-06	23	8.37e-06	12	3.89e-06	8	8.65e-06
3000	10	478	9.96e-06	270	8.36e-06	189	9.92e-06	120	9.77e-06
3000	50	90	8.74e-06	159	9.02e-06	89	9.65e-06	56	9.61e-06
3000	100	45	8.36e-06	103	7.78e-06	53	9.20e-06	34	9.91e-06

Table 2 Comparison results of LALM, C-PPA, PC-PDHG, and NPC-PDHG (OS=10)

Problems		LALM [34]		C-PPA [16]		PC-PDHG [23]		NPC-PDHG	
n	ra	Iter.	RE	Iter.	RE	Iter.	RE	Iter.	RE
200	10	11	3.11e-06	21	6.54e-06	7	8.51e-06	6	6.03e-06
500	10	39	8.75e-06	47	6.81e-06	25	9.37e-06	18	9.40e-06
1000	10	82	8.96e-06	75	9.42e-06	49	5.59e-06	34	9.18e-06
1000	50	10	6.85e-06	22	8.05e-06	8	8.38e-06	6	8.86e-06
3000	10	243	9.62e-06	216	2.57e-06	131	9.73e-06	86	9.91e-06
3000	50	47	9.81e-06	91	9.90e-06	40	9.49e-06	27	9.83e-06
3000	100	22	6.90e-06	46	7.92e-06	19	9.78e-06	13	9.25e-06

Table 3 Comparison results of LALM, C-PPA, PC-PDHG, and NPC-PDHG (OS=15)

Problems		LALM [34]		C-PPA [16]		PC-PDHG [23]		NPC-PDHG	
n	ra	Iter.	RE	Iter.	RE	Iter.	RE	Iter.	RE
300	10	10	4.83e-06	23	6.43e-06	7	7.52e-06	7	5.54e-06
500	10	25	5.20e-06	33	7.67e-06	17	8.13e-06	12	9.10e-06
1000	10	55	7.96e-06	61	9.60e-06	36	8.83e-06	25	9.77e-06
1000	30	13	7.56e-06	24	7.42e-06	8	5.13e-06	6	7.64e-06
3000	15	118	8.18e-06	145	9.70e-06	76	9.62e-06	49	9.79e-06
3000	50	32	5.03e-06	58	8.76e-06	25	9.48e-06	18	7.09e-06
3000	100	10	5.82e-06	26	8.34e-06	8	6.09e-06	9	6.58e-06

5.1.1 Image proccessing problem

Here, we reported the numerical results obtained for four algorithms on the wavelet-based image reconstruction problem. First, we provide a brief description of the wavelet-based image inpainting background. Let $\mathbf{x} \in \mathbb{R}^n$ be a digital image with $n = n_1 \times n_2$, and $W \in \mathbb{R}^{n\times m}$ be a wavelet dictionary (the matrix whose columns are the elements of a wavelet frame, and in general normalized to a unit l_2-norm). The image $\mathbf{x}$ is often sparse under wavelet transform W and we have $\mathbf{x} = Wx$ with x being a sparse vector (we refer to [32] for more details). Wavelet-based image inpainting is an inverse problem that consists in finding an approximation of the original clean image $\mathbf{x}$ based on some observation b which might have some convolutions or missing pixels, by solving the following minimization problem

$$\begin{aligned} \min \quad & \|x\|_1 \\ s.t. \quad & BWx = b, \end{aligned}$$

where the l_1 norm in the objective function is to deduce a sparse representation under the wavelet dictionary. The mask $B \in \mathbb{R}^{l\times n}$ is a diagonal matrix with 0 and 1 diagonal elements and is a

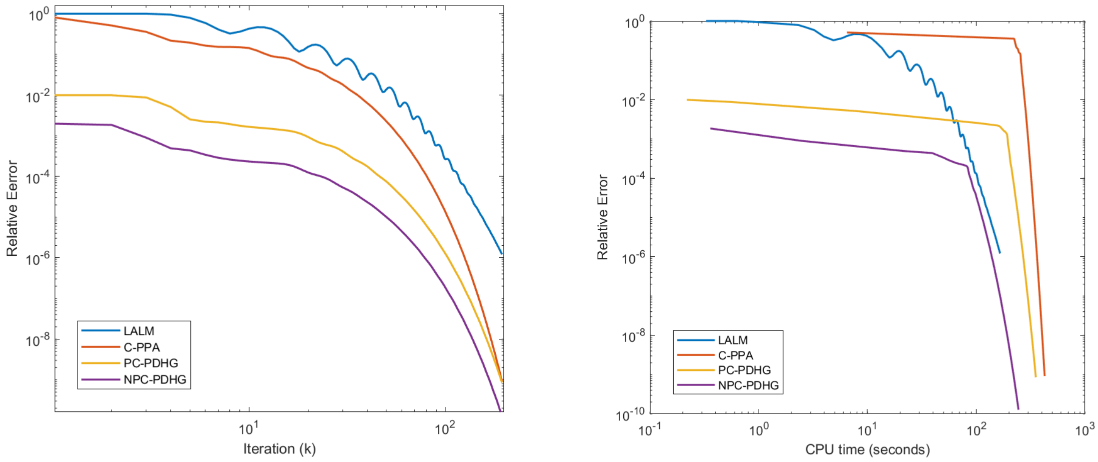

Fig. 1 Performance comparison of LALM, C-PPA, PC-PDHG, and NPC-PDHG algorithms. Relative error versus iteration number (left) and relative error versus CPU time (right).

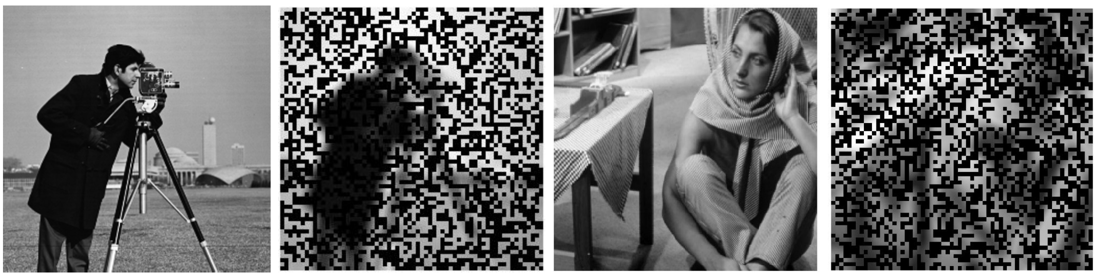

Fig. 2 From left to right: original Cameraman, degraded Cameraman, original Barbara, degraded Barbara

matrix representation of downsampling or blurry operators. The dictionary W has the property $W^T W = I$, hence $\|A^T A\| = \|W^T B^T BW\| = 1$ $(A := BW)$.

In our performances, we consider the reflective boundary condition to create blurry images. In this situation, the mask B can be written as $B = SH$, where $S \in \mathbb{R}^{l\times n}$ is a binary downsampling matrix in which the locations of 0 and 1 correspond to missing and known pixels of the image, respectively, and $H \in \mathbb{R}^{n\times n}$ is a blurry operator that can be diagonalized by the discrete cosine transform (DCT) (for more details we refer to [12]). We use the inverse discrete Haar wavelet transform with a level of 6 for the dictionary W. For the blurring effect, we utilize the out-of-focus kernel with a radius of 7, using the functions `fspecial` and `imfilter` from the MATLAB Image Processing Toolbox. Finally, we hide 50% of pixels by applying a mask operator S. The locations of missing pixels are assigned randomly. Fig. 2 depicts the original and degraded images of Barbara.png and Cameraman.tif (with the size of 256×256). For all the tests, we take $r = 0.6$, $s = 0.6$, $w_1 = -0.1$, $w_2 = 0.1$, $\alpha = 0$ and $\beta = 3.7$ for NPC-PDHG, $r = 0.8$, $s = 0.7$ for PC-PDHG, $r = 0.6$, $s = 1.7$, $\gamma = 1.9$, for C-PPA and $\tau = 1$, $\beta = 2.5/n$ for LALM as suggested in [23, 16, 34]. We set the initial point to be $(x^0, y^0) = (W^T(b), 0)$ and employed 500 iterations for each algorithm. Fig. 3 shows the reconstructed images. The quality of the restoration is measured by

Fig. 3 Restored images by the tested algorithms

the signal-to-noise ratio (SNR) in decibel (dB) that is defined by

$$\mathrm{SNR} := 10 \log_{10} \frac{\|\mathbf{x}\|}{\|\bar{\mathbf{x}} - \mathbf{x}\|}, \tag{36}$$

where $\bar{\mathbf{x}}$ is restored image and $\mathbf{x}$ is an original image. Fig. 5 depicts the NPC-PDHG method might initially (for the small values of SNR) require a little more time than the C-PPA method, however, the further we go, it needs less time to achieve the acceptable values of SNR for the restored images compared to the other three algorithms. Also, Fig. 4 illustrates the outperformance of the NPC-PDHG in terms of iteration number, for the aforementioned images. We constructed these profiles based on SNR and both iteration number and CPU time for all the methods.

6 Conclusion

We presented a new PDHG variant method to solve the linearly constrained convex problem, in this work. The algorithm tries to improve performance by a prediction-correction approach. It first utilizes a prediction by the original PDHG and then updates the new iterate using two correction stages. We established the global convergence of the method that requires the parameters involved in the algorithm to satisfy (6a)-(6c) and (7a)-(7c). Then we concluded the $O(1/N)$ nonergodic convergence rate for the method. we showed that our proposed method can be extended to separable convex minimization. Finally, we compared our method with three related existing algorithms proposed in [16, 34, 23]. Numerical results showed the outperformance of our algorithm than other algorithms, in terms of both iteration number and CPU time.

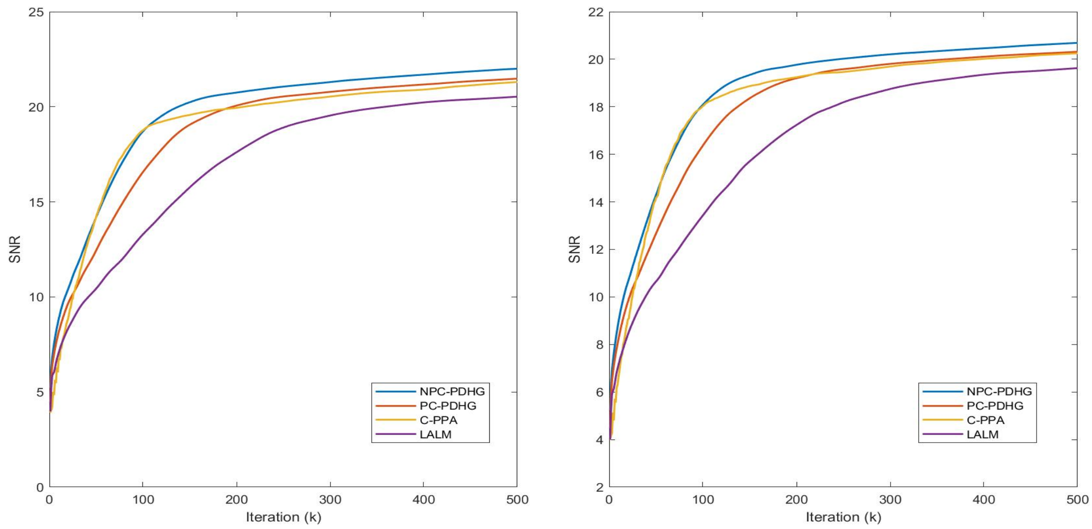

Fig. 4 Performance comparison of LALM, C-PPA, PC-PDHG, and NPC-PDHG algorithms (SNR vs iteration number) on images 'Cameraman' (left) and 'Barbara' (right).

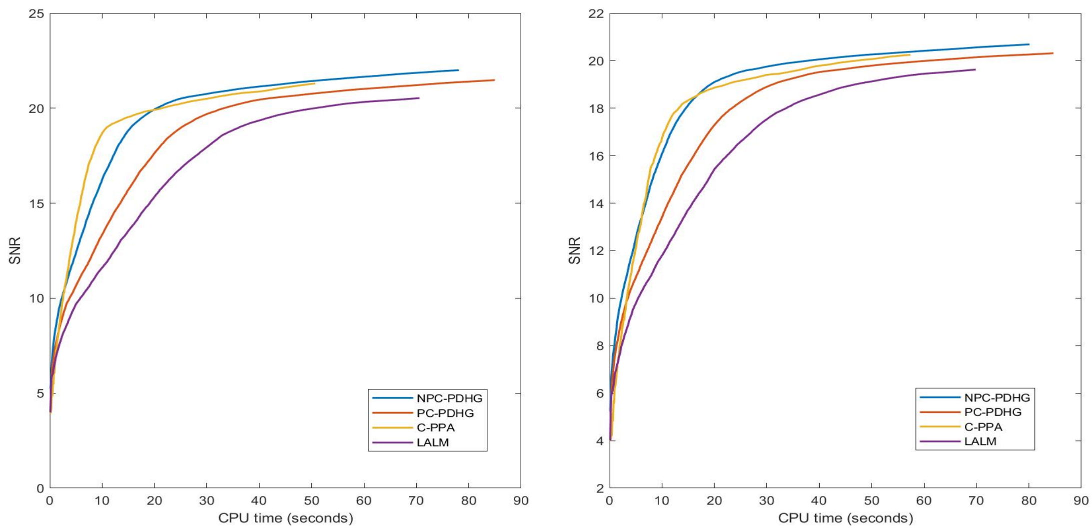

Fig. 5 Performance comparison of LALM, C-PPA, PC-PDHG, and NPC-PDHG (SNR vs CPU time) algorithms on images 'Cameraman' (left) and 'Barbara' (right).

References

1. Bonettini, S., Ruggiero, V.: On the convergence of primal-dual hybrid gradient algorithms for total variation image restoration. J. Math. Imaging Vision 44, 236-253 (2012)
2. Cai, J.F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. 20, 1956-1982 (2010)
3. Cai, X.J., Han, D.R., Xu, L.L.: An improved first-order primal-dual algorithm with a new correction step. J. Glob. Optim. 57, 1419-1428 (2013)
4. Candès, E.J., Recht, B.: Exact matrix completion via convex optimization. Found Comput. Math. 9, 717-772 (2009)

5. Chambolle, A., Pock, T.: A first-order primal-dual algorithms for convex problem with applications to imaging. J. Math. Imaging Vis. 40, 120-145 (2011)
6. Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal-dual algorithm. Math. Prog., Series A 159, 253-287 (2016)
7. Chan, R.H., Ma, S.Q., Yang, J.F.: Inertial primal-dual algorithms for structured convex optimization. arXiv:1409.2992 (2014)
8. Chen, Y., Lan, G., Ouyang, Y.: Optimal primal-dual methods for a class of saddle point problems. SIAM J. Optim. 24, 1779-1814 (2014)
9. Chen, S.S., Donoho, D.L., Saunders, M.A.: Atomic decomposition by basis pursuit. SIAM Rev. 43, 129-159 (2001)
10. Esser, E., Zhang, X.Q., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. SIAM J. Imag. Sci. 3, 1015-1046 (2010)
11. Goldstein, T., Li, M., Yuan, X.: Adaptive Primal-Dual Splitting Methods for Statistical Learning and Image Processing. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems 28, pp. 2089-2097. Curran Associates, Inc. (2015). http://papers.nips.cc/paper/5723-adaptive-primal-dual-splitting-methods-for-statistical-learningand-image-processing.pdf
12. Hansen, P. C., Nagy, J. G., O'leary, D. P.: Deblurring images: matrices, spectra, and filtering. Society for Industrial and Applied Mathematics (2006)
13. He, B.S., Ma, F., Yuan, X.M.: An algorithmic framework of generalized primal-dual hybrid gradient methods for saddle point problems. J. Math. Imaging Vis. 58, 279-293 (2017)
14. He, B.S., Xu, M.H., Yuan, X.M.: Solving large-scale least squares semidefinite programming by alternating direction methods. SIAM J. Matrix Anal. Appl. 3, 136152 (2011)
15. He, B.S., Yuan, X.M.: Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective. SIAM J. Imaging Sci. 5, 119-149 (2012)
16. He, B.S., Yuan, X.M., Zhang, W.X.: A customized proximal point algorithm for convex minimization with linear constraints. Comput. Optim Appl. 56, 559-572 (2013)
17. He, B.S., You, Y.F., Yuan, X.M.: On the convergence of primal-dual hybrid gradient algorithm. SIAM J. Imaging Sci. 7, 2526-2537 (2014)
18. He, H.J., Desai, J., Wang, K.: A primal-dual prediction-correction algorithm for saddle point optimization. J. Glob. Optim. 66, 573-583 (2016)
19. Hestenes, M.R.: Multiplier and gradient methods. J. Optim. Theory Appli. 4, 303-320 (1969)
20. Hu, L., Zhang, W., Cai, X., Han, D.: A parallel operator splitting algorithm for solving constrained total-variation retinex. Inverse Problems and Imaging 14 (6), 1135-1156 (2020)
21. Martinet, B.: Regularision dinequations variationnelles par approximations successive. Revue Fran-caise dAutomatique et Informatique Recherche Operationnelle 126, 154159 (1970)
22. Ma, F., Ni, M.F.: A class of customized proximal point algorithms for linearly constrained convex optimization. Comput. Appl. Math. 37, 896911 (2018)
23. Ma, F., Bi, Y., Gao, B.: A prediction-correction-based primaldual hybrid gradient method for linearly constrained convex minimization. Numerical Algorithms, 82, 641-662, (2019)
24. Ma, S.Q., Goldfarb, D., Chen, L.: Fixed point and Bregman iterative methods for matrix rank minimization. Math. Prog. Ser. A. 128, 321-353 (2011)
25. Ng, M. K., Ngan, H.Y., Yuan, X., Zhang, W.: Lattice-based patterned fabric inspection by using total variation with sparsity and low-rank representations. SIAM Journal on Imaging Sciences, 10(4), 2140-2164 (2017)
26. Parikh, N., Boyd, S.: Proximal algorithms. Found. Trends Optim. 1, 127-239 (2014)
27. Powell, M.J.D.: A Method for Nonlinear Constraints in Minimization Problems. In: Fletcher, R. (ed.) Optimization, pp. 283-298. Academic Press, New York (1969)
28. Rockafellar, R.T.: Convex analysis. Princeton University Press, (1970)
29. Rockafellar, R.T.: Monotone operators and the proximal point algorithm, SIAM J. Control Optim. 14, 877-898 (1976)
30. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D 60, 259-268 (1992)
31. Shen, Y., Zhang, W., He, B.S.: Relaxed augmented Lagrangian-based proximal point algorithms for convex optimization with linear constraints. J. Ind. Manag. Optim. 10, 743759 (2014)
32. Starck, J.L., Murtagh, F., Fadili, J.M.: Sparse image and signal processing, wavelets, curvelets, morphological diversity. Cambridge University Press, Cambridge (2010)
33. Takahashi, W., Zembayashi, K.: Strong and weak convergence theorems for equilibrium problems and relatively nonexpansive mappings in Banach spaces. Nonlinear Anal. 70, 4557 (2008)
34. Yang, J.F., Yuan, X.M.: Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. Math. Comp. 82, 301-329 (2013)
35. Yin, W., Osher, S., Goldfarb, D., Darbon, J.: Bregman iterative algorithms for l_1-minimization with applications to compressed sensing. SIAM J. Imaging Sci. 1, 143168 (2008)

36. Zhang, X.Q., Burger, M., Osher, S.: A unified primal-dual algorithm framework based on Bregman iteration. J. Sci. Comput. 46, 20-46 (2010)
37. Zhu, M., Chan, T.F.: An efficient primal-dual hybrid gradient algorithm for total variation image restoration, CAM Report 08-34, UCLA USA (2008)