

## A SYMMETRIC PRIMAL-DUAL ALGORITHMIC FRAMEWORK FOR SADDLE POINT PROBLEMS\*

Hongjin He<sup>1)</sup>

*School of Mathematics and Statistics, Ningbo University, Ningbo 315211, China*

*Email: [hehongjin@nbu.edu.cn](mailto:hehongjin@nbu.edu.cn)*

Kai Wang

*School of Mathematics and Statistics, Nanjing University of Science and Technology,*

*Nanjing 210094, China*

*Email: [wangkaihawk@njust.edu.cn](mailto:wangkaihawk@njust.edu.cn)*

Jintao Yu

*School of Mathematics and Statistics, Ningbo University, Ningbo 315211, China*

*Email: [yujintao0045@163.com](mailto:yujintao0045@163.com)*

### Abstract

In this paper, we propose a new primal-dual algorithmic framework for a class of convex-concave saddle point problems frequently arising from image processing and machine learning. Our algorithmic framework updates the primal variable between the twice calculations of the dual variable, thereby appearing a symmetric iterative scheme, which is accordingly called the symmetric primal-dual algorithm (SPIDA). It is noteworthy that the subproblems of our SPIDA are equipped with Bregman proximal regularization terms, which make SPIDA versatile in the sense that it enjoys an algorithmic framework to understand the iterative schemes of some existing algorithms, such as the classical augmented Lagrangian method (ALM), linearized ALM, and Jacobian splitting algorithms for linearly constrained optimization problems. Besides, our algorithmic framework allows us to derive some customized versions so that SPIDA works as efficiently as possible for structured optimization problems. Theoretically, under some mild conditions, we prove the global convergence of SPIDA and estimate the linear convergence rate under a generalized error bound condition defined by Bregman distance. Finally, a series of numerical experiments on the basis pursuit, robust principal component analysis, and image restoration demonstrate that our SPIDA works well on synthetic and real-world datasets.

*Mathematics subject classification:* 90C25, 90C47, 90C90.

*Key words:* Primal-dual algorithm, Saddle point problem, Bregman distance, Augmented Lagrangian method, Convex programming.

### 1. Introduction

Recently, saddle point problems have received considerable attention in the signal/image processing, machine learning, and optimization communities, e.g. see [10, 11, 19, 53, 63], to name just a few. In this paper, we are interested in the convex-concave saddle point problem with a bilinear coupling term, which takes the following form:

$$\min_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} \{\mathcal{L}(x, y) := f(x) + \langle Ax, y \rangle - g(y)\}, \quad (1.1)$$

---

\* Received May 9, 2024 / Revised version received January 14, 2025 / Accepted May 12, 2025 /

Published online October 17, 2025 /

<sup>1)</sup> Corresponding author

where  $\mathcal{X} \subseteq \mathbb{R}^n$  and  $\mathcal{Y} \subseteq \mathbb{R}^m$  are two closed nonempty convex sets, both  $f(\cdot) : \mathcal{X} \rightarrow (-\infty, \infty]$  and  $g(\cdot) : \mathcal{Y} \rightarrow (-\infty, \infty]$  are proper closed convex (possibly nonsmooth) functions,  $\langle \cdot, \cdot \rangle$  represents the standard inner product of vectors, and  $A : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is a bounded linear operator. It is interesting that (1.1) provides a unified framework for the treatment of convex composite optimization problem

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

and the canonical convex minimization problem with linear constraints (see Section 4), where  $g^*(\cdot)$  is the Fenchel conjugate of function  $g(\cdot)$ .

To efficiently exploit the min-max structure of (1.1), a seminal work can be traced back to the Arrow-Hurwicz primal-dual (AHPD) method [1], which updates the primal and dual variables in a sequential order by solving two optimization subproblems as follows:

$$\begin{cases} x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \langle Ax, y^k \rangle + \frac{\mu}{2} \|x - x^k\|^2 \right\}, \\ y^{k+1} = \arg \max_{y \in \mathcal{Y}} \left\{ -g(y) + \langle Ax^{k+1}, y \rangle - \frac{\gamma}{2} \|y - y^k\|^2 \right\}, \end{cases} \quad (1.2)$$

where  $\mu$  and  $\gamma$  are two positive proximal regularization parameters serving as step sizes for updating. In the literature, such a method is also reemphasized as primal-dual hybrid gradient (PDHG) method with fruitful applications in image processing [5, 19, 27, 63]. Although some convergence properties have been established under additional conditions [19, 30, 46], the most recent work [29] showed that the AHPD method with any constant step size is not necessarily convergent for solving generic convex-concave saddle point problems. In 2011, Chambolle and Pock [10] judiciously introduced a first-order primal-dual algorithm by absorbing an extrapolation step for algorithmic acceleration. Note that such an algorithm is commonly denoted by PDHG in the optimization literature. Therefore, we also use PDHG to represent the first-order primal-dual algorithm [10] throughout this paper. For given the  $k$ -th iterate  $(x^k, y^k)$ , the iterative scheme of the PDHG [10] reads as

$$\begin{cases} x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \langle Ax, y^k \rangle + \frac{\mu}{2} \|x - x^k\|^2 \right\}, \\ \tilde{x}^{k+1} = x^{k+1} + \tau(x^{k+1} - x^k), \\ y^{k+1} = \arg \max_{y \in \mathcal{Y}} \left\{ -g(y) + \langle A\tilde{x}^{k+1}, y \rangle - \frac{\gamma}{2} \|y - y^k\|^2 \right\}, \end{cases} \quad (1.3)$$

where  $\tau \in [0, 1]$  is an extrapolation parameter and both  $\mu > 0$  and  $\gamma > 0$  are regularization parameters. In particular, they further proved some convergence properties for the special case  $\tau = 1$  under the following requirement:

$$\|AA^\top\| < \mu\gamma. \quad (1.4)$$

It is notable that the PDHG not only requires weaker convergence-guaranteeing conditions, but also runs faster than the AHPD method (see [10, 12]). In recent years, there are some papers contributed to further studies on the extrapolation step, e.g. see [8, 13, 31, 33, 58]. As aforementioned, both  $\mu$  and  $\gamma$  serve as step sizes for updating. From computational perspective, larger step sizes usually lead to faster convergence, which accordingly encourages researchers to relax condition (1.4), e.g. see [26, 38, 41, 42]. When the  $x$ - and  $y$ -subproblems are not easily to be solved or the maximum eigenvalue of  $AA^\top$  cannot be easily evaluated in some cases, a better way is to solve the underlying subproblems in an inexact way or the employment of line search

for avoiding the calculation of  $\|AA^\top\|$ , e.g. see [14, 36, 37, 44, 52]. In recent years, saddle point problems and primal-dual algorithms received much attention in the fields of machine learning, imaging science, and optimization. Here, we only refer the reader to [16, 39, 53, 56] for recent surveys and references therein along this direction.

It is well-known that the saddle point problem (1.1) provides a powerful treatment for linearly constrained convex optimization problems (e.g. see Section 4). In this application, the dual variable  $y$  in (1.1) serves as the so-called Lagrangian multiplier. However, as shown in the excellent overview on a phenomenon of slow convergence of optimization algorithms [35] (also see some comments [20, 45, 47, 54]), critical multipliers are nonempty for optimal solutions with nonunique Lagrangian multipliers, which play a crucial negative role in numerical optimization yielding slow convergence of major primal-dual algorithms, including Newton and Newton-related methods, the augmented Lagrangian method, and the sequential quadratic programming method. Therefore, these surprising discoveries clearly demonstrate that the updating scheme of the dual variable (i.e. Lagrangian multiplier) is very important for algorithmic acceleration, which motivates us to develop some “dual stabilization techniques” for solving (1.1). Besides, most of saddle point problems arising from machine learning and image processing display unbalanced primal and dual subproblems in the sense that the dual problem is often easier than the primal one (e.g. see [10, 11] and also Section 4). As studied in the most recent work [32, 43], balancing the subproblems of the classical ALM is able to greatly speed up the convergence of solving linearly constrained optimization problems. Therefore, how to balance both subproblems and what will be produced by some balancing technique for (1.1) are also motivations of this paper.

Considering the different complexity of primal and dual subproblems, we in this paper employ the symmetric spirit to design a new primal-dual algorithmic framework for saddle point problem (1.1), where the primal variable is updated once between the twice calculations of the dual variable. To a certain extent, our algorithm is able to balance the computation of primal and dual subproblems, and the one more calculation of the dual variable can be regarded as some dual stabilization technique from numerical perspective. Since the proposed algorithm appears a symmetric updating order on the dual variable, we call it symmetric primal-dual algorithm and denote it by SPIDA for simplicity. A toy example shows that our SPIDA has a nice convergence behavior, while the AHPD fails to converge and the PDHG runs a little slower than our SPIDA for some proximal parameters. Notice that each subproblem is equipped with a Bregman proximal term to make our algorithm versatile so that we can easily derive the iterative schemes of some classical first-order optimization methods, including the ALM and its linearized version for one-block linearly constrained convex optimization problems, and some Jacobian splitting algorithms for multi-block linearly constrained convex minimization problems. Particularly, our algorithmic framework is of benefit for producing some customized variants for linearly constrained optimization problems. Theoretically, we prove that our SPIDA is globally convergent under standard conditions, while the linear convergence rate is also estimated under a generalized error bound condition defined by Bregman distance. Finally, a series of numerical experiments on basis pursuit, robust principal component analysis (RPCA), and image restoration demonstrate that our SPIDA performs better than some state-of-the-art primal-dual algorithms in many cases.

The remainder of paper is organized as follows. In Section 2, we recall some notations and definitions that will be used throughout this paper. In Section 3, we first describe the details of our proposed SPIDA for (1.1). Then, we prove its global convergence and estimate

its linear convergence rate. In Section 4, we apply our SPIDA to solve the linearly constrained convex minimization problems, and show that some classical first-order optimization methods are special cases of the proposed SPIDA. In Section 5, we conduct the numerical performance of our SPIDA on solving some structured optimization problems with synthetic and real datasets. Finally, we complete this paper with drawing some conclusions in Section 6.

## 2. Preliminaries

In this section, we summarize some notations and basic concepts that will be used in subsequent analysis.

Throughout this paper, the superscript symbol  $^\top$  represents the transpose for vectors and matrices. Let  $\mathbb{R}^n$  be an  $n$ -dimensional Euclidean space endowed with the  $M$ -inner product  $\langle x, y \rangle_M = \langle x, My \rangle = x^\top My$ , where  $x, y \in \mathbb{R}^n$  and  $M$  is a symmetric and positive definite (or semi-definite) matrix ( $M \succ 0$  (or  $\succeq 0$ ) for short). Consequently, for a given vector  $x \in \mathbb{R}^n$ , we define the  $M$ -norm by

$$\|x\|_M = \sqrt{\langle x, Mx \rangle}.$$

In particular, when  $M$  is an identity matrix, the  $M$ -norm reduces to the standard Euclidean norm (denoted by  $\|x\|$ ). Moreover, for a matrix  $A$ , we use  $\|A\|$  to represent the square root of the maximum eigenvalue of  $A^\top A$ .

**Definition 2.1.** Let  $f(\cdot) : \mathbb{R}^n \rightarrow [-\infty, +\infty]$  be an extended real-valued function, and denote the domain of  $f(\cdot)$  by  $\text{dom}(f) := \{x \in \mathbb{R}^n \mid f(x) < \infty\}$ . Then, we say that the function  $f(\cdot)$  is

- (i) proper, if  $f(x) > -\infty$  for all  $x \in \mathbb{R}^n$  and  $\text{dom}(f) \neq \emptyset$ ;
- (ii) convex, if  $f(tx + (1-t)y) \leq tf(x) + (1-t)f(y)$  for any  $x, y \in \text{dom}(f)$  and  $t \in [0, 1]$ ;
- (iii)  $\varrho$ -strongly convex with a given  $\varrho > 0$ , if  $\text{dom}(f)$  is convex and the following inequality holds for any  $x, y \in \text{dom}(f)$  and  $t \in [0, 1]$ :

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y) - \frac{\varrho}{2}t(1-t)\|x - y\|^2.$$

Let  $f(\cdot) : \mathbb{R}^n \rightarrow (-\infty, +\infty]$  be a proper, closed and convex function, then the subdifferential of  $f(\cdot)$  at  $x \in \text{dom}(f)$  is given by

$$\partial f(x) = \{\xi \mid f(z) \geq f(x) + \langle z - x, \xi \rangle, \forall z \in \text{dom}(f)\}.$$

In what follows, we denote  $\text{dom}(\partial f) := \{x \in \mathbb{R}^n \mid \partial f(x) \neq \emptyset\}$ . Then, the following first-order characterizations of strong convexity are frequently used for analysis (e.g. see [3, Theorem 5.24]).

**Lemma 2.1.** Let  $f(\cdot) : \mathbb{R}^n \rightarrow (-\infty, \infty]$  be a proper closed and convex function. Then, for a given  $\varrho > 0$ , the following three claims are equivalent:

- (i)  $f(\cdot)$  is  $\varrho$ -strongly convex.
- (ii)  $f(y) \geq f(x) + \langle \xi, y - x \rangle + \varrho\|y - x\|^2/2$  for any  $x \in \text{dom}(\partial f)$ ,  $y \in \text{dom}(f)$ , and  $\xi \in \partial f(x)$ .
- (iii)  $\langle \xi - \eta, x - y \rangle \geq \varrho\|x - y\|^2$  for any  $x, y \in \text{dom}(\partial f)$  and  $\xi \in \partial f(x)$ ,  $\eta \in \partial f(y)$ .

The proximal operator of  $f$  (see [48, 50]), denoted by  $\text{Prox}_f(\cdot)$ , is given by

$$\text{Prox}_f(a) = \arg \min_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{1}{2} \|x - a\|^2 \right\}, \quad a \in \mathbb{R}^n.$$

Particularly, if  $f(\cdot)$  is the indicator function  $\delta_{\mathcal{X}}(\cdot)$  associated with the nonempty convex set  $\mathcal{X}$ , i.e.

$$\delta_{\mathcal{X}}(x) = \begin{cases} 0, & \text{if } x \in \mathcal{X}, \\ +\infty, & \text{otherwise,} \end{cases}$$

then the proximal operator  $\text{Prox}_f(\cdot)$  immediately reduces to the projection operator, i.e.  $\text{Prox}_f(\cdot) \equiv \Pi_{\mathcal{X}}(\cdot)$ . Let  $\Omega$  be a nonempty closed convex set of  $\mathbb{R}^n$ , we define

$$\text{dist}_M(x, \Omega) := \min \{ \|x - z\|_M \mid z \in \Omega \}$$

as the distance from any  $x \in \mathbb{R}^n$  to the set  $\Omega$  in the sense of matrix norm, where  $M$  is a given symmetric and positive definite matrix. In particular, when  $M$  is an identity matrix, we use  $\text{dist}(x, \Omega)$  to denote the Euclidean distance from any  $x$  to the set  $\Omega$  for simplicity.

Given a proper closed strictly convex function  $\phi(\cdot) : \mathbb{R}^n \rightarrow (-\infty, +\infty]$ , finite at  $x, y$  and differentiable at  $y$ , the Bregman distance [7] between  $x$  and  $y$  associated with the kernel function  $\phi$  is defined as

$$\mathcal{B}_{\phi}(x, y) = \phi(x) - \phi(y) - \langle \nabla \phi(y), x - y \rangle,$$

where  $\nabla \phi(y)$  represents the gradient of  $\phi$  at point  $y$ . It is not difficult to see that the Bregman distance covers the standard Euclidean distance as its special case when  $\phi(\cdot) = \|\cdot\|^2/2$ . Here, we summarize three widely used Bregman distances in Table 2.1. However, the Bregman distance does not always share the symmetry and the triangle inequality property with the Euclidean distance.

Below, we summarize some properties of Bregman distance [3].

**Lemma 2.2.** *Suppose that  $\Omega \subseteq \mathbb{R}^n$  is nonempty closed and convex, and the function  $\phi(\cdot)$  is proper closed convex and differentiable over  $\text{dom}(\partial\phi)$ . If  $\Omega \subseteq \text{dom}(\phi)$  and  $\phi(\cdot) + \delta_{\Omega}(\cdot)$  is  $\varrho$ -strongly convex ( $\varrho > 0$ ), then the Bregman distance  $\mathcal{B}_{\phi}(\cdot, \cdot)$  associated with  $\phi(\cdot)$  has the following properties:*

- (i)  $\mathcal{B}_{\phi}(x, y) \geq \varrho \|x - y\|^2/2$  for all  $x \in \Omega$  and  $y \in \Omega \cap \text{dom}(\partial\phi)$ .
- (ii) Let  $x \in \Omega$  and  $y \in \Omega \cap \text{dom}(\partial\phi)$ . Then  $\mathcal{B}_{\phi}(x, y) \geq 0$ , and in particular, the equality holds if and only if  $x = y$ .
- (iii) For  $a, b \in \text{dom}(\partial\phi)$  and  $c \in \text{dom}(\phi)$ , the following equality holds:

$$\langle \nabla \phi(b) - \nabla \phi(a), c - a \rangle = \mathcal{B}_{\phi}(c, a) + \mathcal{B}_{\phi}(a, b) - \mathcal{B}_{\phi}(c, b).$$

Table 2.1: Three popular Bregman distances.

Type	Kernel function $\phi(\cdot)$	Bregman distance $\mathcal{B}_{\phi}(x, y)$
I	$\ z\ ^2/2$	$\ x - y\ ^2/2, \quad \forall x, y \in \mathbb{R}^n$
II	$\ z\ _M^2/2$ with $M \succ 0$	$\ x - y\ _M^2/2, \quad \forall x, y \in \mathbb{R}^n$
III	$\sum_{i=1}^n z_i \log z_i$	$\sum_{i=1}^n x_i \log(x_i/y_i) + y_i - x_i, \quad \forall x \in \mathbb{R}_+^n, \forall y \in \mathbb{R}_{++}^n$

Below, we present the first-order optimality condition of (1.1). The pair  $(x^*, y^*)$  defined on  $\mathcal{X} \times \mathcal{Y}$  is called a saddle point of (1.1) if it satisfies the following inequalities:

$$\mathcal{L}(x^*, y) \leq \mathcal{L}(x^*, y^*) \leq \mathcal{L}(x, y^*), \quad \forall x \in \mathcal{X}, \quad \forall y \in \mathcal{Y},$$

which can be further reformulated as a mixed variational inequality

$$\begin{cases} f(x) - f(x^*) + \langle x - x^*, A^\top y^* \rangle \geq 0, & \forall x \in \mathcal{X}, \\ g(y) - g(y^*) + \langle y - y^*, -Ax^* \rangle \geq 0, & \forall y \in \mathcal{Y}, \end{cases} \quad (2.1)$$

or equivalently,

$$\Upsilon(\mathbf{u}) - \Upsilon(\mathbf{u}^*) + \langle \mathbf{u} - \mathbf{u}^*, G\mathbf{u}^* \rangle \geq 0, \quad \forall \mathbf{u} \in \mathcal{U}, \quad (2.2a)$$

where

$$\mathbf{u} = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \Upsilon(\mathbf{u}) = f(x) + g(y), \quad G = \begin{pmatrix} 0 & A^\top \\ -A & 0 \end{pmatrix}, \quad \mathcal{U} = \mathcal{X} \times \mathcal{Y}. \quad (2.2b)$$

Alternatively, it is well-known (see [61]) that solving (2.2) amounts to finding a solution of a generalized projection equation, which is shown by the following lemma.

**Lemma 2.3.** *The variational inequality problem (2.2) amounts to finding  $\mathbf{u}^* := (x^*, y^*)$  such that  $0 \in \mathcal{E}(\mathbf{u}^*, t)$ , i.e.*

$$\text{dist}^2(0, \mathcal{E}(\mathbf{u}^*, t)) = 0,$$

where the set-valued mapping  $\mathcal{E}(\mathbf{u}, t)$  is defined as

$$\mathcal{E}(\mathbf{u}, t) := \begin{pmatrix} \mathcal{E}_\mathcal{X}(\mathbf{u}, t) := x - \Pi_\mathcal{X}[x - t(\partial f(x) + A^\top y)] \\ \mathcal{E}_\mathcal{Y}(\mathbf{u}, t) := y - \Pi_\mathcal{Y}[y - t(\partial g(y) - Ax)] \end{pmatrix} \quad (2.3)$$

with  $t > 0$  being an arbitrary scalar.

Throughout this paper, we let  $\mathcal{U}^*$  be the solution set of (2.2), which is assumed to be nonempty. Clearly, it follows from Lemma 2.3 that

$$\mathcal{U}^* = \{\mathbf{u}^* \mid \text{dist}(0, \mathcal{E}(\mathbf{u}^*, t)) = 0\}.$$

Notice that our convergence rate analysis under the error bound condition is based on the variational inequality characterization (2.1) and the related theory of variational inequalities.

### 3. Algorithm and Convergence Properties

In this section, we first present the algorithmic framework for (1.1) and show that our algorithm has a nice convergence behavior through a toy example. Then, we prove that our algorithm is globally convergent and has a linear convergence rate under some standard conditions (see (1.4) or more details in Remark 3.5) used in [10].

#### 3.1. Algorithmic framework

Considering the possibly unbalanced complexity of primal and dual subproblems in some cases, we are motivated to update the primal variable between the twice calculations of the dual variable so that updating the dual variable in a symmetric way. The extra calculation of the dual variable accordingly balances the speed of both variables approaching to their optimal solutions, which can also be regarded as one stabilization on the dual variable. Our algorithmic framework is described formally in Algorithm 3.1.

**Algorithm 3.1:** The Symmetric Primal-dual Algorithmic Framework for (1.1).

1 Choose starting points  $x^0 \in \mathcal{X}, y^0 \in \mathcal{Y}$  and parameters  $\gamma > 0$  and  $\mu > 0$ .

2 **repeat**

3     Update  $x$  and  $y$  via

$$\tilde{y}^{k+1} = \arg \max_{y \in \mathcal{Y}} \{-g(y) + \langle Ax^k, y \rangle - \gamma \mathcal{B}_\phi(y, y^k)\}, \quad (3.1)$$

$$x^{k+1} = \arg \min_{x \in \mathcal{X}} \{f(x) + \langle Ax, \tilde{y}^{k+1} \rangle + \mu \mathcal{B}_\psi(x, x^k)\}, \quad (3.2)$$

$$y^{k+1} = \arg \max_{y \in \mathcal{Y}} \{-g(y) + \langle Ax^{k+1}, y \rangle - \gamma \mathcal{B}_\phi(y, y^k)\}. \quad (3.3)$$

4 **until** Some stopping criterion is satisfied.

5 **return** An approximate saddle point  $(\hat{x}, \hat{y})$ .

**Remark 3.1.** It is noteworthy that the embedded Bregman proximal regularization terms (i.e.  $\mathcal{B}_\psi(x, x^k)$  and  $\mathcal{B}_\phi(y, y^k)$ ) make our Algorithm 3.1 versatile in the sense that we can choose appropriate Bregman kernel functions as listed in Table 2.1 to interpret some state-of-the-art first-order optimization solvers (see Section 4), or design customized variants for some real-world problems (see Section 5).

**Remark 3.2.** We employ a toy example used in [29] to show that our Algorithm 3.1 enjoys a nice convergence behavior. Consider the following linear programming:

$$\min_{x_1, x_2} \{2x_1 + x_2 \mid x_1 + x_2 = 1, x_1 \geq 0, x_2 \geq 0\}, \quad (3.4)$$

which has a unique solution  $(x_1^*, x_2^*) = (0, 1)$ . Moreover, the dual problem of (3.4) is

$$\max_y \{y \mid y \leq 1, y \leq 2\}$$

and its optimal solution is  $y^* = 1$ . Accordingly, we reformulate (3.4) as a standard form of saddle point problems, i.e.

$$\min_{x_1 \geq 0, x_2 \geq 0} \max_y \{2x_1 + x_2 - y(x_1 + x_2 - 1)\}. \quad (3.5)$$

Applying Algorithm 3.1 to (3.5) by setting the Bregman kernel functions as the first type listed in Table 2.1, the iterative scheme is immediately specified as

$$\begin{cases} \tilde{y}^{k+1} = y^k - \frac{1}{\gamma}(x_1^k + x_2^k - 1), \\ x_1^{k+1} = \max \left\{ \left( -\frac{2}{\mu} + \frac{1}{\mu} \tilde{y}^{k+1} + x_1^k \right), 0 \right\}, \\ x_2^{k+1} = \max \left\{ \left( -\frac{1}{\mu} + \frac{1}{\mu} \tilde{y}^{k+1} + x_2^k \right), 0 \right\}, \\ y^{k+1} = y^k - \frac{1}{\gamma}(x_1^{k+1} + x_2^{k+1} - 1). \end{cases}$$

Also, we implement the AHPD method (1.2) and the PDHG (1.3) with  $\tau = 1$  to solve (3.5). Here, we take  $(x_1^0, x_2^0, y^0) = (0, 0, 0)$  as starting points and plot trajectories of the sequences

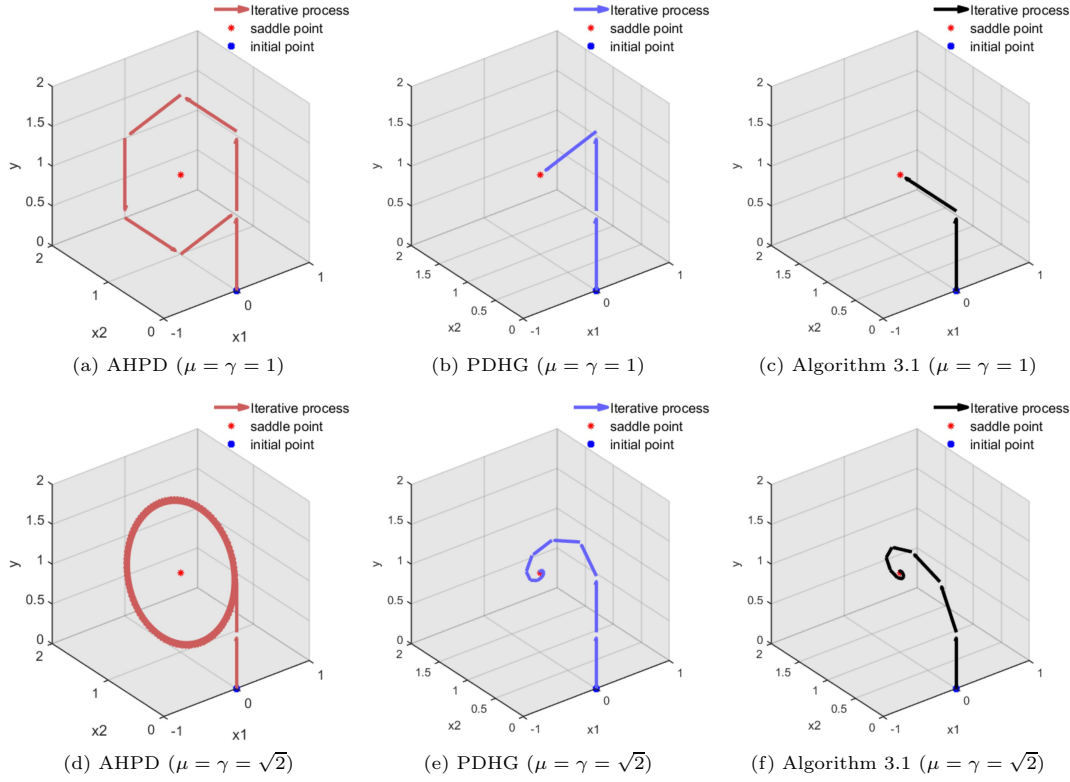


Fig. 3.1. Illustration of the convergence behaviors of AHPD, PDHG and Algorithm 3.1 for the toy example (3.4) with setting different parameters  $\mu$  and  $\gamma$ .

generated by the three algorithms in Fig. 3.1 for the cases where  $\mu = \gamma = 1$  or  $\mu = \gamma = \sqrt{2}$ , respectively. It can be easily seen from Fig. 3.1 that the sequence generated by our Algorithm 3.1 converges slightly faster than PDHG to the unique optimal solution, while the AHPD method generates a cyclic sequence. Such a toy example tells us that our Algorithm 3.1 possibly has superiority over some existing state-of-the-art primal-dual algorithms on some saddle point problems.

**Remark 3.3.** When assuming that both  $f(x)$  and  $g(y)$  are differentiable, the saddle point problem (1.1) can be also reformulated as a standard variational inequality problem (VIP), i.e. finding a point  $\mathbf{u}^* \in \mathcal{U}$  such that

$$\langle \mathbf{u} - \mathbf{u}^*, \nabla \mathcal{L}(\mathbf{u}^*) \rangle \geq 0, \quad \forall \mathbf{u} \in \mathcal{U},$$

where  $\mathbf{u}$  and  $\mathcal{U}$  are given in (2.2b), and  $\nabla \mathcal{L}(\mathbf{u})$  is specified as

$$\nabla \mathcal{L}(\mathbf{u}) = \begin{pmatrix} \nabla_x \mathcal{L}(x, y) \\ -\nabla_y \mathcal{L}(x, y) \end{pmatrix} = \begin{pmatrix} \nabla f(x) + A^\top y \\ \nabla g(y) - Ax \end{pmatrix}.$$

Consequently, we can gainfully employ some efficient algorithms, such as the projection methods and extragradient methods (e.g. [40, 49, 51, 62]) tailored for VIPs to deal with those saddle point problems with differentiable objectives. However, many real-world problems do not necessarily

satisfy the Lipschitz continuity of  $\nabla \mathcal{L}(\mathbf{u})$  required in convergence analysis. Besides, the aforementioned VIP-type methods such as the well-known extragradient methods (e.g. [40, 49, 51, 62]) treat both  $x$  and  $y$  as a single entity, thereby potentially ignoring the distinct properties associated with each individual variable. As a consequence, when implementing the aforementioned extragradient methods, we must update  $x$  and  $y$  simultaneously, thereby reducing the implementability of extragradient methods for (1.1). Comparatively, our Algorithm 3.1 does not require the Lipschitz continuity of  $\nabla \mathcal{L}(\mathbf{u})$ . More promisingly, our method is able to maximally exploit the structure of (1.1) so that each subproblem is easily implemented for many real-world problems (see Section 5).

**Remark 3.4.** After we finished this manuscript, we were brought to the so-named alternating extragradient method (AEM) introduced by Bonettini and Ruggiero [4, 6] and the primal-dual fixed point algorithm (PDFP) proposed by Chen *et al.* [15]. Firstly, under the same differentiability requirement on  $\mathcal{L}(x, y)$ , the AEM is an improved variant of the classical extragradient methods [40, 49, 51], where the respective structure associated with  $x$  and  $y$  can be efficiently explored. Moreover, the AEM enjoys an effective adaptive stepsize for primal and dual subproblems. However, the Lipschitz continuity assumption for the AEM would possibly preclude its applicability to some nonsmooth real-world problems. Comparatively, although our Algorithm 3.1 shares the similar symmetric spirit to update the primal and dual variables with the AEM, our Algorithm 3.1 is easily applicable to nonsmooth saddle point problems with a bilinear coupling term. Moreover, as a theoretical complement, we not only prove the global convergence of Algorithm 3.1, but also estimate the linear convergence rate under the generalized error bound defined by Bregman distance. Secondly, when comparing with the PDFP in [15], we can see that PDFP shares a highly similar iterative scheme with our SPIDA by computing the  $y$ -subproblem twice at each iteration. In particular, when the proximal term in the  $x$ -subproblem (3.2) of SPIDA is chosen as the Euclidean norm, i.e.

$$\mathcal{B}_\psi(x, x^k) = \frac{1}{2} \|x - x^k\|^2,$$

the iterative form for the  $x$ -subproblem coincides with the one of PDFP except the appearance of proximal parameter  $\mu$ . On the other hand, our SPIDA has versatile Bregman proximal terms making it flexible for many structured optimization problems. From this point, our SPIDA is more general than the PDFP.

### 3.2. Global convergence

In this subsection, we will prove the global convergence of Algorithm 3.1 under some standard conditions used in the primal-dual literature.

Note that both subproblems (3.1) and (3.3) share a similar form, so we denote

$$\mathbf{y}_w = \arg \max_{y \in \mathcal{Y}} \{ -g(y) + \langle Aw, y \rangle - \gamma \mathcal{B}_\phi(y, y^k) \}$$

with some given  $w \in \mathbb{R}^m$  for convenience, which can also be rewritten as

$$\mathbf{y}_w = \arg \min_{y \in \mathcal{Y}} \{ g(y) - \langle Aw, y \rangle + \gamma \phi(y) - \gamma \langle \nabla \phi(y^k), y - y^k \rangle \}. \quad (3.6)$$

Hereafter, we begin our analysis with the following lemma.

**Lemma 3.1.** *Suppose that the Bregman kernel function  $\phi(\cdot)$  is  $\varrho$ -strongly convex. Let  $\mathbf{y}_u$  and  $\mathbf{y}_w$  be solutions of (3.6) for some given  $u$  and  $w$ , respectively. Then, the following inequality:*

$$\begin{aligned} & g(\mathbf{y}_u) - g(y) + \langle \mathbf{y}_u - y, -Aw \rangle \\ & \leq \gamma \mathcal{B}_\phi(y, y^k) - \gamma \mathcal{B}_\phi(y, \mathbf{y}_w) - \gamma \mathcal{B}_\phi(\mathbf{y}_u, y^k) \\ & \quad + \frac{\alpha}{2} \|A^\top A\| \|w - u\|^2 - \left( \gamma - \frac{1}{\alpha \varrho} \right) \mathcal{B}_\phi(\mathbf{y}_w, \mathbf{y}_u) \end{aligned} \quad (3.7)$$

holds for all  $y \in \mathcal{Y}$ , where  $\alpha$  is a positive constant.

*Proof.* Note that  $\mathbf{y}_u$  is a minimizer of (3.6) for given  $u \in \mathbb{R}^m$ . It then follows from the first-order optimality condition of (3.6) that

$$g(\mathbf{y}_u) - g(y) \leq \langle y - \mathbf{y}_u, -Au + \gamma \nabla \phi(\mathbf{y}_u) - \gamma \nabla \phi(y^k) \rangle, \quad \forall y \in \mathcal{Y}. \quad (3.8)$$

Consequently, by using the arbitrariness of  $y$  in (3.8) with setting  $y = \mathbf{y}_w$ , we have

$$g(\mathbf{y}_u) - g(\mathbf{y}_w) \leq \langle \mathbf{y}_w - \mathbf{y}_u, -Au + \gamma \nabla \phi(\mathbf{y}_u) - \gamma \nabla \phi(y^k) \rangle. \quad (3.9)$$

Similarly, by the definition of  $\mathbf{y}_w$ , it then follows from the first-order optimality condition of (3.6) that

$$g(\mathbf{y}_w) - g(y) \leq \langle y - \mathbf{y}_w, -Aw + \gamma \nabla \phi(\mathbf{y}_w) - \gamma \nabla \phi(y^k) \rangle, \quad \forall y \in \mathcal{Y}. \quad (3.10)$$

Setting  $y = \mathbf{y}_u$  in (3.10) arrives at

$$g(\mathbf{y}_w) - g(\mathbf{y}_u) \leq \langle \mathbf{y}_u - \mathbf{y}_w, -Aw + \gamma \nabla \phi(\mathbf{y}_w) - \gamma \nabla \phi(y^k) \rangle. \quad (3.11)$$

By invoking the definition of Bregman distance, we have

$$\begin{aligned} & \gamma \mathcal{B}_\phi(y, y^k) - \gamma \mathcal{B}_\phi(y, \mathbf{y}_w) \\ & = \gamma \phi(\mathbf{y}_w) - \gamma \phi(y^k) + \gamma \langle \nabla \phi(\mathbf{y}_w), y - \mathbf{y}_w \rangle - \gamma \langle \nabla \phi(y^k), y - y^k \rangle \\ & = \gamma \phi(\mathbf{y}_w) - \gamma \phi(y^k) + \langle Aw, y - \mathbf{y}_w \rangle + \gamma \langle \nabla \phi(y^k), y^k - \mathbf{y}_w \rangle \\ & \quad + \langle \gamma \nabla \phi(\mathbf{y}_w) - \gamma \nabla \phi(y^k) - Aw, y - \mathbf{y}_w \rangle. \end{aligned} \quad (3.12)$$

Using inequality (3.10) instead of the last term of (3.12) leads to

$$\begin{aligned} & \gamma \mathcal{B}_\phi(y, y^k) - \gamma \mathcal{B}_\phi(y, \mathbf{y}_w) \\ & \geq \gamma \phi(\mathbf{y}_w) - \gamma \phi(y^k) + \langle Aw, y - \mathbf{y}_w \rangle + \gamma \langle \nabla \phi(y^k), y^k - \mathbf{y}_w \rangle + g(\mathbf{y}_w) - g(y) \\ & = \gamma \phi(\mathbf{y}_w) - \gamma \phi(y^k) + \langle Aw, \mathbf{y}_u - \mathbf{y}_w \rangle + \gamma \langle \nabla \phi(y^k), y^k - \mathbf{y}_w \rangle \\ & \quad - \langle Aw, \mathbf{y}_u - y \rangle + g(\mathbf{y}_w) - g(y) \\ & = \Phi(y^k, \mathbf{y}_u, \mathbf{y}_w) - \langle Aw, \mathbf{y}_u - y \rangle + g(\mathbf{y}_w) - g(y), \end{aligned} \quad (3.13)$$

where

$$\Phi(y^k, \mathbf{y}_u, \mathbf{y}_w) := \gamma \phi(\mathbf{y}_w) - \gamma \phi(y^k) + \langle Aw, \mathbf{y}_u - \mathbf{y}_w \rangle + \gamma \langle \nabla \phi(y^k), y^k - \mathbf{y}_w \rangle. \quad (3.14)$$

Below, we focus on  $\Phi(y^k, \mathbf{y}_u, \mathbf{y}_w)$ . First, we have

$$\begin{aligned} \langle Aw, \mathbf{y}_u - \mathbf{y}_w \rangle & = \langle \mathbf{y}_u - \mathbf{y}_w, Aw - Au \rangle + \langle \mathbf{y}_u - \mathbf{y}_w, Au \rangle \\ & = \langle \mathbf{y}_u - \mathbf{y}_w, Aw - Au \rangle + \langle \mathbf{y}_w - \mathbf{y}_u, \gamma \nabla \phi(y^k) - \gamma \nabla \phi(\mathbf{y}_u) \rangle \\ & \quad + \langle \mathbf{y}_w - \mathbf{y}_u, -Au + \gamma \nabla \phi(\mathbf{y}_u) - \gamma \nabla \phi(y^k) \rangle \\ & \geq \langle \mathbf{y}_u - \mathbf{y}_w, Aw - Au \rangle + \langle \mathbf{y}_w - \mathbf{y}_u, \gamma \nabla \phi(y^k) - \gamma \nabla \phi(\mathbf{y}_u) \rangle \\ & \quad + g(\mathbf{y}_u) - g(\mathbf{y}_w), \end{aligned} \quad (3.15)$$

where the last inequality follows from (3.9). As a result, substituting (3.15) into (3.14) immediately arrives at

$$\begin{aligned}
\Phi(y^k, \mathbf{y}_u, \mathbf{y}_w) &= \gamma\phi(\mathbf{y}_w) - \gamma\phi(y^k) + \gamma\langle \nabla\phi(y^k), y^k - \mathbf{y}_w \rangle + \langle Aw, \mathbf{y}_u - \mathbf{y}_w \rangle \\
&\geq \gamma\phi(\mathbf{y}_w) - \gamma\phi(y^k) + \gamma\langle \nabla\phi(y^k), y^k - \mathbf{y}_w \rangle + \langle \mathbf{y}_u - \mathbf{y}_w, Aw - Au \rangle \\
&\quad + \langle \mathbf{y}_w - \mathbf{y}_u, \gamma\nabla\phi(y^k) - \gamma\nabla\phi(\mathbf{y}_u) \rangle + g(\mathbf{y}_u) - g(\mathbf{y}_w) \\
&= \gamma\phi(\mathbf{y}_w) - \gamma\phi(y^k) + \langle \mathbf{y}_u - \mathbf{y}_w, Aw - Au \rangle - \langle \gamma\nabla\phi(y^k), \mathbf{y}_u - y^k \rangle \\
&\quad - \langle \gamma\nabla\phi(\mathbf{y}_u), \mathbf{y}_w - \mathbf{y}_u \rangle + g(\mathbf{y}_u) - g(\mathbf{y}_w) \\
&= \gamma\mathcal{B}_\phi(\mathbf{y}_w, \mathbf{y}_u) + \gamma\mathcal{B}_\phi(\mathbf{y}_u, y^k) \\
&\quad + \langle \mathbf{y}_u - \mathbf{y}_w, Aw - Au \rangle + g(\mathbf{y}_u) - g(\mathbf{y}_w),
\end{aligned} \tag{3.16}$$

where the last equality follows from the definition of Bregman distance. Since the Bregman kernel function  $\phi(\cdot)$  is  $\varrho$ -strongly convex, an application of the fact

$$\langle a, b \rangle \geq -\frac{1}{2\alpha}\|a\|^2 - \frac{\alpha}{2}\|b\|^2, \quad \forall a, b \in \mathbb{R}^n, \quad \alpha > 0$$

immediately yields

$$\begin{aligned}
&\langle \mathbf{y}_u - \mathbf{y}_w, Aw - Au \rangle \\
&\geq -\frac{1}{2\alpha}\|\mathbf{y}_w - \mathbf{y}_u\|^2 - \frac{\alpha}{2}\|A^\top A\|\|w - u\|^2 \\
&\geq -\frac{1}{\alpha\varrho}\mathcal{B}_\phi(\mathbf{y}_w, \mathbf{y}_u) - \frac{\alpha}{2}\|A^\top A\|\|w - u\|^2.
\end{aligned} \tag{3.17}$$

Plugging (3.17) into (3.16), we have

$$\begin{aligned}
\Phi(y^k, \mathbf{y}_u, \mathbf{y}_w) &\geq \left(\gamma - \frac{1}{\alpha\varrho}\right)\mathcal{B}_\phi(\mathbf{y}_w, \mathbf{y}_u) + \gamma\mathcal{B}_\phi(\mathbf{y}_u, y^k) \\
&\quad - \frac{\alpha}{2}\|A^\top A\|\|w - u\|^2 + g(\mathbf{y}_u) - g(\mathbf{y}_w),
\end{aligned}$$

which, together with (3.13), implies that

$$\begin{aligned}
&\gamma\mathcal{B}_\phi(y, y^k) - \gamma\mathcal{B}_\phi(y, \mathbf{y}_w) \\
&\geq \Phi(y^k, \mathbf{y}_u, \mathbf{y}_w) - \langle Aw, \mathbf{y}_u - y \rangle + g(\mathbf{y}_w) - g(y) \\
&\geq \left(\gamma - \frac{1}{\alpha\varrho}\right)\mathcal{B}_\phi(\mathbf{y}_w, \mathbf{y}_u) + \gamma\mathcal{B}_\phi(\mathbf{y}_u, y^k) \\
&\quad - \frac{\alpha}{2}\|A^\top A\|\|w - u\|^2 + g(\mathbf{y}_u) - \langle Aw, \mathbf{y}_u - y \rangle - g(y).
\end{aligned}$$

Rearranging terms of the above inequality completes the proof.  $\square$

**Lemma 3.2.** *Let  $\{(x^{k+1}, \tilde{y}^{k+1}, y^{k+1})\}$  be a sequence generated by Algorithm 3.1. Then, for all  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$ , we have*

$$\begin{aligned}
&\mathcal{L}(x^{k+1}, y) - \mathcal{L}(x^{k+1}, \tilde{y}^{k+1}) \\
&\leq \gamma\mathcal{B}_\phi(y, y^k) - \gamma\mathcal{B}_\phi(y, y^{k+1}) - \gamma\mathcal{B}_\phi(\tilde{y}^{k+1}, y^k) \\
&\quad + \frac{\alpha}{2}\|A^\top A\|\|x^{k+1} - x^k\|^2 - \left(\gamma - \frac{1}{\alpha\varrho}\right)\mathcal{B}_\phi(y^{k+1}, \tilde{y}^{k+1}),
\end{aligned} \tag{3.18}$$

$$\begin{aligned}
&\mathcal{L}(x^{k+1}, \tilde{y}^{k+1}) - \mathcal{L}(x, \tilde{y}^{k+1}) \\
&\leq \mu\mathcal{B}_\psi(x, x^k) - \mu\mathcal{B}_\psi(x, x^{k+1}) - \mu\mathcal{B}_\psi(x^{k+1}, x^k).
\end{aligned} \tag{3.19}$$

*Proof.* First, it is clear from the notation in (1.1) that

$$\begin{aligned} & \mathcal{L}(x^{k+1}, y) - \mathcal{L}(x^{k+1}, \tilde{y}^{k+1}) \\ &= g(\tilde{y}^{k+1}) - g(y) + \langle \tilde{y}^{k+1} - y, -Ax^{k+1} \rangle. \end{aligned} \quad (3.20)$$

Since  $\tilde{y}^{k+1}$  and  $y^{k+1}$  are solutions of (3.1) and (3.3), respectively, it immediately follows from (3.20) and Lemma 3.1 with settings  $\mathbf{y}_u = \tilde{y}^{k+1}$ ,  $\mathbf{y}_w = y^{k+1}$ ,  $u = x^k$ ,  $w = x^{k+1}$  that

$$\begin{aligned} & \mathcal{L}(x^{k+1}, y) - \mathcal{L}(x^{k+1}, \tilde{y}^{k+1}) \\ &= g(\tilde{y}^{k+1}) - g(y) + \langle \tilde{y}^{k+1} - y, -Ax^{k+1} \rangle \\ &\leq \gamma \mathcal{B}_\phi(y, y^k) - \gamma \mathcal{B}_\phi(y, y^{k+1}) - \gamma \mathcal{B}_\phi(\tilde{y}^{k+1}, y^k) \\ &\quad + \frac{\alpha}{2} \|A^\top A\| \|x^{k+1} - x^k\|^2 - \left( \gamma - \frac{1}{\alpha\varrho} \right) \mathcal{B}_\phi(y^{k+1}, \tilde{y}^{k+1}), \end{aligned} \quad (3.21)$$

which is precisely the same as (3.18). We proved the first assertion of this lemma.

Below, we show (3.19). It follows from the notation of  $\mathcal{L}(x, y)$  in (1.1) that

$$\begin{aligned} & \mathcal{L}(x, \tilde{y}^{k+1}) - \mathcal{L}(x^{k+1}, \tilde{y}^{k+1}) \\ &= f(x) - f(x^{k+1}) + \langle Ax - Ax^{k+1}, \tilde{y}^{k+1} \rangle. \end{aligned} \quad (3.22)$$

On the other hand, the first-order optimality condition of (3.2) is

$$f(x) - f(x^{k+1}) + \langle x - x^{k+1}, A^\top \tilde{y}^{k+1} + \mu \nabla \psi(x^{k+1}) - \mu \nabla \psi(x^k) \rangle \geq 0, \quad \forall x \in \mathcal{X}. \quad (3.23)$$

Consequently, combining (3.22) and (3.23) leads to

$$\begin{aligned} & \mathcal{L}(x^{k+1}, \tilde{y}^{k+1}) - \mathcal{L}(x, \tilde{y}^{k+1}) \\ &= -f(x) + f(x^{k+1}) - \langle Ax - Ax^{k+1}, \tilde{y}^{k+1} \rangle \\ &\leq \langle x - x^{k+1}, \mu \nabla \psi(x^{k+1}) - \mu \nabla \psi(x^k) \rangle \\ &= -\mu \langle x - x^{k+1}, \nabla \psi(x^k) - \nabla \psi(x^{k+1}) \rangle. \end{aligned} \quad (3.24)$$

Applying the three-point property of Bregman distance (2.2) to (3.24) with setting  $a = x^{k+1}$ ,  $b = x^k$ , and  $c = x$ , we conclude that

$$\begin{aligned} & \mathcal{L}(x^{k+1}, \tilde{y}^{k+1}) - \mathcal{L}(x, \tilde{y}^{k+1}) \\ &\leq -\mu \langle x - x^{k+1}, \nabla \psi(x^k) - \nabla \psi(x^{k+1}) \rangle \\ &= \mu \mathcal{B}_\psi(x, x^k) - \mu \mathcal{B}_\psi(x, x^{k+1}) - \mu \mathcal{B}_\psi(x^{k+1}, x^k). \end{aligned}$$

This completes the proof of this lemma.  $\square$

Before presenting the global convergence theorem, we first make the following assumption.

**Assumption 3.1.** The Bregman kernel function  $\phi(\cdot)$  associated with the  $y$ -subproblem is  $\varrho$ -strongly convex such that the proximal parameter  $\gamma$  and constant  $\alpha$  satisfying  $\alpha\varrho\gamma \geq 1$ . Moreover, the Bregman kernel function  $\psi(\cdot)$  associated with the  $x$ -subproblem is  $\kappa$ -strongly convex such that  $\kappa\mu > \alpha\|A^\top A\|$ .

**Remark 3.5.** Note that the condition assumed in Assumption 3.1 is closely related to the standard requirement in [10]. By invoking the arbitrariness of  $\alpha$  in the derivation of (3.17), both conditions  $\alpha\varrho\gamma \geq 1$  and  $\kappa\mu > \alpha\|A^\top A\|$  in Assumption 3.1 imply that  $\mu\gamma\kappa\varrho > \|A^\top A\|$ . In this situation, when  $\kappa = \varrho = 1$ , such a condition reduces to the standard requirement of the PDHG [10].

With the above preparations, we now state our global convergence theorem.

**Theorem 3.1.** *Suppose that Assumption 3.1 holds. Let  $\{(x^{k+1}, \tilde{y}^{k+1}, y^{k+1})\}$  be a sequence generated by Algorithm 3.1. Then, for all  $x \in \mathcal{X}$ ,  $y \in \mathcal{Y}$ , we have*

$$\mathcal{L}(\hat{x}^N, y) - \mathcal{L}(x, \hat{y}^N) \leq \frac{1}{N} (\gamma \mathcal{B}_\phi(y, y^0) + \mu \mathcal{B}_\psi(x, x^0)),$$

where  $N$  is a positive integer,

$$\hat{x}^N = \frac{1}{N} \sum_{k=0}^{N-1} x^k, \quad \hat{y}^N = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{y}^k.$$

*Proof.* Under Assumption 3.1, it is easy to check by Lemma 2.2 that there exists a constant

$$\hat{\beta} := \frac{\kappa\mu - \alpha\|A^\top A\|}{2\mu} > 0$$

such that

$$\mathcal{B}_\psi(x^{k+1}, x^k) - \frac{\alpha}{2\mu} \|A^\top A\| \|x^{k+1} - x^k\|^2 \geq \hat{\beta} \|x^{k+1} - x^k\|^2 > 0. \quad (3.25)$$

Consequently, by adding (3.18) and (3.19), it follows from (3.25) and the positivity of the Bregman distance ( $\alpha\varrho\gamma \geq 1$ ) that

$$\begin{aligned} & \mathcal{L}(x^{k+1}, y) - \mathcal{L}(x, \tilde{y}^{k+1}) \\ & \leq \gamma \mathcal{B}_\phi(y, y^k) - \gamma \mathcal{B}_\phi(y, y^{k+1}) + \mu \mathcal{B}_\psi(x, x^k) - \mu \mathcal{B}_\psi(x, x^{k+1}) \\ & \quad - \mu \left( \mathcal{B}_\psi(x^{k+1}, x^k) - \frac{\alpha}{2\mu} \|A^\top A\| \|x^{k+1} - x^k\|^2 \right) \\ & \quad - \gamma \mathcal{B}_\phi(\tilde{y}^{k+1}, y^k) - \left( \gamma - \frac{1}{\alpha\varrho} \right) \mathcal{B}_\phi(y^{k+1}, \tilde{y}^{k+1}) \\ & \leq \gamma \mathcal{B}_\phi(y, y^k) - \gamma \mathcal{B}_\phi(y, y^{k+1}) + \mu \mathcal{B}_\psi(x, x^k) - \mu \mathcal{B}_\psi(x, x^{k+1}) \\ & \quad - \hat{\beta} \|x^{k+1} - x^k\|^2 - \gamma \mathcal{B}_\phi(\tilde{y}^{k+1}, y^k) - \left( \gamma - \frac{1}{\alpha\varrho} \right) \mathcal{B}_\phi(y^{k+1}, \tilde{y}^{k+1}) \\ & \leq \gamma \mathcal{B}_\phi(y, y^k) - \gamma \mathcal{B}_\phi(y, y^{k+1}) + \mu \mathcal{B}_\psi(x, x^k) - \mu \mathcal{B}_\psi(x, x^{k+1}). \end{aligned} \quad (3.26)$$

Hence, summing up (3.26) from  $k = 0$  to  $N - 1$  leads to

$$\begin{aligned} & \sum_{k=0}^{N-1} (\mathcal{L}(x^{k+1}, y) - \mathcal{L}(x, \tilde{y}^{k+1})) \\ & \leq \gamma \mathcal{B}_\phi(y, y^0) - \gamma \mathcal{B}_\phi(y, y^N) + \mu \mathcal{B}_\psi(x, x^0) - \mu \mathcal{B}_\psi(x, x^N) \\ & \leq \gamma \mathcal{B}_\phi(y, y^0) + \mu \mathcal{B}_\psi(x, x^0). \end{aligned} \quad (3.27)$$

Notice that  $\mathcal{L}(x, y)$  and  $-\mathcal{L}(x, y)$  are convex with respect to  $x$  and  $y$ , respectively. It then follows from the Jensen inequality and (3.27) that

$$\begin{aligned} & \mathcal{L}(\hat{x}^N, y) - \mathcal{L}(x, \hat{y}^N) \\ & \leq \frac{1}{N} \sum_{k=0}^{N-1} (\mathcal{L}(x^{k+1}, y) - \mathcal{L}(x, \tilde{y}^{k+1})) \\ & \leq \frac{1}{N} (\gamma \mathcal{B}_\phi(y, y^0) + \mu \mathcal{B}_\psi(x, x^0)). \end{aligned}$$

We complete the proof.  $\square$

**Theorem 3.2.** *Suppose that Assumption 3.1 holds. The sequence  $\{(x^{k+1}, y^{k+1})\}$  generated by Algorithm 3.1 converges to a saddle point of  $\mathcal{L}(x, y)$  in (1.1).*

*Proof.* Letting  $(x^*, y^*)$  be a saddle point of (1.1), we immediately have

$$\mathcal{L}(x^*, y) \leq \mathcal{L}(x^*, y^*) \leq \mathcal{L}(x, y^*), \quad \forall x \in \mathcal{X}, \quad \forall y \in \mathcal{Y}. \quad (3.28)$$

On the other hand, by setting  $x = x^{k+1}$  and  $y = \tilde{y}^{k+1}$  in (3.28), it follows from (3.26) with setting  $x = x^*$  and  $y = y^*$  that

$$\begin{aligned} 0 &\leq \mathcal{L}(x^{k+1}, y^*) - \mathcal{L}(x^*, \tilde{y}^{k+1}) \\ &\leq \gamma \mathcal{B}_\phi(y^*, y^k) - \gamma \mathcal{B}_\phi(y^*, y^{k+1}) \\ &\quad + \mu \mathcal{B}_\psi(x^*, x^k) - \mu \mathcal{B}_\psi(x^*, x^{k+1}), \end{aligned}$$

which clearly implies that

$$\begin{aligned} 0 &\leq \gamma \mathcal{B}_\phi(y^*, y^{k+1}) + \mu \mathcal{B}_\psi(x^*, x^{k+1}) \\ &\leq \gamma \mathcal{B}_\phi(y^*, y^k) + \mu \mathcal{B}_\psi(x^*, x^k). \end{aligned} \quad (3.29)$$

Obviously, inequality (3.29) means that the sequence  $\{\gamma \mathcal{B}_\phi(y^*, y^k) + \mu \mathcal{B}_\psi(x^*, x^k)\}$  is Bregman monotone (see [2]) decreasing and bounded. By the strong convexity of the Bregman distance, the sequence  $\{(x^k, y^k)\}$  is bounded. Therefore, there exists a subsequence  $\{(x^{k_j}, y^{k_j})\}$  converging to a cluster point, denoted by  $(x^\infty, y^\infty)$ . Below, we aim to show that such a cluster point is a saddle point of (1.1).

According to (3.26), we also have

$$\begin{aligned} &\gamma \mathcal{B}_\phi(\tilde{y}^{k+1}, y^k) + \left( \gamma - \frac{1}{\alpha \varrho} \right) \mathcal{B}_\phi(y^{k+1}, \tilde{y}^{k+1}) + \hat{\beta} \|x^{k+1} - x^k\|^2 \\ &\leq \gamma \mathcal{B}_\phi(y^*, y^k) - \gamma \mathcal{B}_\phi(y^*, y^{k+1}) + \mu \mathcal{B}_\psi(x^*, x^k) - \mu \mathcal{B}_\psi(x^*, x^{k+1}). \end{aligned} \quad (3.30)$$

Summing the above inequality from  $k = 0$  to  $N$ , since  $\alpha > 0, \gamma > 0, \beta > 0$  and  $\alpha\gamma\varrho \geq 1$  are constant, we have

$$\begin{aligned} &\sum_{k=0}^N \left( \gamma \mathcal{B}_\phi(\tilde{y}^{k+1}, y^k) + \left( \gamma - \frac{1}{\alpha \varrho} \right) \mathcal{B}_\phi(y^{k+1}, \tilde{y}^{k+1}) + \hat{\beta} \|x^{k+1} - x^k\|^2 \right) \\ &\leq \gamma \mathcal{B}_\phi(y^*, y^0) + \mu \mathcal{B}_\psi(x^*, x^0), \end{aligned}$$

which implies that

$$\lim_{k \rightarrow \infty} \mathcal{B}_\phi(\tilde{y}^{k+1}, y^k) = \lim_{k \rightarrow \infty} \mathcal{B}_\phi(y^{k+1}, \tilde{y}^{k+1}) = \lim_{k \rightarrow \infty} \|x^{k+1} - x^k\|^2 = 0. \quad (3.31)$$

Moreover, we conclude from (3.31) that both  $\{y^k\}$  and  $\{\tilde{y}^k\}$  converge to the same point. Therefore,  $(x^\infty, y^\infty)$  is also a cluster point of the subsequence  $\{(x^{k_j}, \tilde{y}^{k_j})\}$ . It then follows from the first-order optimality conditions of (3.2) and (3.3) that, for all  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$ , the following inequalities hold:

$$\begin{cases} f(x) - f(x^{k+1}) + \langle x - x^{k+1}, A^\top \tilde{y}^{k+1} + \mu \nabla \psi(x^{k+1}) - \mu \nabla \psi(x^k) \rangle \geq 0, \\ g(y) - g(y^{k+1}) + \langle y - y^{k+1}, -Ax^{k+1} + \gamma \nabla \phi(y^{k+1}) - \gamma \nabla \phi(y^k) \rangle \geq 0. \end{cases}$$

Therefore, for all  $x \in \mathcal{X}$  and  $y \in \mathcal{Y}$ , the subsequence  $\{(x^{k_j}, \tilde{y}^{k_j}, y^{k_j})\}$  satisfies

$$\begin{cases} f(x) - f(x^{k_j+1}) + \langle x - x^{k_j+1}, A^\top \tilde{y}^{k_j+1} + \mu \nabla \psi(x^{k_j+1}) - \mu \nabla \psi(x^{k_j}) \rangle \geq 0, \\ g(y) - g(y^{k_j+1}) + \langle y - y^{k_j+1}, -Ax^{k_j+1} + \gamma \nabla \phi(y^{k_j+1}) - \gamma \nabla \phi(y^{k_j}) \rangle \geq 0. \end{cases} \quad (3.32)$$

Consequently, taking limit as  $j \rightarrow \infty$  over the subsequence  $k_j$  in (3.32), it follows from (3.31) that

$$\begin{cases} f(x) - f(x^\infty) + \langle x - x^\infty, A^\top y^\infty \rangle \geq 0, & \forall x \in \mathcal{X}, \\ g(y) - g(y^\infty) + \langle y - y^\infty, -Ax^\infty \rangle \geq 0, & \forall y \in \mathcal{Y}, \end{cases}$$

which, together with (2.1), means that the limit point  $(x^\infty, y^\infty)$  is a saddle point of  $\mathcal{L}(x, y)$  in (1.1). So  $(x^*, y^*)$  can be replaced by  $(x^\infty, y^\infty)$  in the sequence  $\{\gamma \mathcal{B}_\phi(y^*, y^k) + \mu \mathcal{B}_\psi(x^*, x^k)\}$ . Thus,  $\{\gamma \mathcal{B}_\phi(y^\infty, y^k) + \mu \mathcal{B}_\psi(x^\infty, x^k)\}$  is monotone decreasing and bounded, we obtain

$$\lim_{k \rightarrow \infty} (\gamma \mathcal{B}_\phi(y^\infty, y^k) + \mu \mathcal{B}_\psi(x^\infty, x^k)) = \lim_{j \rightarrow \infty} (\gamma \mathcal{B}_\phi(y^\infty, y^{k_j}) + \mu \mathcal{B}_\psi(x^\infty, x^{k_j})) = 0,$$

which means that  $x^k \rightarrow x^\infty$  and  $y^k \rightarrow y^\infty$  as  $k \rightarrow \infty$ . We complete the proof.  $\square$

### 3.3. Linear convergence

In this subsection, we turn our attention to establishing the linear convergence in the context of a generalized error bound conditions. First, we make the following assumption, which is an extended version used in the literature, e.g. [37, 38, 58, 61].

**Assumption 3.2.** Assume that, for any  $\omega > 0$ , there exists  $\eta > 0$  such that

$$\text{dist}_{\mathcal{B}}(\mathbf{u}, \mathcal{U}^*) \leq \eta \text{dist}(0, \mathcal{E}(\mathbf{u}, 1)), \quad \forall \|\mathbf{u}\| \leq \omega, \quad \mathbf{u} \in \mathcal{U}, \quad (3.33)$$

where  $\mathcal{E}(\mathbf{u}, 1)$  is given by (2.3) with  $t = 1$ . Let  $\gamma > 0$ ,  $\mu > 0$  and  $\text{dist}_{\mathcal{B}}(\mathbf{u}, \mathcal{U}^*)$  be defined by

$$\text{dist}_{\mathcal{B}}(\mathbf{u}, \mathcal{U}^*) = \min \{ \sqrt{\mathcal{B}(\mathbf{u}^*, \mathbf{u})} \mid \mathbf{u}^* \in \mathcal{U}^* \} \quad (3.34)$$

with

$$\mathcal{B}(\mathbf{u}^*, \mathbf{u}) := \gamma \mathcal{B}_\phi(y^*, y) + \mu \mathcal{B}_\psi(x^*, x).$$

Moreover, we assume that the gradients of  $\psi(\cdot)$  and  $\phi(\cdot)$  are Lipschitz continuous with modulus  $L_\psi$  and  $L_\phi$ , respectively.

Hereafter, we establish the global linear convergence of Algorithm 3.1 under Assumption 3.2 in the context of a generalized error bound condition defined by Bregman distance. We begin our analysis with the following lemma.

**Lemma 3.3.** Let  $\{(x^{k+1}, \tilde{y}^{k+1}, y^{k+1})\}$  be a sequence generated by Algorithm 3.1. Then, we have

$$\begin{aligned} \text{dist}^2(0, \mathcal{E}(\mathbf{u}^{k+1}, 1)) &\leq 2(\|AA^\top\| + \gamma^2 L_\phi^2) \|\tilde{y}^{k+1} - y^{k+1}\|^2 \\ &\quad + 2\mu^2 L_\psi^2 \|x^{k+1} - x^k\|^2 + 2\gamma^2 L_\phi^2 \|\tilde{y}^{k+1} - y^k\|^2, \end{aligned} \quad (3.35)$$

where  $L_\phi$  and  $L_\psi$  are the Lipschitz continuity constants of  $\nabla \phi(\cdot)$  and  $\nabla \psi(\cdot)$ , respectively.

*Proof.* Letting  $\xi^{k+1} \in \partial f(x^{k+1})$ , it then follows from the first-order optimality condition of  $x$ -subproblem (3.2) that

$$x^{k+1} = \Pi_{\mathcal{X}}[x^{k+1} - (\xi^{k+1} - A^\top \tilde{y}^{k+1} + \mu \nabla \mathcal{B}_\psi(x^{k+1}, x^k))], \quad (3.36)$$

where

$$\nabla \mathcal{B}_\psi(x^{k+1}, x^k) = \langle \nabla \psi(x^{k+1}) - \nabla \psi(x^k), x^{k+1} - x^k \rangle.$$

By using (3.36) and the nonexpansiveness of projection operator  $\Pi_{\mathcal{X}}(\cdot)$ , we have

$$\begin{aligned} \text{dist}^2(0, \mathcal{E}_{\mathcal{X}}(\mathbf{u}^{k+1}, 1)) &= \text{dist}^2(x^{k+1}, \Pi_{\mathcal{X}}[x^{k+1} - (\partial f(x^{k+1}) + A^\top y^{k+1})]) \\ &\leq \|\Pi_{\mathcal{X}}[x^{k+1} - (\xi^{k+1} + A^\top \tilde{y}^{k+1} + \mu \nabla \mathcal{B}_\psi(x^{k+1}, x^k))] \\ &\quad - \Pi_{\mathcal{X}}[x^{k+1} - (\xi^{k+1} + A^\top y^{k+1})]\|^2 \\ &\leq \|A^\top(y^{k+1} - \tilde{y}^{k+1}) - \mu \nabla \mathcal{B}_\psi(x^{k+1}, x^k)\|^2 \\ &\leq 2\|AA^\top\| \|\tilde{y}^{k+1} - y^{k+1}\|^2 + 2\mu^2 \|\nabla \mathcal{B}_\psi(x^{k+1}, x^k)\|^2 \\ &\leq 2\|AA^\top\| \|\tilde{y}^{k+1} - y^{k+1}\|^2 + 2\mu^2 L_\psi^2 \|x^{k+1} - x^k\|^2, \end{aligned} \quad (3.37)$$

where the second inequality is derived by the fact that  $\|a + b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$  holds for all  $a, b \in \mathbb{R}^n$ , and the last inequality follows from the Lipschitz continuity of  $\nabla \psi(\cdot)$ .

Similarly, it follows from the first-order optimality condition of the  $y$ -subproblem (3.3) that

$$y^{k+1} = \Pi_{\mathcal{Y}}[y^{k+1} - (\zeta^{k+1} - Ax^{k+1} + \gamma \nabla \mathcal{B}_\phi(y^{k+1}, y^k))].$$

Then, we have

$$\begin{aligned} \text{dist}^2(0, \mathcal{E}_{\mathcal{Y}}(\mathbf{u}^{k+1}, 1)) &= \text{dist}^2(y^{k+1}, \Pi_{\mathcal{Y}}[y^{k+1} - (\partial g(y^{k+1}) - Ax^{k+1})]) \\ &\leq \|\Pi_{\mathcal{Y}}[y^{k+1} - (\zeta^{k+1} - Ax^{k+1} + \gamma \nabla \mathcal{B}_\phi(y^{k+1}, y^k))] \\ &\quad - \Pi_{\mathcal{Y}}[y^{k+1} - (\zeta^{k+1} - Ax^{k+1})]\|^2 \\ &\leq \gamma^2 \|\nabla \mathcal{B}_\phi(y^{k+1}, y^k)\|^2 \\ &\leq \gamma^2 L_\phi^2 \|y^{k+1} - y^k\|^2 \\ &\leq 2\gamma^2 L_\phi^2 \|y^{k+1} - \tilde{y}^{k+1}\|^2 + 2\gamma^2 L_\phi^2 \|\tilde{y}^{k+1} - y^k\|^2. \end{aligned} \quad (3.38)$$

Consequently, combining (3.37) and (3.38) leads to

$$\begin{aligned} \text{dist}^2(0, \mathcal{E}(\mathbf{u}^{k+1}, 1)) &= \text{dist}^2(0, \mathcal{E}_{\mathcal{X}}(\mathbf{u}^{k+1}, 1)) + \text{dist}^2(0, \mathcal{E}_{\mathcal{Y}}(\mathbf{u}^{k+1}, 1)) \\ &\leq 2(\|AA^\top\| + \gamma^2 L_\phi^2) \|\tilde{y}^{k+1} - y^{k+1}\|^2 \\ &\quad + 2\mu^2 L_\psi^2 \|x^{k+1} - x^k\|^2 + 2\gamma^2 L_\phi^2 \|\tilde{y}^{k+1} - y^k\|^2. \end{aligned}$$

Hence, the assertion of this lemma is obtained.  $\square$

Now, we establish the linear convergence rate of Algorithm 3.1 by the following theorem.

**Theorem 3.3.** *Let  $\{(x^{k+1}, \tilde{y}^{k+1}, y^{k+1})\}$  be the sequence generated by Algorithm 3.1. Suppose that Assumptions 3.1 and 3.2 hold. Then, we have*

$$(1 + \vartheta) \text{dist}_{\mathcal{Z}}^2(\mathbf{u}^{k+1}, \mathcal{U}^*) \leq \text{dist}_{\mathcal{Z}}^2(\mathbf{u}^k, \mathcal{U}^*),$$

where  $\vartheta$  is a positive constant given by

$$\vartheta = \min \left\{ \frac{\varrho}{4\gamma\eta^2 L_\phi^2}, \frac{\gamma\alpha\varrho - 1}{4\alpha\eta^2 (\|AA^\top\| + \gamma^2 L_\phi^2)}, \frac{\kappa}{4\mu\eta^2 L_\psi^2} \right\}.$$

*Proof.* It first follows from (3.30) that

$$\begin{aligned}
& \gamma \mathcal{B}_\phi(y^\star, y^{k+1}) + \mu \mathcal{B}_\psi(x^\star, x^{k+1}) \\
& \leq \gamma \mathcal{B}_\phi(y^\star, y^k) + \mu \mathcal{B}_\psi(x^\star, x^k) - \gamma \mathcal{B}_\phi(\tilde{y}^{k+1}, y^k) \\
& \quad - \left( \gamma - \frac{1}{\alpha \varrho} \right) \mathcal{B}_\phi(y^{k+1}, \tilde{y}^{k+1}) - \widehat{\beta} \|x^{k+1} - x^k\|^2.
\end{aligned} \tag{3.39}$$

In accordance with Assumption 3.2, i.e. inequality (3.33), it follows from (3.35) that

$$\begin{aligned}
\text{dist}_{\mathcal{B}}^2(\mathbf{u}^{k+1}, \mathcal{U}^\star) & \leq \eta^2 \text{dist}^2(0, \mathcal{E}(\mathbf{u}^{k+1}, 1)) \\
& \leq \eta^2 [2(\|AA^\top\| + \gamma^2 L_\phi^2) \|\tilde{y}^{k+1} - y^{k+1}\|^2 \\
& \quad + 2\mu^2 L_\psi^2 \|x^{k+1} - x^k\|^2 + 2\gamma^2 L_\phi^2 \|\tilde{y}^{k+1} - y^k\|^2].
\end{aligned} \tag{3.40}$$

Consequently, by the definition of  $\text{dist}_{\mathcal{B}}(\mathbf{u}, \mathcal{U}^\star)$  given by (3.34), we have

$$\begin{aligned}
& (1 + \vartheta) \text{dist}_{\mathcal{B}}^2(\mathbf{u}^{k+1}, \mathcal{U}^\star) \\
& = \text{dist}_{\mathcal{B}}^2(\mathbf{u}^{k+1}, \mathcal{U}^\star) + \vartheta \text{dist}_{\mathcal{B}}^2(\mathbf{u}^{k+1}, \mathcal{U}^\star) \\
& \leq [\gamma \mathcal{B}_\phi(y^\star, y^{k+1}) + \mu \mathcal{B}_\psi(x^\star, x^{k+1})] + \vartheta \text{dist}_{\mathcal{B}}^2(\mathbf{u}^{k+1}, \mathcal{U}^\star) \\
& \leq \gamma \mathcal{B}_\phi(y^\star, y^k) + \mu \mathcal{B}_\psi(x^\star, x^k) - \gamma \mathcal{B}_\phi(\tilde{y}^{k+1}, y^k) \\
& \quad - \left( \gamma - \frac{1}{\alpha \varrho} \right) \mathcal{B}_\phi(y^{k+1}, \tilde{y}^{k+1}) - \widehat{\beta} \|x^{k+1} - x^k\|^2 \\
& \quad + \vartheta \eta^2 [2(\|AA^\top\| + \gamma^2 L_\phi^2) \|\tilde{y}^{k+1} - y^{k+1}\|^2 \\
& \quad + 2\mu^2 L_\psi^2 \|x^{k+1} - x^k\|^2 + 2\gamma^2 L_\phi^2 \|\tilde{y}^{k+1} - y^k\|^2] \\
& \leq \gamma \mathcal{B}_\phi(y^\star, y^k) + \mu \mathcal{B}_\psi(x^\star, x^k) - \left[ \frac{\gamma \varrho}{2} - 2\vartheta \eta^2 \gamma^2 L_\phi^2 \right] \|\tilde{y}^{k+1} - y^k\|^2 \\
& \quad - \left[ \frac{\gamma \alpha \varrho - 1}{2\alpha} - 2\vartheta \eta^2 (\|AA^\top\| + \gamma^2 L_\phi^2) \right] \|\tilde{y}^{k+1} - y^{k+1}\|^2 \\
& \quad - \left[ \frac{\widehat{\beta} \kappa}{2} - 2\vartheta \eta^2 \mu^2 L_\psi^2 \right] \|x^{k+1} - x^k\|^2,
\end{aligned} \tag{3.41}$$

where the second inequality is derived by (3.39), and the third inequality follows from the strong convexity of  $\mathcal{B}_\phi$  and  $\mathcal{B}_\psi$ . As a consequence, by using the definition of  $\vartheta$ , inequality (3.41) implies that

$$(1 + \vartheta) \text{dist}_{\mathcal{B}}^2(\mathbf{u}^{k+1}, \mathcal{U}^\star) \leq \gamma \mathcal{B}_\phi(y^\star, y^k) + \mu \mathcal{B}_\psi(x^\star, x^k) - \Gamma(x^{k+1}, \tilde{y}^{k+1}, y^{k+1}), \tag{3.42}$$

where  $\Gamma(x^{k+1}, \tilde{y}^{k+1}, y^{k+1})$  is a positive term composed by the last three terms of (3.41). Then, setting

$$\text{dist}_{\mathcal{B}}^2(\mathbf{u}^k, \mathcal{U}^\star) = \gamma \mathcal{B}_\phi(y^\star, y^k) + \mu \mathcal{B}_\psi(x^\star, x^k)$$

in (3.42) immediately arrives at

$$(1 + \vartheta) \text{dist}_{\mathcal{B}}^2(\mathbf{u}^{k+1}, \mathcal{U}^\star) \leq \gamma \mathcal{B}_\phi(y^\star, y^k) + \mu \mathcal{B}_\psi(x^\star, x^k) = \text{dist}_{\mathcal{B}}^2(\mathbf{u}^k, \mathcal{U}^\star).$$

The assertion of this theorem is obtained.  $\square$

## 4. Applications to Linearly Constrained Convex Minimization

In this section, we shall show that our Algorithm 3.1 will recover the iterative schemes of some classical first-order algorithms, when applying it to linearly constrained convex optimization problems.

### 4.1. One-block case: Augmented Lagrangian method

In this subsection, we consider the following one-block convex minimization problem with linear constraints:

$$\min_x \{f(x) \mid Ax = b, x \in \mathcal{X} \subseteq \mathbb{R}^n\}, \quad (4.1)$$

where  $f(\cdot) : \mathcal{X} \rightarrow (-\infty, +\infty]$  is a proper closed convex function and  $A \in \mathbb{R}^{m \times n}$  is a given matrix, and  $b \in \mathbb{R}^m$  is a given vector. Accordingly, its augmented Lagrangian function reads as

$$\mathcal{L}_\gamma(x, y) = f(x) + \langle Ax - b, y \rangle + \frac{1}{2\gamma} \|Ax - b\|^2, \quad \gamma > 0,$$

and the augmented Lagrangian method for (4.1) is

$$\begin{cases} x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \langle Ax - b, y^k \rangle + \frac{1}{2\gamma} \|Ax - b\|^2 \right\}, \\ y^{k+1} = y^k + \frac{1}{\gamma} (Ax^{k+1} - b), \end{cases}$$

or equivalently,

$$\begin{cases} x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \frac{1}{2\gamma} \|Ax - b + \gamma y^k\|^2 \right\}, \\ y^{k+1} = y^k + \frac{1}{\gamma} (Ax^{k+1} - b). \end{cases} \quad (4.2)$$

It is not difficult to observe that directly solving the  $x$ -subproblem in (4.2) is not an easy task (at least is not easily implementable), when  $f(x)$  is a nonsmooth function (e.g.  $\|x\|_1$  or nuclear norm for matrices) and  $A$  is a general matrix, even for the case  $\mathcal{X} = \mathbb{R}^n$  (e.g. see [60]). Therefore, a natural way to make (4.2) implementable is linearizing the quadratic penalty term at  $x^k$  as follows:

$$\frac{1}{2\gamma} \|Ax - b\|^2 \approx \frac{1}{2\gamma} \|Ax^k - b\|^2 + \frac{1}{\gamma} \langle A^\top (Ax^k - b), x - x^k \rangle + \frac{\mu}{2} \|x - x^k\|^2, \quad (4.3)$$

where  $\mu\gamma \geq \|A^\top A\|$ . Consequently, using the above approximation (4.3) instead of quadratic penalty term in (4.2), we immediately obtain the so-called linearized augmented Lagrangian method (LALM) for (4.1), i.e.

$$\begin{cases} x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \frac{\mu}{2} \left\| x - \left( x^k + \frac{1}{\mu} A^\top \left( y^k + \frac{1}{\gamma} (Ax^k - b) \right) \right) \right\|^2 \right\}, \\ y^{k+1} = y^k + \frac{1}{\gamma} (Ax^{k+1} - b). \end{cases} \quad (4.4)$$

When applying our Algorithm 3.1 to (4.1), we first reformulate (4.1) as the following saddle point problem:

$$\min_{x \in \mathcal{X}} \max_{y \in \mathbb{R}^m} \{ \mathcal{L}(x, y) := f(x) + \langle Ax, y \rangle - \langle b, y \rangle \}. \quad (4.5)$$

Then, the specific iterative scheme of Algorithm 3.1 for (4.5) reads as

$$\begin{cases} \tilde{y}^{k+1} = \arg \max_{y \in \mathbb{R}^m} \{ -\langle b, y \rangle + \langle Ax^k, y \rangle - \gamma \mathcal{B}_\phi(y, y^k) \}, \\ x^{k+1} = \arg \min_{x \in \mathcal{X}} \{ f(x) + \langle Ax, \tilde{y}^{k+1} \rangle + \mu \mathcal{B}_\psi(x, x^k) \}, \\ y^{k+1} = \arg \max_{y \in \mathbb{R}^m} \{ -\langle b, y \rangle + \langle Ax^{k+1}, y \rangle - \gamma \mathcal{B}_\phi(y, y^k) \}. \end{cases} \quad (4.6)$$

Clearly, by setting the Bregman kernel functions as  $\phi(y) = \|y\|^2/2$  and  $\psi(x) = \|x\|_{A^\top A}^2/2$ , the iterative scheme (4.6) reads as

$$\begin{cases} \tilde{y}^{k+1} = \arg \max_{y \in \mathbb{R}^m} \left\{ -\langle b, y \rangle + \langle Ax^k, y \rangle - \frac{\gamma}{2} \|y - y^k\|^2 \right\}, \\ x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \langle Ax, \tilde{y}^{k+1} \rangle + \frac{\mu}{2} \|Ax - Ax^k\|^2 \right\}, \\ y^{k+1} = \arg \max_{y \in \mathbb{R}^m} \left\{ -\langle b, y \rangle + \langle Ax^{k+1}, y \rangle - \frac{\gamma}{2} \|y - y^k\|^2 \right\}, \end{cases}$$

which, by using the first-order optimality conditions of both  $y$ -subproblems, can be immediately simplified as

$$\begin{cases} \tilde{y}^{k+1} = y^k + \frac{1}{\gamma}(Ax^k - b), \\ x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \frac{\mu}{2} \left\| Ax - \left( Ax^k - \frac{1}{\mu} \tilde{y}^{k+1} \right) \right\|^2 \right\}, \\ y^{k+1} = y^k + \frac{1}{\gamma}(Ax^{k+1} - b). \end{cases}$$

It is trivial that substituting  $\tilde{y}^{k+1}$  into the update of  $x^{k+1}$  and setting  $\mu = 1/\gamma$  immediately yields the ALM (4.2).

On the other hand, by setting the Bregman kernel functions as  $\phi(y) = \|y\|^2/2$  and  $\psi(x) = \|x\|^2/2$ , the iterative scheme (4.6) reads as

$$\begin{cases} \tilde{y}^{k+1} = \arg \max_{y \in \mathbb{R}^m} \left\{ -\langle b, y \rangle + \langle Ax^k, y \rangle - \frac{\gamma}{2} \|y - y^k\|^2 \right\}, \\ x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \langle Ax, \tilde{y}^{k+1} \rangle + \frac{\mu}{2} \|x - x^k\|^2 \right\}, \\ y^{k+1} = \arg \max_{y \in \mathbb{R}^m} \left\{ -\langle b, y \rangle + \langle Ax^{k+1}, y \rangle - \frac{\gamma}{2} \|y - y^k\|^2 \right\}, \end{cases}$$

which, by using the first-order optimality conditions of both  $y$ -subproblems, can be immediately simplified as

$$\begin{cases} \tilde{y}^{k+1} = y^k + \frac{1}{\gamma}(Ax^k - b), \\ x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \frac{\mu}{2} \left\| x - \left( x^k - \frac{1}{\mu} A^\top \tilde{y}^{k+1} \right) \right\|^2 \right\}, \\ y^{k+1} = y^k + \frac{1}{\gamma}(Ax^{k+1} - b). \end{cases}$$

Clearly, plugging the formula of  $\tilde{y}^{k+1}$  into the update scheme of  $x^{k+1}$  immediately yields the LALM (4.4).

It is interesting to notice that our algorithmic framework allows us to take different Bregman kernel functions. Therefore, we here follow the novel idea of the newly introduced balanced augmented Lagrangian method [32] to further consider taking  $\phi(y) = \|y\|_{(AA^\top + \epsilon I)}^2/2$  and  $\psi(x) = \|x\|^2/2$  for model (4.1), where  $I$  stands for an identity matrix which always automatically matches the size for operations, and  $I_n$  specially denotes the  $n$ -dimensional identity matrix. Specifically, the concrete iterative scheme of Algorithm 3.1 reads as

$$\begin{cases} \tilde{y}^{k+1} = y^k + \frac{1}{\gamma}(AA^\top + \epsilon I)^{-1}(Ax^k - b), \\ x^{k+1} = \arg \min_{x \in \mathcal{X}} \left\{ f(x) + \frac{\mu}{2} \left\| x - \left( x^k - \frac{1}{\mu} A^\top \tilde{y}^{k+1} \right) \right\|^2 \right\}, \\ y^{k+1} = y^k + \frac{1}{\gamma}(AA^\top + \epsilon I)^{-1}(Ax^{k+1} - b). \end{cases} \quad (4.7)$$

It should be noted that, for the case where  $n \ll m$ , we can employ the well-known Sherman-Morrison-Woodbury theorem to reduce the computational cost of computing  $(AA^\top + \epsilon I)^{-1}$ , i.e.

$$(AA^\top + \epsilon I)^{-1} = \epsilon^{-1}(I_m - A(\epsilon I_n + A^\top A)^{-1}A^\top).$$

In what follows, we call the scheme (4.7) doubly balanced augmented Lagrangian method (DBALM). Moreover, when  $f(x)$  is a quadratic function, i.e.  $f(x) = \|Bx - q\|^2/2$ , we can also take  $\psi(x) = \|x\|_{(\tau I - B^\top B)}^2/2$  to derive a linearized version of DBALM to deal with the case where  $\mathcal{X}$  is a simple convex set whose projection is easily calculated.

## 4.2. Multi-block case: Jacobian splitting method

In this part, we are concerned with the multi-block linearly constrained convex minimization, which takes the form

$$\min_{x_1, x_2, \dots, x_p} \left\{ \sum_{i=1}^p f_i(x_i) \mid \sum_{i=1}^p A_i x_i = b, x_i \in \mathcal{X}_i \subseteq \mathbb{R}^{n_i}, i = 1, 2, \dots, p \right\}, \quad (4.8)$$

where  $f_i(\cdot) : \mathcal{X}_i \rightarrow (-\infty, +\infty]$  for  $i = 1, 2, \dots, p$  are proper closed convex functions,  $A_i \in \mathbb{R}^{m \times n_i}$  are given matrices, and  $b \in \mathbb{R}^m$  is a given vector. In the past decades, the multi-block model (4.8) has received much considerable attention due to its widespread applications in computer sciences and automatic control, e.g. see [23, 55]. Although such a model is also a linearly constrained optimization problem, it cannot be easily solved via the aforementioned ALM (4.2) since the linear constraints make the ALM suffer from coupled subproblems so that the separability of the objective function cannot be fully exploited in algorithmic implementation. Accordingly, a series of augmented Lagrangian-based splitting methods were developed in the optimization literature, e.g. see [23–25, 28, 34, 59] and references therein.

Below, we first show that our Algorithm 3.1 is applicable to solving (4.8). In particular, we can easily derive that our Algorithm 3.1 is indeed the fully Jacobian splitting algorithm [28, 57] by choosing appropriate Bregman kernel functions. Moreover, we can obtain some new Jacobian splitting methods for (4.8).

First, it is clear that (4.8) can be rewritten into a compact form as follows:

$$\min_x \{ \mathbf{f}(x) \mid \mathbf{A}x = b, x \in \mathcal{X} \}, \quad (4.9)$$

where

$$\begin{aligned}\mathbf{x} &:= (x_1^\top, x_2^\top, \dots, x_p^\top)^\top, \quad \mathbf{f}(\mathbf{x}) := \sum_{i=1}^p f_i(x_i), \\ \mathbf{A} &:= [A_1, A_2, \dots, A_p], \quad \mathcal{X} := \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_p.\end{aligned}$$

Therefore, we can reformulate (4.9) as the form of (4.5), i.e.

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{y \in \mathbb{R}^m} \{ \mathcal{L}(\mathbf{x}, y) := \mathbf{f}(\mathbf{x}) + \langle \mathbf{A}\mathbf{x}, y \rangle - \langle b, y \rangle \}. \quad (4.10)$$

Then, the specific iterative scheme of Algorithm 3.1 for (4.5) reads as

$$\begin{cases} \tilde{y}^{k+1} = \arg \max_{y \in \mathbb{R}^m} \{ -\langle b, y \rangle + \langle \mathbf{A}\mathbf{x}^k, y \rangle - \gamma \mathcal{B}_\phi(y, y^k) \}, \\ \mathbf{x}^{k+1} = \arg \min_{\mathbf{x} \in \mathcal{X}} \{ \mathbf{f}(\mathbf{x}) + \langle \mathbf{A}\mathbf{x}, \tilde{y}^{k+1} \rangle + \mu \mathcal{B}_\psi(\mathbf{x}, \mathbf{x}^k) \}, \\ y^{k+1} = \arg \max_{y \in \mathbb{R}^m} \{ -\langle b, y \rangle + \langle \mathbf{A}\mathbf{x}^{k+1}, y \rangle - \gamma \mathcal{B}_\phi(y, y^k) \}. \end{cases} \quad (4.11)$$

Now, we denote

$$M := \begin{pmatrix} \beta_1 A_1^\top A_1 & 0 & \dots & 0 \\ 0 & \beta_2 A_2^\top A_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \beta_p A_p^\top A_p \end{pmatrix}.$$

Then, by setting the Bregman kernel functions as  $\phi(y) = \|y\|^2/2$  and  $\psi(\mathbf{x}) = \|\mathbf{x}\|_M^2/2$  and using the separability of the objective function, for  $\mu = 1$ , the iterative scheme (4.11) reads as

$$\begin{cases} \tilde{y}^{k+1} = y^k + \frac{1}{\gamma} \left( \sum_{i=1}^p A_i x_i^k - b \right), \\ x_i^{k+1} = \arg \min_{x_i \in \mathcal{X}_i} \left\{ f_i(x_i) + \frac{\beta_i}{2} \left\| A x_i - \left( A x_i^k - \frac{1}{\beta_i} \tilde{y}^{k+1} \right) \right\|^2 \right\}, \quad i = 1, \dots, p, \\ y^{k+1} = y^k + \frac{1}{\gamma} \left( \sum_{i=1}^p A_i x_i^{k+1} - b \right), \end{cases} \quad (4.12)$$

which actually corresponds to the proximal Jacobian splitting method studied in [28], especially precisely coincides with the augmented Lagrangian-based parallel splitting method [57] by setting  $\beta_i = 1/\gamma$  for  $i = 1, 2, \dots, p$ .

More interestingly, by setting the Bregman kernel functions as  $\phi(y) = \|y\|^2/2$  and  $\psi(\mathbf{x}) = \|\mathbf{x}\|^2/2$ , the specific iterative scheme of Algorithm 3.1 for (4.10) reads as

$$\begin{cases} \tilde{y}^{k+1} = y^k + \frac{1}{\gamma} \left( \sum_{i=1}^p A_i x_i^k - b \right), \\ x_i^{k+1} = \arg \min_{x_i \in \mathcal{X}_i} \left\{ f_i(x_i) + \frac{\mu}{2} \left\| x_i - \left( x_i^k - \frac{1}{\mu} A_i^\top \tilde{y}^{k+1} \right) \right\|^2 \right\}, \quad i = 1, \dots, p, \\ y^{k+1} = y^k + \frac{1}{\gamma} \left( \sum_{i=1}^p A_i x_i^{k+1} - b \right), \end{cases} \quad (4.13)$$

which is a linearized parallel splitting method for (4.8). Comparing with the methods discussed in [21, 23, 25], the variant (4.13) enjoys relatively simpler iterative scheme without correction steps. Combining the ideas of (4.12) and (4.13), we can take

$$\widehat{M} := \begin{pmatrix} \beta_1 A_1^\top A_1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \beta_l A_l^\top A_l & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \beta_{l+1} I & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \beta_p I \end{pmatrix},$$

so that  $\phi(y) = \|y\|^2/2$  and  $\psi(\mathbf{x}) = \|\mathbf{x}\|_{\widehat{M}}^2/2$ , thereby producing a partially linearized Jacobian splitting method (PLJSM) for (4.8) with setting  $\mu = 1$ , i.e.

$$\begin{cases} \tilde{y}^{k+1} = y^k + \frac{1}{\gamma} \left( \sum_{i=1}^p A_i x_i^k - b \right), \\ x_i^{k+1} = \arg \min_{x_i \in \mathcal{X}_i} \left\{ f_i(x_i) + \frac{\beta_i}{2} \left\| Ax_i - \left( Ax_i^k - \frac{1}{\beta_i} \tilde{y}^{k+1} \right) \right\|^2 \right\}, \quad i = 1, \dots, l, \\ x_j^{k+1} = \arg \min_{x_j \in \mathcal{X}_j} \left\{ f_j(x_j) + \frac{\beta_j}{2} \left\| x_j - \left( x_j^k - \frac{1}{\beta_j} A_j^\top \tilde{y}^{k+1} \right) \right\|^2 \right\}, \quad j = l+1, \dots, p, \\ y^{k+1} = y^k + \frac{1}{\gamma} \left( \sum_{i=1}^p A_i x_i^{k+1} - b \right), \end{cases}$$

which, to our best knowledge, is not discussed in the literature. Of course, we can also follow the spirit of the DBALM (4.7) to specify  $\phi(y) = \|y\|_{(\sum_{i=1}^p A_i A_i^\top + \epsilon I)}^2/2$  and  $\psi(\mathbf{x}) = \|\mathbf{x}\|_{\widehat{M}}^2/2$  to develop a doubly balanced PLJSM for (4.8).

## 5. Numerical Experiments

In this section, we conduct the numerical performance of Algorithm 3.1 (denoted by SPIDA) on some well tested problems, including the basis pursuit, RPCA, and image restoration with synthetic and real-world datasets. We also compare our Algorithm 3.1 with some existing state-of-the-art primal-dual-type algorithms for the purpose of showing the numerical improvement of our Algorithm 3.1. All algorithms are implemented in MATLAB 2021a and all experiments are conducted on a 64-bit Windows personal computer with Intel(R) Core(TM) i5-12500h CPU@2.50 GHz and 8 GB of RAM.

### 5.1. Basis pursuit

As discussed in Section 4, our Algorithm 3.1 (SPIDA) is applicable to dealing with linearly constrained optimization problem (4.1). In this part, we are interested in the basis pursuit problem, which can be expressed mathematically as an  $\ell_1$ -norm minimization problem with linear constraints, i.e.

$$\min_{x \in \mathbb{R}^n} \{\|x\|_1 \mid Ax = b\}, \quad (5.1)$$

where  $A \in \mathbb{R}^{m \times n}$  is a sample matrix and  $b \in \mathbb{R}^m$  is a measurement vector. Such a model (5.1) is a fundamental problem in compressed sensing [18] and can be efficiently solved via a large number of optimization solvers. Here, we just employ this example to investigate the ability of our SPIDA on solving linearly constrained optimization problems, in addition to showing the superiority of SPIDA over some popular first-order optimization methods.

First, by the Lagrangian function, we reformulate (5.1) as the following min-max saddle point problem:

$$\min_{x \in \mathbb{R}^n} \max_{y \in \mathbb{R}^m} \{\mathcal{L}(x, y) = \|x\|_1 + \langle Ax, y \rangle - \langle b, y \rangle\}.$$

We compare our SPIDA with PDHG (setting  $\tau = 1$  in (1.3)) and GRPDA [13]. Most recently, He and Yuan [32] introduced a novel balanced augmented Lagrangian method (BALM) for linearly convex programming, which is a great improvement of the classical ALM (4.2). The iterative scheme of BALM for (5.1) reads as

$$\begin{cases} x^{k+1} = \arg \min_{x \in \mathbb{R}^n} \left\{ f(x) + \frac{\gamma}{2} \left\| x - \left( x^k - \frac{1}{\gamma} A^\top y^k \right) \right\|^2 \right\}, \\ y^{k+1} = y^k + \left( \frac{1}{\gamma} AA^\top + \epsilon I \right)^{-1} (A(2x^{k+1} - x^k) - b). \end{cases} \quad (5.2)$$

In this part, we also compare our SPIDA with the BALM (5.2). Moreover, we will follow the idea of BALM to produce a doubly balanced augmented Lagrangian method (see (4.7) and denote it by SPIDA-II) via choosing the type II Bregman kernel function in Table 2.1. As discussed in Section 4.1, when the Bregman kernel functions are specified as the type I of Table 2.1, our SPIDA reduces to the LALM (4.4), which will be denoted by SPIDA-I in our numerical comparison.

In the experiments, we first construct a randomly  $s$ -sparse vector  $x^* \in \mathbb{R}^n$ , where  $s$  is the number of nonzero components. Then, we randomly generate a sample matrix  $A \in \mathbb{R}^{m \times n}$  to construct the measurement vector  $b$  via  $b = Ax^*$ . Here, we consider two different ways to generate the sample matrix  $A$ :

- $A$  is a random Gaussian matrix.
- $A$  is a random partial DCT (discrete cosine transform) matrix.

We conduct different sizes of the problems by setting  $(m, n, s) = (180i, 960i, 30i), i = 1, 2, \dots, 10$ . Besides, to implement these algorithms, we take  $(\gamma, \mu) = (1, 1)$  for PDHG,  $(\varrho, \tau, \sigma) = (2, \sqrt{2}/2, \sqrt{2}/2)$  for GRPDA (where  $\varrho$  is the parameter associated with the Golden ratio step, and  $(\tau, \sigma)$  are the proximal parameters for the  $x$ - and  $y$ -subproblems, respectively),  $(\gamma, \epsilon) = (1.5, 0.015)$  for BALM (5.2),  $(\gamma, \mu) = (0.6, 0.6)$  for SPIDA-I and  $(\gamma, \mu, \epsilon) = (0.6, 0.6, 0.01)$  for SPIDA-II. All algorithms start with zero initial points and stop at

$$\text{Tol} := \frac{\|(x^{k+1}, y^{k+1}) - (x^k, y^k)\|}{\|(x^k, y^k)\|} \leq \varepsilon. \quad (5.3)$$

They are terminated when satisfying the stopping criterion (5.3) with  $\varepsilon = 10^{-6}$ . The average calculation results of 10 trials of randomly generating matrix  $A$  are summarized in Tables 5.1 and 5.2.

It can be easily seen from Table 5.1 that our SPIDA-II takes the fewest iterations to obtain approximate solutions for the case where  $A$  is a random Gaussian matrix. When dealing with

Table 5.1: Numerical results for basis pursuit: (i)  $A$  is a random Gaussian matrix.

$i$	PDHG	GRPDA	BALM	SPIDA-I	SPIDA-II
	Iter. / Time	Iter. / Time	Iter. / Time	Iter. / Time	Iter. / Time
$i = 1$	689.5 / 1.36	587.5 / 1.16	292.6 / 0.60	348.3 / 0.70	167.6 / 0.41
$i = 2$	1036.8 / 6.36	811.2 / 4.98	374.7 / 2.33	572.9 / 3.57	257.4 / 1.68
$i = 3$	1525.3 / 16.20	1135.2 / 12.06	518.8 / 5.58	875.2 / 9.42	393.5 / 6.54
$i = 4$	1595.8 / 27.53	1182.8 / 20.39	536.6 / 9.38	919.4 / 16.02	410.7 / 10.73
$i = 5$	1295.3 / 31.37	980.6 / 23.78	451.2 / 11.19	737.3 / 18.03	335.9 / 11.96
$i = 6$	3272.4 / 113.45	2346.8 / 81.19	1037.2 / 36.20	1938.0 / 67.91	887.9 / 42.56
$i = 7$	1038.6 / 43.53	794.8 / 33.35	380.3 / 16.12	583.8 / 24.82	275.2 / 15.78
$i = 8$	1567.3 / 85.37	1161.6 / 63.31	538.2 / 29.68	906.2 / 50.64	415.7 / 30.49
$i = 9$	7314.6 / 472.55	5177.1 / 334.80	2389.7 / 155.92	4379.6 / 290.47	2073.1 / 179.49
$i = 10$	8750.3 / 740.80	6198.3 / 527.87	2667.6 / 227.76	5238.9 / 450.59	2299.4 / 250.56

Table 5.2: Numerical results for basis pursuit: (ii)  $A$  is a random partial DCT matrix.

$i$	PDHG	GRPDA	BALM	SPIDA-I	SPIDA-II
	Iter. / Time	Iter. / Time	Iter. / Time	Iter. / Time	Iter. / Time
$i = 1$	291.2 / 0.57	253.8 / 0.50	255.5 / 0.52	142.0 / 0.29	143.8 / 0.36
$i = 2$	1278.4 / 8.21	901.5 / 5.77	873.4 / 5.63	756.0 / 4.95	763.8 / 5.46
$i = 3$	397.0 / 4.61	325.0 / 3.76	322.3 / 3.77	207.9 / 2.43	210.1 / 3.79
$i = 4$	1207.0 / 22.70	873.6 / 16.45	846.5 / 16.07	707.6 / 13.40	714.8 / 18.49
$i = 5$	366.1 / 9.10	304.2 / 7.59	304.1 / 7.63	189.3 / 4.79	191.2 / 6.71
$i = 6$	2558.8 / 91.87	1843.5 / 67.13	1781.4 / 65.38	1523.1 / 56.47	1538.3 / 73.30
$i = 7$	1317.5 / 60.81	965.5 / 44.67	936.3 / 43.68	772.1 / 36.32	779.8 / 46.91
$i = 8$	3931.9 / 231.36	2757.4 / 161.90	2658.5 / 157.81	2353.4 / 142.21	2376.7 / 181.08
$i = 9$	631.7 / 45.56	482.6 / 34.82	471.0 / 34.29	357.9 / 26.63	361.6 / 33.40
$i = 10$	1001.0 / 87.46	728.6 / 63.62	707.4 / 62.20	584.1 / 52.57	589.9 / 65.05

the other case where  $A$  is a partial DCT matrix, results in Table 5.2 tell us that SPIDA-I and SPIDA-II have the almost same performance, while taking fewer iterations to achieve high-quality solutions than the other three first-order algorithms for (5.1). These computational results demonstrate that the symmetric updating way on the dual variable (i.e. twice calculations) equipped with a general proximal regularization can improve the numerical performance of the classical ALM. To further show the convergence behavior of our SPIDA, we focus on the case with  $(m, n, s) = (360, 1920, 60)$  and plot the convergence curve of the relative error defined by  $\|x^k - x^*\|/\|x^*\|$  with respect to iterations in Fig. 5.1. We see from Fig. 5.1 that our SPIDA has a promisingly linear convergence behavior for one-block linearly constrained optimization problems.

Due to the randomness of the generated data sets for basis pursuit, we are further interested in the stability of our SPIDA in practice. Therefore, we show the averaged time and iterations of ten trials by the bars, and their standard deviations by the line segments for the cases  $i = 9$  and  $i = 10$  in Fig. 5.2. Comparatively, it is illustrated in Fig. 5.2 that our SPIDA performs stably for the random data sets.

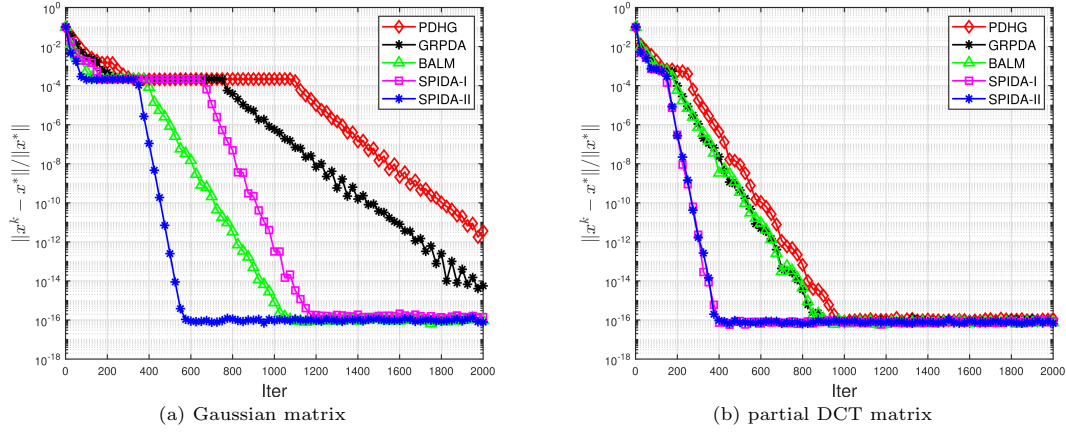


Fig. 5.1. Evolution of the relative error with respect to iterations for solving the basis pursuit problem (5.1).

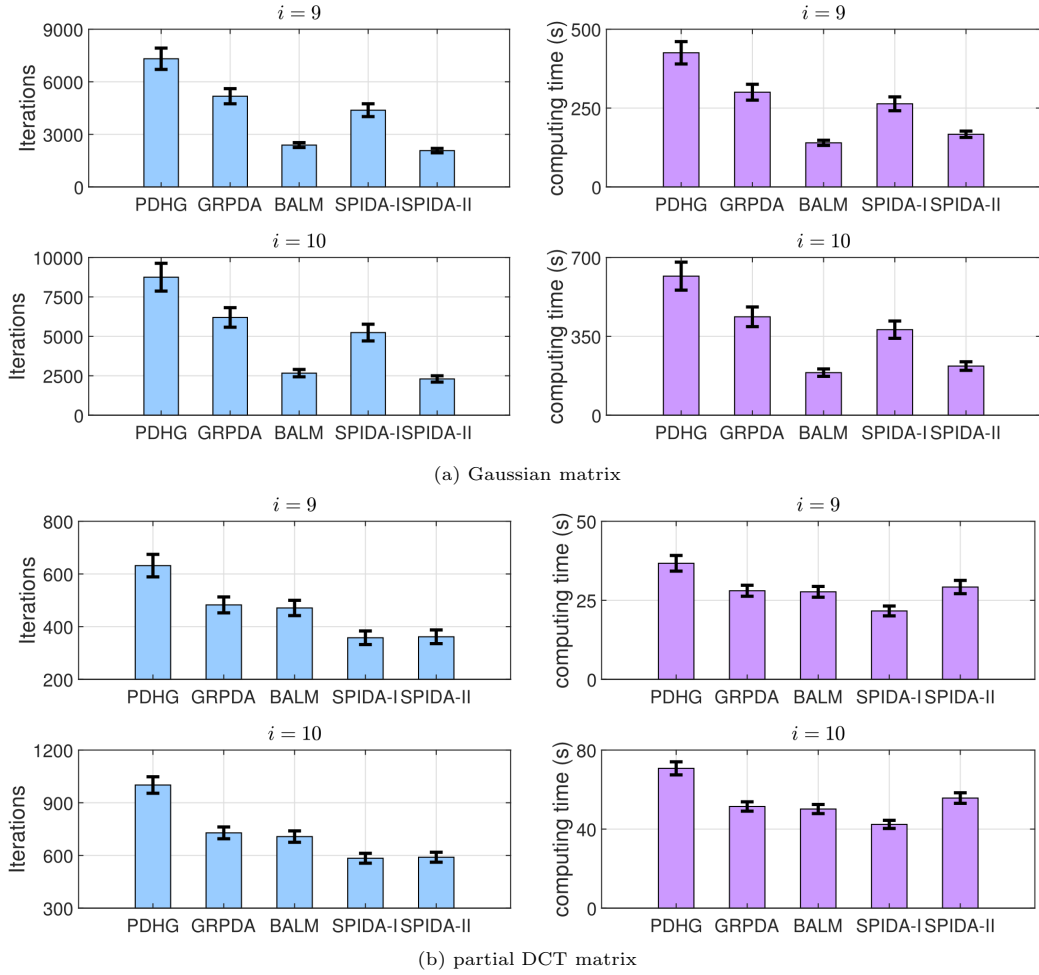


Fig. 5.2. Stability investigation of all algorithms for solving basis pursuit by setting  $i = 9$  and  $i = 10$ , where the averaged iterations and computing time are shown by the bars, and their standard deviations is plotted by the line segments.

## 5.2. RPCA

In this subsection, we give a numerical feedback to the applicability of our SPIDA to multi-block convex programming discussed in Section 4.2. Therefore, we consider a well-studied RPCA model [9], which refers to the task of recovering a sparse matrix and a low-rank one. Specifically, the RPCA under consideration takes the form

$$\min_{X,Z} \{\|X\|_* + \lambda\|Z\|_1 \mid X + Z = H\}, \quad (5.4)$$

where  $\|X\|_*$  is the nuclear norm (i.e. the sum of all singular values of  $X$ ) for promoting the low-rankness of  $X \in \mathbb{R}^{m \times n}$ ,  $\|Z\|_1$  represents the  $\ell_1$ -norm for inducing the sparsity of  $Z \in \mathbb{R}^{m \times n}$ , and  $H \in \mathbb{R}^{m \times n}$  is a given matrix. Obviously, the RPCA model (5.4) is a special case of the multi-block model (4.8) with two-block structure. Therefore, we can easily reformulate (5.4) as the following separable saddle point problem:

$$\min_{X,Z} \max_Y \{\mathcal{L}(X, Z, Y) = \|X\|_* + \lambda\|Z\|_1 + \langle X + Z, Y \rangle - \langle H, Y \rangle\}.$$

In this part, we also mainly compare SPIDA with PDHG and GRPDA. Here, we consider the RPCA with synthetic and real data sets to verify the reliability of our algorithm for multi-block convex programming.

We first conduct the numerical performance of these algorithms on synthetic data sets. In this situation, we generate a low-rank matrix  $X^*$  via  $X^* = UV$ , where  $U \in \mathbb{R}^{n \times r}$  and  $V \in \mathbb{R}^{r \times n}$  are independently random matrices whose entries are drawn from Gaussian distribution  $\mathcal{N}(0, 1)$ . Then, we generate a sparse matrix  $Z^*$  by randomly choosing a support set  $\Omega$  of size  $0.1 \times n^2$  (i.e. 10% nonzero components), and all elements are independently sampled from a uniform distribution in  $[-50, 50]$ . Finally, we let  $H = X^* + Z^*$  be the observed matrix. Clearly,  $(X^*, Z^*)$  is the true solution of (5.4). Throughout our experiments, we still employ the stopping criterion (5.3) with setting  $\varepsilon = 10^{-5}$  for all algorithms. Besides, we take  $(\gamma, \mu) = (70.7107, 0.0283)$  for PDHG,

$$(\psi, \tau, \sigma) = \left(1.618, \frac{70.7107}{\sqrt{1.618}}, \frac{0.0283}{\sqrt{1.618}}\right)$$

for GRPDA,  $(\gamma, \mu) = (0.77 * 70.7107, 0.0283)$  for SPIDA. In Table 5.3, we additionally report the rank of the obtained low-rank matrix ( $\text{rank}(\hat{X})$ ), the number of nonzero components of the obtained sparse matrix ( $\|\hat{Z}\|_0$ ), the relative error (**Rerr**) defined by

$$\text{Rerr} = \frac{\|\hat{X} + \hat{Z} - X^* - Z^*\|_F}{\|X^* + Z^*\|_F},$$

where  $\hat{X}$  and  $\hat{Z}$  respectively represent the low-rank and sparse matrices obtained by the algorithms. It can be seen from Table 5.3 that our SPIDA takes less iterations and computing time than both PDHG and GRPDA to achieve almost the same low-rank and sparse separation on the observed matrix  $H$ .

Below, we are concerned with the numerical performance of SPIDA on RPCA with real data sets. So, we consider the application of model (5.4) in background separation of surveillance video. Here, we select three well-tested videos, i.e. **Shoppingmall**, **Lobby**, and **Hall Airport**, and select the first 200 frames of each video to construct an observed matrix  $H \in \mathbb{R}^{n \times 200}$ , where  $n = n_1 \times n_2$  with  $n_1$  and  $n_2$  representing the height and width of the video, respectively. Notice that the true rank and sparsity of these videos are unknown. Therefore, we shall report

the number of iterations, the computing time in seconds, the objective values (**Obj**), and the error (**Err**) defined by

$$\mathbf{Err} = \frac{\|\hat{X} + \hat{Z} - H\|_F}{\|H\|_F}.$$

Throughout, we set  $\varepsilon = 5 \times 10^{-4}$  in (5.3) as the stopping tolerance for all algorithms. Computational results are summarized in Table 5.4, which also demonstrate that our SPIDA runs a little faster than both PDHG and GRPDA for real-world data sets. In Fig. 5.3, we list the separated background and foreground of some frames. We can see from these results that all primal-dual-type algorithms are reliable for multi-block convex programming (4.8), especially for RPCA.

Table 5.3: Numerical results of RPCA with synthetic data sets.

$(n, r)$	Methods	$\text{rank}(\hat{X})$	$\ \hat{Z}\ _0$	<b>Rerr</b>	Iter.	Time
(256, 13)	PDHG	13	6528	$6.3011 \times 10^{-4}$	146	1.13
	GRPDA	13	6518	$4.6516 \times 10^{-4}$	138	1.06
	SPIDA	13	6523	$6.3007 \times 10^{-4}$	112	0.82
(512, 26)	PDHG	26	26143	$1.2775 \times 10^{-4}$	127	4.73
	GRPDA	26	26151	$1.1530 \times 10^{-4}$	119	4.50
	SPIDA	26	26128	$1.7869 \times 10^{-4}$	86	3.23
(1024, 51)	PDHG	51	104677	$7.0827 \times 10^{-5}$	78	14.32
	GRPDA	51	104725	$5.0445 \times 10^{-5}$	87	15.48
	SPIDA	51	104677	$7.0421 \times 10^{-5}$	61	10.88
(2048, 102)	PDHG	102	419106	$2.4605 \times 10^{-5}$	80	269.52
	GRPDA	102	419221	$1.7315 \times 10^{-5}$	105	378.50
	SPIDA	102	419106	$2.4006 \times 10^{-5}$	67	205.77
(2560, 128)	PDHG	128	654970	$1.6256 \times 10^{-5}$	93	299.85
	GRPDA	128	655097	$1.5364 \times 10^{-5}$	124	383.87
	SPIDA	128	654965	$1.5685 \times 10^{-5}$	80	248.64

Table 5.4: Numerical results of RPCA with real-world data sets.

$(m, n)$	Methods	<b>Obj</b>	<b>Err</b>	Iter.	Time
Shoppingmall (81920, 200)	PDHG	3262.1	$3.0359 \times 10^{-3}$	58	33.72
	GRPDA	3275.2	$2.3097 \times 10^{-3}$	60	30.15
	SPIDA	3267.7	$2.6796 \times 10^{-3}$	49	28.52
Lobby (20480, 200)	PDHG	969.65	$4.1045 \times 10^{-3}$	128	13.53
	GRPDA	977.38	$2.9694 \times 10^{-3}$	127	13.66
	SPIDA	973.61	$3.5057 \times 10^{-3}$	109	11.58
Hall airport (25344, 200)	PDHG	2178.5	$3.6816 \times 10^{-3}$	66	10.34
	GRPDA	2188.3	$2.9182 \times 10^{-3}$	68	8.98
	SPIDA	2183.2	$3.2527 \times 10^{-3}$	57	7.50



Fig. 5.3. Background and foreground separations of surveillance videos. Three lines of pictures from top to bottom correspond to the Hall airport, Lobby and Shoppingmall in the video. The first column corresponds to the 100-th frame of the video. The second and third columns are the results extracted by PDHG. The fourth and fifth columns are the results extracted by PDHG. The last two columns are the results obtained by our SPIDA.

### 5.3. Image restoration

In this subsection, we apply the proposed SPIDA to solve an image restoration model with pixels constraints introduced in [22]. Such a model takes the form

$$\min_{x \in \mathbb{B}} \left\{ \|Dx\|_1 + \frac{\lambda}{2} \|Kx - b\|^2 \right\}, \quad (5.5)$$

where  $\mathbb{B}$  is a box area characterizing the pixels of an image (indeed,  $\mathbb{B} = [0, 1]$  and  $\mathbb{B} = [0, 255]$  if the image are double precision and 8-bit gray scale, respectively);  $\lambda$  is a positive trade-off parameter between the data-fidelity and regularization terms;  $D := (\partial_1, \partial_2)$  denotes the gradient operator with  $\partial_1$  and  $\partial_2$  being the discretized derivatives in the horizontal and vertical directions, respectively;  $K$  is the matrix representation of a blur operator and  $b$  is a corrupted image with additive noise. Clearly, model (5.5) is equivalent to the following saddle point problem:

$$\min_{x \in \mathbb{B}} \max_{y \in \mathbb{B}_\infty} \left\{ \langle Dx, y \rangle + \frac{\lambda}{2} \|Kx - b\|^2 \right\}, \quad (5.6)$$

where

$$\mathbb{B}_\infty := \{y \mid \|y\|_\infty \leq 1\}.$$

Consequently, when applying PDHG (1.3) to (5.6), the iterative scheme is specified as

$$\begin{cases} y^{k+1} = \arg \max_{y \in \mathbb{B}_\infty} \left\{ \langle Dx^k, y \rangle - \frac{\gamma}{2} \|y - y^k\|^2 \right\} \equiv \Pi_{\mathbb{B}_\infty}(y^k + \gamma^{-1} Dx^k), \\ x^{k+1} = \arg \min_{x \in \mathbb{B}} \left\{ \frac{\lambda}{2} \|Kx - b\|^2 + \langle Dx, 2y^{k+1} - y^k \rangle + \frac{\mu}{2} \|x - x^k\|^2 \right\}, \end{cases} \quad (5.7)$$

where the updating order of  $x$  and  $y$  is exchanged since the  $y$ -subproblem is simpler than the  $x$ -part, while the appearance of the deblurring matrix  $K$  makes  $x$ -subproblem relatively difficult without a closed-form solution. In this case, we employ the projected Barzilai-Borwein method in [17] and allow a maximal number of 50 for the inner loop to find an approximate solution of the  $x$ -subproblem in (5.7). To apply our SPIDA to (5.6), we consider two choices on the Bregman kernel functions:

$$(i) \quad \psi(x) = \frac{1}{2}\|x\|^2 \text{ and } \phi(y) = \frac{1}{2}\|y\|^2,$$

$$(ii) \quad \psi(x) = \frac{1}{2}\|x\|_{(I-\lambda\mu^{-1}K^\top K)}^2 \text{ and } \phi(y) = \frac{1}{2}\|y\|^2.$$

In what follows, we denote our SPIDA equipped with the above two kernel functions by SPIDA-I and SPIDA-II, respectively. As a consequence, SPIDA-I and SPIDA-II are specified as

$$\begin{aligned} \text{(SPIDA-I)} \quad & \begin{cases} \tilde{y}^{k+1} = \Pi_{\mathbb{B}_\infty}(y^k + \gamma^{-1}Dx^k), \\ x^{k+1} = \arg \min_{x \in \mathbb{B}} \left\{ \frac{\lambda}{2}\|Kx - b\|^2 + \langle Dx, \tilde{y}^{k+1} \rangle + \frac{\mu}{2}\|x - x^k\|^2 \right\}, \\ y^{k+1} = \Pi_{\mathbb{B}_\infty}(y^k + \gamma^{-1}Dx^{k+1}), \end{cases} \\ \text{(SPIDA-II)} \quad & \begin{cases} \tilde{y}^{k+1} = \Pi_{\mathbb{B}_\infty}(y^k + \gamma^{-1}Dx^k), \\ x^{k+1} = \Pi_{\mathbb{B}}(x^k - \lambda\mu^{-1}K^\top(Kx^k - b) - \mu^{-1}D^\top\tilde{y}^{k+1}), \\ y^{k+1} = \Pi_{\mathbb{B}_\infty}(y^k + \gamma^{-1}Dx^{k+1}). \end{cases} \end{aligned}$$

We also employ the projected Barzilai-Borwein method to find an approximate solution of the  $x$ -subproblem of SPIDA-I.

In our experiments, we consider several widely tested images as listed in the first column of Fig. 5.4. These images are corrupted by the blur operator with  $21 \times 21$  uniform kernels. Then, the blurred images are further corrupted by adding the zero-white Gaussian noise with standard deviation 0.002. The degraded images are listed in the second column of Fig. 5.4. We



Fig. 5.4. From left to right: the original image, observed image corrupted by blurring kernel size  $21 \times 21$ , recovered images by PDHG, GRPDA, SPIDA-I and SPIDA-II, respectively.

set  $\varepsilon = 10^{-4}$  in (5.6) as the stopping tolerance for all algorithms. The trade-off parameter  $\lambda$  is specified as  $\lambda = 1000$ . Moreover, we take  $(\gamma, \mu) = (0.016, 500)$  for PDHG,

$$(\psi, \tau, \sigma) = \left( 1.618, 0.016, \frac{500}{1.618} \right)$$

for GRPDA,  $(\gamma, \mu) = (0.016, 0.75 * 500)$  for SPIDA-I and  $(\gamma, \mu) = (0.016, 500)$  for SPIDA-II. Some preliminary numerical results are summarized in Table 5.5, which clearly shows that our SPIDA-I and SPIDA-II perform better than PDHG and GRPDA in terms of iterations and computing time. Note that SPIDA-I requires less iterations to achieve the almost same SNR (Signal-to-Noise Ratio, which is given by  $\text{SNR} = 20 \log_{10} \frac{\|b\|}{\|\tilde{b} - b\|}$ , where  $\tilde{b}$  is a reconstructed image and  $b$  is the ground truth) values than SPIDA-II. However, SPIDA-II takes much less computing time than SPIDA-I, thanks to the closed-form solutions of SPIDA-II. These results efficiently support that our symmetric idea is able to improve the numerical performance of the original primal-dual algorithm.

In Fig. 5.4, we list the recovered images by PDHG, GRPDA, SPIDA-I, and SPIDA-II from the third column to the last one, respectively. It can be seen that all methods are reliable to solve model (5.6). Finally, we list the evolution of SNR value with respect to computing time for solving image restoration model (5.6) in Fig. 5.5. These curves further show our SPIDA-II with easy subproblems has a superiority over the other algorithms in terms of computing time.

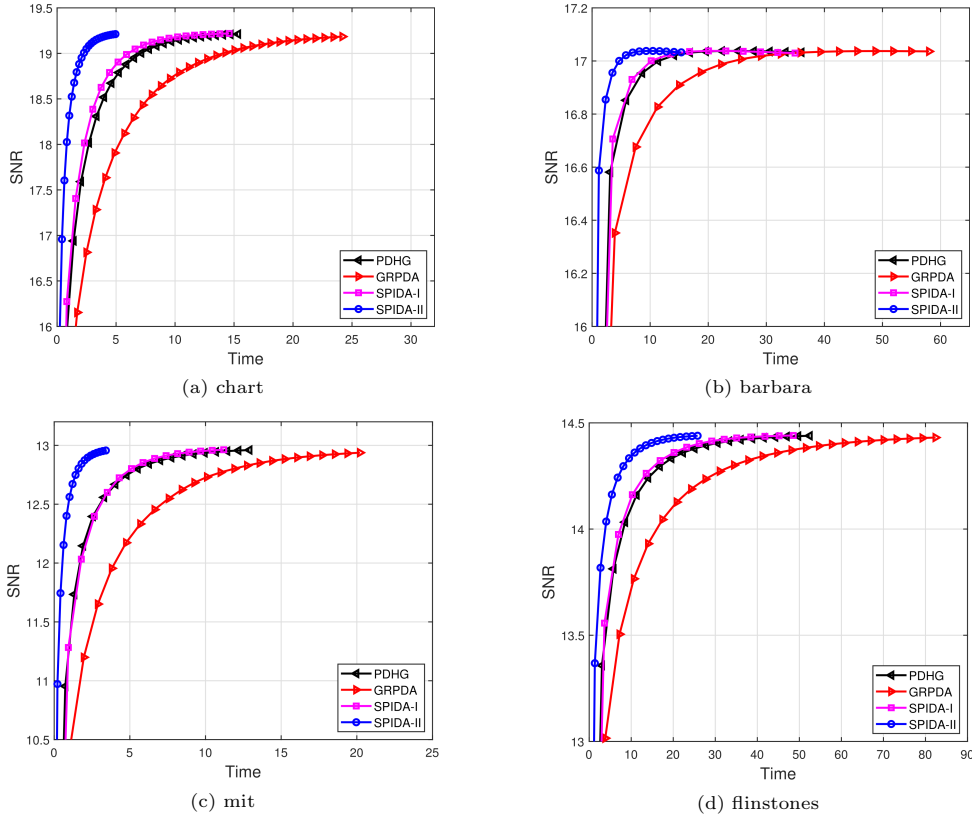


Fig. 5.5. Evolution of SNR value with respect to computing time for solving image restoration model (5.6).

Table 5.5: Numerical results for image restoration.

	PDHG	GRPDA	SPIDA-I	SPIDA-II
Image	Iter. / Time / SNR	Iter. / Time / SNR	Iter. / Time / SNR	Iter. / Time / SNR
chart	1476 / 16.37 / 19.214	1890 / 25.55 / 19.187	1217 / 15.12 / 19.220	1470 / 5.00 / 19.213
barbara	793 / 37.37 / 17.032	1035 / 59.67 / 17.036	677 / 36.64 / 17.029	792 / 15.73 / 17.032
mit	1080 / 13.11 / 12.960	1433 / 21.94 / 12.942	886 / 12.22 / 12.965	1078 / 3.81 / 12.960
flinstones	1141 / 54.02 / 14.439	1509 / 83.65 / 14.431	937 / 49.62 / 14.441	1171 / 23.47 / 14.440

## 6. Conclusions

In this paper, we proposed a new primal-dual algorithmic framework for convex-concave saddle point problems by applying the symmetric idea to compute the dual variable twice. Notice that a Bregman proximal regularization term is embedded in each subproblem, which is of benefit for us designing customized algorithms for some structured optimization problems. Moreover, we can gainfully understand the classical (linearized) augmented Lagrangian method and some parallel augmented Lagrangian-based splitting methods for linearly constrained convex optimization. A series of experiments demonstrate that our new algorithm works better than the other two compared methods as long as the dual subproblem (i.e.  $y$ -subproblem) is easy enough with cheap computational cost. In the future, we will consider some acceleration techniques such as extrapolation on the method. Besides, we notice that all subproblems of our algorithm are required to be solved exactly, which is expensive or impossible in some cases (e.g. see experiments in image restoration). Therefore, designing a practical inexact version of the proposed algorithm is also one of our future concerns.

**Acknowledgments.** The authors are grateful to the anonymous referees for their close reading, insightful comments, and valuable suggestions, which greatly help us improve the quality of this paper. Moreover, the first author would like to thank Professor Min Yan for bringing his attention to [15].

This research was supported in part by the National Natural Science Foundation of China (Grant Nos. 12371303, 11901294), by the Zhejiang Provincial Natural Science Foundation of China (Grant No. LZ24A010001), and by the Ningbo Natural Science Foundation (Grant No. 2023J014).

## References

- [1] K. Arrow, L. Hurwicz, and H. Uzawa, Studies in Linear and Non-Linear Programming, in: *Stanford Mathematical Studies in the Social Science*, Vol. II, Stanford University Press, 1958.
- [2] H. Bauschke, J. Borwein, and P. Combettes, Bregman monotone optimization algorithms, *SIAM J. Optim.*, **42** (2003), 596–636.
- [3] A. Beck, *First-Order Methods in Optimization*, SIAM, 2017.
- [4] S. Bonettini and V. Ruggiero, An alternating extragradient method for total variation based image restoration from Poisson data, *Inverse Problems*, **27** (2011), 9.
- [5] S. Bonettini and V. Ruggiero, On the convergence of primal-dual hybrid gradient algorithms for total variation image restoration, *J. Math. Imaging Vision*, **44** (2012), 236–253.

- [6] S. Bonettini and V. Ruggiero, An alternating extragradient method with non Euclidean projections for saddle point problems, *Comput. Optim. Appl.*, **59** (2014), 511–540.
- [7] L. Brègman, Relaxation method for finding a common point of convex sets and its application to optimization problems, *Dokl. Akad. Nauk*, **171** (1966), 1019–1022.
- [8] X. Cai, D. Han, and L. Xu, An improved first-order primal-dual algorithm with a new correction step, *J. Global Optim.*, **57** (2013), 1419–1428.
- [9] E. Candès, X. Li, Y. Ma, and J. Wright, Robust principal component analysis?, *J. ACM*, **58** (2011), 11.
- [10] A. Chambolle and T. Pock, A first-order primal-dual algorithm for convex problems with applications to imaging, *J. Math. Imaging Vision*, **40** (2011), 120–145.
- [11] A. Chambolle and T. Pock, An introduction to continuous optimization for imaging, *Acta Numer.*, **25** (2016), 161–319.
- [12] A. Chambolle and T. Pock, On the ergodic convergence rates of a first-order primal-dual algorithm, *Math. Program.*, **159** (2016), 253–287.
- [13] X. Chang and J. Yang, A golden ratio primal-dual algorithm for structured convex optimization, *J. Sci. Comput.*, **87**:2 (2021), 47.
- [14] X. Chang, J. Yang, and H. Zhang, Golden ratio primal-dual algorithm with linesearch, *SIAM J. Optim.*, **32** (2022), 1584–1613.
- [15] P. Chen, J. Huang, and X. Zhang, A primal-dual fixed point algorithm for multi-block convex minimization, *J. Comput. Math.*, **34** (2016), 723–738.
- [16] L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi, Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists, *SIAM Rev.*, **65**:2 (2023), 375–435.
- [17] Y. Dai and R. Fletcher, Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming, *Numer. Math.*, **100** (2005), 21–47.
- [18] D. Donoho, Compressed sensing, *IEEE Trans. Inform. Theory*, **52** (2006), 1289–1306.
- [19] E. Esser, X. Zhang, and T. Chan, A general framework for a class of first-order primal-dual algorithms for convex optimization in imaging sciences, *SIAM J. Imaging Sci.*, **3** (2010), 1015–1046.
- [20] A. Fisher, Comments on: Critical Lagrange multipliers: What we currently know about them, how they spoil our lives, and what we can do about it, *TOP*, **23** (2015), 27–31.
- [21] D. Han, H. He, and L. Xu, A proximal parallel splitting method for minimizing sum of convex functions, *J. Comput. Appl. Math.*, **256** (2014), 36–51.
- [22] D. Han, H. He, H. Yang, and X. Yuan, A customized Douglas-Rachford splitting algorithm for separable convex minimization with linear constraints, *Numer. Math.*, **127** (2014), 167–200.
- [23] D. Han, X. Yuan, and W. Zhang, An augmented-Lagrangian-based parallel splitting method for separable convex minimization with applications to image processing, *Math. Comput.*, **83** (2014), 2263–2291.
- [24] B. He, Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities, *Comput. Optim. Appl.*, **42** (2009), 195–212.
- [25] B. He, L. Hou, and X. Yuan, On full Jacobian decomposition of the augmented Lagrangian method for separable convex programming, *SIAM J. Optim.*, **25**:4 (2015), 2274–2312.
- [26] B. He, F. Ma, S. Xu, and X. Yuan, A generalized primal-dual algorithm with improved convergence condition for saddle point problems, *SIAM J. Imaging Sci.*, **15** (2022), 1157–1183.
- [27] B. He, F. Ma, and X. Yuan, An algorithmic framework of generalized primal-dual hybrid gradient methods for saddle point problems, *J. Math. Imaging Vision*, **58**:2 (2017), 279–293.
- [28] B. He, H. Xu, and X. Yuan, On the proximal Jacobian decomposition of ALM for multiple-block separable convex minimization problems and its relationship to ADMM, *J. Sci. Comput.*, **66** (2016), 1204–1217.
- [29] B. He, S. Xu, and X. Yuan, On convergence of the Arrow-Hurwicz method for saddle point

- problems, *J. Math. Imaging Vision*, **64** (2022), 662–671.
- [30] B. He, Y. You, and X. Yuan, On the convergence of primal dual hybrid gradient algorithm, *SIAM J. Imaging Sci.*, **7** (2015), 2526–2537.
  - [31] B. He and X. Yuan, Convergence analysis of primal-dual algorithms for a saddle-point problem: From contraction perspective, *SIAM J. Imaging Sci.*, **5** (2012), 119–149.
  - [32] B. He and X. Yuan, Balanced augmented Lagrangian method for convex programming, *arXiv: 2108.08554*, 2021.
  - [33] H. He, J. Desai, and K. Wang, A primal dual prediction correction algorithm for saddle point optimization, *J. Global Optim.*, **66** (2016), 573–583.
  - [34] L. Hou, H. He, and J. Yang, A partially parallel splitting method for multiple-block separable convex programming with applications to robust PCA, *Comput. Optim. Appl.*, **63**:1 (2016), 273–303.
  - [35] A. Izmailov and M. Solodov, Critical Lagrange multipliers: What we currently know about them, how they spoil our lives, and what we can do about it, *TOP*, **23** (2015), 1–26.
  - [36] F. Jiang, X. Cai, Z. Wu, and D. Han, Approximate first-order primal-dual algorithms for saddle point problems, *Math. Comput.*, **90** (2021), 1227–1262.
  - [37] F. Jiang, Z. Wu, X. Cai, and H. Zhang, A first-order inexact primal-dual algorithm for a class of convex-concave saddle point problems, *Numer. Algorithms*, **88** (2021), 1109–1136.
  - [38] F. Jiang, Z. Zhang, and H. He, Solving saddle point problems: A landscape of primal-dual algorithm with larger stepsizes, *J. Global Optim.*, **85** (2023), 821–846.
  - [39] N. Komodakis and J.C. Pesquet, Playing with duality an overview of recent primal dual approaches for solving large scale optimization problems, *IEEE Signal Process Mag.*, **32**:6 (2015), 31–54.
  - [40] G.M. Korpelevich, The extragradient method for finding saddle points and other problems, *Ekonom. Mat. Metody*, **12** (1976), 747–756.
  - [41] Y. Li and M. Yan, On the improved conditions for some primal-dual algorithms, *J. Sci. Comput.*, **99** (2024), 74.
  - [42] Z. Li and M. Yan, New convergence analysis of a primal-dual algorithm with large stepsizes, *Adv. Comput. Math.*, **47**:1 (2021), 9.
  - [43] Y. Ma, X. Cai, B. Jiang, and D. Han, Understanding the convergence of the preconditioned PDHG method: A view of indefinite proximal ADMM, *J. Sci. Comput.*, **94** (2023), 60.
  - [44] Y. Malitsky and T. Pock, A first-order primal-dual algorithm with linesearch, *SIAM J. Optim.*, **28**:1 (2018), 411–432.
  - [45] J. Martínez, Comments on: Critical Lagrange multipliers: What we currently know about them, how they spoil our lives, and what we can do about it, *TOP*, **23** (2015), 32–34.
  - [46] T. Möllenhoff, E. Strekalovskiy, M. Moeller, and D. Cremers, The primal-dual hybrid gradient method for semiconvex splittings, *SIAM J. Imaging Sci.*, **8** (2015), 827–857.
  - [47] B. Mordukhovich, Comments on: Critical Lagrange multipliers: What we currently know about them, how they spoil our lives, and what we can do about it, *TOP*, **23** (2015), 35–42.
  - [48] J. Moreau, Fonctions convexe dudual et points proximaux dans un espace Hilbertien, *C.R. Math.*, **255** (1962), 2897–2899.
  - [49] A. Nemirovski, Prox-method with rate of convergence  $O(1/t)$  for variational inequality with Lipschitz continuous monotone operators and smooth convex-concave saddle points problems, *SIAM J. Optim.*, **15** (2004), 229–251.
  - [50] N. Parikh and S. Boyd, Proximal algorithms, *Found. Trends Optim.*, **1** (2013), 123–231.
  - [51] L. Popov, A modification of the Arrow-Hurwicz method for search of saddle points, *Math. Notes*, **28**:5 (1980), 845–848.
  - [52] J. Rasch and A. Chambolle, Inexact first-order primal-dual algorithms, *Comput. Optim. Appl.*, **76**:2 (2020), 381–430.
  - [53] M. Razaviyayn, T. Huang, S. Lu, M. Nouiehed, M. Sanjabi, and M. Hong, Nonconvex min-max optimization: Applications, challenges, and recent theoretical advances, *IEEE Signal Process*

- Mag.*, **37**:5 (2020), 55–66.
- [54] D. Robinson, Comments on: Critical Lagrange multipliers: What we currently know about them, how they spoil our lives, and what we can do about it, *TOP*, **23** (2015), 43–47.
  - [55] M. Tao and X. Yuan, Recovering low-rank and sparse components of matrices from incomplete and noisy observations, *SIAM J. Optim.*, **21** (2011), 57–81.
  - [56] T. Valkonen, First-order primal-dual methods for nonsmooth non-convex optimisation, in: *Handbook of Mathematical Models and Algorithms in Computer Vision and Imaging: Mathematical Imaging and Vision*, Springer, (2021), 1–42.
  - [57] K. Wang, J. Desai, and H. He, A note on augmented Lagrangian-based parallel splitting method, *Optim. Lett.*, **9** (2015), 1199–1212.
  - [58] K. Wang and H. He, A double extrapolation primal-dual algorithm for saddle point problems, *J. Sci. Comput.*, **85**:3 (2020), 1–30.
  - [59] X. Wang, M. Hong, S. Ma, and Z. Luo, Solving multiple-block separable convex minimization problems using two-block alternating direction method of multipliers, *Pac. J. Optim.*, **11** (2015), 645–667.
  - [60] J. Yang and X. Yuan, Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization, *Math. Comput.*, **82** (2013), 301–329.
  - [61] W. Yang and D. Han, Linear convergence of the alternating direction method of multipliers for a class of convex optimization problems, *SIAM J. Numer. Anal.*, **54**:2 (2016), 625–640.
  - [62] H. Zhang, Extragradient and extrapolation methods with generalized Bregman distances for saddle point problems, *Oper. Res. Lett.*, **50** (2022), 329–334.
  - [63] M. Zhu and T. Chan, *An Efficient Primal-dual Hybrid Gradient Algorithm for Total Variation Image Restoration*, UCLA CAM Reports 08-34, 2008. <https://ww3.math.ucla.edu/camreport/cam08-34.pdf>