Journal of Scientific Computing (2024) 98:9
https://doi.org/10.1007/s10915-023-02397-x

A Partially Inertial Customized Douglas–Rachford Splitting Method for a Class of Structured Optimization Problems

Yunfei Qu[1] · Hongjin He[2] · Deren Han[1]

Received: 2 April 2023 / Revised: 12 August 2023 / Accepted: 25 October 2023 /
Published online: 22 November 2023

Abstract
In this paper, we are concerned with a class of structured optimization problems frequently arising from image processing and statistical learning, where the objective function is the sum of a quadratic term and a nonsmooth part, and the constraint set consists of a linear equality constraint and two simple convex sets in the sense that projections onto simple sets are easy to compute. To fully exploit the quadratic and separable structure of the problem under consideration, we accordingly propose a partially inertial Douglas–Rachford splitting method. It is noteworthy that our algorithm enjoys easy subproblems for the case where the underlying two simple convex sets are not the whole spaces. Theoretically, we establish the global convergence of the proposed algorithm under some mild conditions. A series of computational results on the constrained Lasso and constrained total-variation (TV) based image restoration demonstrate that our proposed method is competitive with some state-of-the-art first-order solvers.

Keywords Douglas–Rachford splitting method · Alternating direction method of multipliers · Structured optimization · Lasso · Image restoration

1 Introduction

We consider a class of structured optimization problems as follows:

$$\min_{x,y} \quad f(x) + g(y)$$

✉ Deren Han
handr@buaa.edu.cn

Yunfei Qu
yunfei_math@hotmail.com

Hongjin He
hehongjin@nbu.edu.cn

[1] LMIB of the Ministry of Education, School of Mathematical Sciences, Beihang University, Beijing 100191, China

[2] School of Mathematics and Statistics, Ningbo University, Ningbo 315211, China

$$\begin{aligned} \text{s.t.} \quad & Ax + By = b, \\ & x \in \mathcal{X}, \, y \in \mathcal{Y}, \end{aligned} \tag{1.1}$$

where $f(x)$ is in particular specified as a quadratic function, i.e., $f(x) = \frac{1}{2}\|Kx - d\|^2$ with $K \in \mathbb{R}^{m_1 \times n_1}$ and $d \in \mathbb{R}^{m_1}$; $g : \mathbb{R}^{n_2} \to \mathbb{R}$ is a continuous closed convex (usually assumed to be nonsmooth) function; $A \in \mathbb{R}^{m_2 \times n_1}$ and $B \in \mathbb{R}^{m_2 \times n_2}$ are given matrices; $b \in \mathbb{R}^{m_2}$ is a given vector; both $\mathcal{X} \subset \mathbb{R}^{n_1}$ and $\mathcal{Y} \subset \mathbb{R}^{n_2}$ are simple closed convex sets in the sense that projections onto them are very easy (e.g., nonnegative orthant, spheroidal or box areas). Throughout this paper, we assume that the solution set of problem (1.1) is nonempty. Besides, it is noteworthy that although we restrict the later theoretical discussion to the case of model (1.1) with vector variables, all our results actually are applicable to the case with matrix variables.

The main reason that we focus on model (1.1) with a specific quadratic term is its fruitful applications in the areas of image processing, machine learning, and statistical analysis. Generally speaking, the quadratic part $f(x)$ often serves as a data-fidelity term and the nonsmooth part $g(y)$ usually represents a regularization term to promote some special structure (e.g., sparsity and low-rankness) of solutions. Below, we just list three motivating examples which are special cases of model (1.1).

- *Constrained TV-based image restoration*. The pixels constrained TV-based image restoration model (see [29, 39]) takes the form of

$$\min_{x \in \mathcal{B}} \left\{ \frac{1}{2}\|Kx - d\|^2 + \mu \| |\boldsymbol{D}x| \|_1 \right\}, \tag{1.2}$$

where d is an observed image; $K \in \mathbb{R}^{n \times n}$ represents a degraded operator with $n = n_1 \times n_2$ being the total number of pixels; $\boldsymbol{D}$ is the discrete gradient operator (see [42] for more details); $\mathcal{B} = [\mathbf{l}, \mathbf{u}]$ is a box area in $\mathbb{R}^n$ (e.g., $\mathcal{B} = [0, 1]^n$ and $[0, 255]^n$ for double precision and 8-bit gray-scale images, respectively); μ is a positive trade-off parameter between the data-fidelity and regularization term. Clearly, introducing an auxiliary variable y to the TV regularization term immediately leads to the following reformulation:

$$\min_{x,y} \left\{ \frac{1}{2}\|Kx - d\|^2 + \mu \| |y| \|_1 \;\middle|\; \boldsymbol{D}x - y = 0, \; x \in \mathcal{B}, \; y \in \mathbb{R}^n \times \mathbb{R}^n \right\}, \tag{1.3}$$

which is a special case of the generic model (1.1) with settings $g(y) = \mu \| |y| \|_1$, $A = \boldsymbol{D}$, $B = -I$, $\mathcal{X} = \mathcal{B}$, and $\mathcal{Y} = \mathbb{R}^n \times \mathbb{R}^n$.

- *Constrained Lasso*. The constrained Lasso problem is an extension of the standard Lasso problem [43] with an extra constraint. Mathematically, it can be expressed as

$$\min_{x \in \mathcal{X}} \left\{ \frac{1}{2}\|Kx - d\|^2 + \mu \|x\|_1 \right\}, \tag{1.4}$$

where $d \in \mathbb{R}^l$ is an observed vector; $K \in \mathbb{R}^{l \times n}$ is a design or dictionary matrix; μ is a positive constant controlling the degree of sparsity and $\mathcal{X} \subset \mathbb{R}^n$ is an extra constraint. For example, the nonnegative constraint, the ball constraint, the sum-to-zero constraint and so on [25]. Such a model has been applied to hyperspectral unmixing [36], classification [17], economics [47], and face recognition [18]. Like model (1.2), we can also rewrite (1.4) as a special instance of (1.1) by introducing an additional variable y to the ℓ_1-norm regularization term, i.e.,

$$\min_{x,y} \left\{ \frac{1}{2}\|Kx - d\|^2 + \mu \|y\|_1 \;\middle|\; x - y = 0, \; x \in \mathcal{X}, \; y \in \mathbb{R}^n \right\}. \tag{1.5}$$

- *Nuclear-norm-regularized least squares*. A fundamental matrix completion model is the so-called nuclear-norm-regularized least squares problem (e.g., see [16, 37, 44]), which has the form of

$$\min_{X\in\mathcal{B}}\left\{\frac{1}{2}\|\mathcal{K}(X)-M\|_F^2+\mu\|X\|_*\right\}, \tag{1.6}$$

  where $\mathcal{K}$ can be regarded as a linear sampling operator; M is an observed incomplete or corrupted matrix; $\|\cdot\|_F$ denotes the standard Frobenius norm; $\|\cdot\|_*$ is the nuclear norm (i.e., the sum of all singular values) promoting the low-rankness of a matrix; $\mu > 0$ is a trade-off parameter; $\mathcal{B}$ is a box area (e.g., $[1, 5]$ and $[-10, 10]$, respectively) usually serving as a range of ratings in recommendation systems such as the classic Netflix problem [5] and Jester Joke problem [27]. Like models (1.2) and (1.4), we reformulate model (1.6) as

$$\min_{X,Y}\left\{\frac{1}{2}\|\mathcal{K}(X)-M\|_F^2+\mu\|Y\|_*\ \mid\ X-Y=0,\ X\in\mathcal{B},\ Y\in\mathbb{R}^{m\times n}\right\}. \tag{1.7}$$

  It is clear that (1.7) is also a special case of model (1.1) with matrix variables.

To solve model (1.1), one of the most popular solvers is the so-called alternating direction method of multipliers (ADMM) proposed in [24, 26], which, for given the k-th iterate (y^k, λ^k), updates the next iterate via

$$\left\{\begin{array}{ll} x^{k+1}\in\arg\min\limits_{x\in\mathcal{X}}\left\{f(x)+\dfrac{\beta}{2}\left\|Ax+By^k-b-\dfrac{1}{\beta}\lambda^k\right\|^2\right\}, & \text{(1.8a)}\\ y^{k+1}\in\arg\min\limits_{y\in\mathcal{Y}}\left\{g(y)+\dfrac{\beta}{2}\left\|Ax^{k+1}+By-b-\dfrac{1}{\beta}\lambda^k\right\|^2\right\}, & \text{(1.8b)}\\ \lambda^{k+1}=\lambda^k-\gamma\beta(Ax^{k+1}+By^{k+1}-b), & \text{(1.8c)}\end{array}\right.$$

where $\lambda^k\in\mathbb{R}^{m_2}$ is the Lagrangian multiplier, $\beta > 0$ is a penalty parameter, and $\gamma\in(0, (1+\sqrt{5})/2)$ is a relaxation factor. Actually, the great success of ADMM in statistical learning is owing to its easy subproblems when applying to some nonsmooth optimization models without extra simple convex sets, such as Lasso and basis pursuit (see [10] and references therein for more applications). In other words, ADMM efficiently exploits the great virtue of some nonsmooth functions being simple enough in the sense that their associated proximal operators enjoy closed-form solutions. However, when revisiting the iterative scheme of ADMM for (1.1), we observe that both subproblems, i.e., (1.8a) and (1.8b), are constrained optimization problems due to the appearance of $\mathcal{X}$ and $\mathcal{Y}$. Consequently, (1.8a) and (1.8b) have no closed-form solutions, even when f and g are simple enough to possess explicit proximal operators (e.g., $\|\cdot\|_1$ and $\|\cdot\|_*$), and $\mathcal{X}$ and $\mathcal{Y}$ are simple enough with cheap projections (e.g., nonnegative orthant and box areas). Indeed, the appearance of general matrices A and B (i.e., A and B are not identity matrices) also makes both subproblems (1.8a) and (1.8b) lose their closed-form solutions for some cases such as $f(\cdot)=\|\cdot\|_1$ and $f(\cdot)=\|\cdot\|_*$. In these situations, we usually need to call some optimization solvers to find approximate solutions of (1.8a) and (1.8b), thereby causing more computing time for large-scale problems than those methods with easy subproblems. e.g., see numerical results in [29, 34, 35, 46]. We here emphasize that there exist closed-form solutions to the subproblems of linearized ADMM when the objective functions are differentiable. However, there are many cases where the objective functions are not necessarily differentiable. Therefore, to make (1.8a) and (1.8b) easy to be solved, Han et al. [29] introduced a so-named customized

Douglas–Rachford splitting method (CDRSM) to find solutions of (1.1) with general convex objective functions, where they showed that such an algorithm can successfully circumvent constrained subproblems so that it has easier subproblems than ADMM in some cases. A series of numerical experiments in [29, 33] verified that CDRSM performs better than ADMM for some constrained optimization problems.

To accelerate speed of first-order optimization methods, inertial (a.k.a., extrapolation) technique originally proposed in [40] is one of the most popular strategies in the literature. Recently, many efficient inertial-type splitting algorithms have been developed, e.g., inertial forward-backward algorithm [3, 9, 38], inertial Peaceman-Rachford splitting algorithm [20, 21], inertial ADMM [7, 15, 28], inertial Douglas–Rachford splitting method [2, 8], to name just a few. Like ADMM (1.8), however, these accelerated splitting algorithms still fail to maximally exploit the favorable structure (e.g., simple convex sets $\mathcal{X}$ and $\mathcal{Y}$ and separability of the objective) of (1.1). For example, when directly applying the inertial Douglas–Rachford splitting method [2, 8], we can see from [29] that both variables x and y are treated as a one-block variable, thereby resulting in a coupled subproblem so that the algorithm is not easy enough to implement. Therefore, we aim to introduce an implementable Douglas–Rachford splitting method equipped with an inertial acceleration step for solving model (1.1). Generally speaking, the quadratic part $f(x)$ often serves as a data-fidelity term, which usually dominates the objective function of (1.1) in many real-world applications, since the trade-off parameter in $g(y)$ (e.g., the μ in (1.2), (1.4) and (1.6)) usually is small. In other words, the quadratic term plays a more important role in the process of solving model (1.1). Accordingly, we in this paper propose a partially inertial Douglas–Rachford splitting method for problem (1.1), where we only consider the inertial acceleration to the quadratic term related variable and Lagrangian multiplier. Since our algorithm is based on the CDRSM [29], our proposed inertial variant of CDRSM fully exploits the separable structure of the objective function, thereby enjoying much easier subproblems than ADMM and the original Douglas–Rachford splitting method (DRSM). Moreover, some numerical experiments on constrained Lasso and image deblurring show that our proposed algorithm has expected promising numerical behaviors.

The rest of this paper is organized as follows. In Sect. 2, we recall some basic results of maximal monotone operators and list two equivalent reformulations of model (1.1). In Sect. 3, we first present our new algorithm (see Algorithm 1) and then establish the global convergence under some mild conditions. In Sect. 4, we conduct some numerical experiments to verify that the embedded inertial technique can really accelerate the CDRSM for solving problem (1.1). Finally, we complete this paper with drawing some concluding remarks in Sect. 5.

2 Preliminaries

In this section, we summarize some basic concepts and well-known results that will be used in the subsequent analysis. Also, we recall two equivalent reformulations of problem (1.1).

Let $\mathbb{R}^n$ be an n-dimensional Euclidean space equipped with the standard Euclidean norm, i.e., $\|x\| = \sqrt{x^\top x}$ for any $x \in \mathbb{R}^n$, where the superscript '$^\top$' represents the transpose of a vector or matrix. For any matrix M, we denote $\|M\|$ as its matrix 2-norm.

Let $P_\Omega[\cdot]$ denote the projection operator onto Ω under the Euclidean norm, i.e.,

$$P_\Omega[x] := \arg\min\{\|x - y\| \mid y \in \Omega\}. \tag{2.1}$$

The following properties with respect to the projection operator are fundamental in the literature and their proof can be found in [6, 12, 22].

Lemma 1 *Let Ω be a nonempty closed convex subset of $\mathbb{R}^n$ and $P_\Omega[\cdot]$ be the projection operator onto Ω defined in (2.1). Then we have*

$$(u - P_\Omega[u])^\top (v - P_\Omega[u]) \leq 0, \quad \forall u \in \mathbb{R}^n, \quad \forall v \in \Omega. \tag{2.2}$$

Lemma 2 *For any $x, y \in \mathbb{R}^n$, and $\tau \in \mathbb{R}$, we have the following identity*

$$\|\tau x + (1-\tau)y\|^2 = \tau\|x\|^2 + (1-\tau)\|y\|^2 - \tau(1-\tau)\|x-y\|^2. \tag{2.3}$$

Hereafter, we recall a lemma which will be used for global convergence analysis.

Lemma 3 *([11]) Let $\{x^k\}$ be a sequence in $\mathbb{R}^n$ and $\mathcal{D}$ be a nonempty set of $\mathbb{R}^n$. Suppose that*

(i). $\lim_{k\to+\infty} \|x^k - x\|$ *exists for all $x \in \mathcal{D}$;*
(ii). *every cluster point of $\{x^k\}$ belongs to $\mathcal{D}$.*

Then, the sequence $\{x^k\}$ converges to a point in $\mathcal{D}$.

Definition 1 A set-valued map $T : \mathbb{R}^n \to 2^{\mathbb{R}^n}$ is said to be *monotone* if

$$(u_1 - u_2)^\top (v_1 - v_2) \geq 0, \quad \forall u_1, \ u_2 \in \mathbb{R}^n, \ v_1 \in T(u_1), \ v_2 \in T(u_2).$$

Definition 2 For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, its *subdifferential* at x is the set-valued operator given by

$$\partial f(x) = \{\xi \mid f(y) \geq f(x) + \xi^\top (y - x), \quad \forall y \in \mathbf{dom} f\},$$

where ξ is called a subgradient of f at x, and $\mathbf{dom} f$ is the effective domain of f.

Below, we recall two equivalent reformulations for problem (1.1) that will be useful for algorithmic design and convergence analysis. Let $\lambda \in \mathbb{R}^{m_2}$ be the Lagrange multiplier associated to the linear constraints of (1.1). Then, the classical variational inequality (VI for short) characterization of (1.1) can be stated as finding a vector $w^* \in \mathcal{W}$ such that

$$(w - w^*)^\top F(w^*) \geq 0, \quad \forall w \in \mathcal{W}, \tag{2.4a}$$

where $w := (x, y, \lambda), \ \mathcal{W} := \mathcal{X} \times \mathcal{Y} \times \mathbb{R}^{m_2}$ and

$$F(w) := \{(\nabla f(x) - A^\top \lambda, \ \zeta - B^\top \lambda, \ Ax + By - b) \mid \zeta \in \partial g(y)\}. \tag{2.4b}$$

Clearly, it is well-known from [41] that the underlying mapping $F(w)$ defined in VI (2.4) is monotone. Moreover, note that the nonempty assumption on the solution set of problem (1.1) implies the nonemptiness of the solution set (denoted by $\mathcal{W}^*$ throughout this paper) of VI (2.4). Alternatively, it also follows from [23, page 3] that VI (2.4) can be further reformulated as finding a zero point of the sum of two maximal monotone operators, i.e., finding $w^* \in \mathcal{W}$ such that

$$0 \in F(w^*) + \mathcal{N}_\mathcal{W}(w^*), \tag{2.5}$$

where $\mathcal{N}_\mathcal{W}(\cdot)$ is the normal cone of $\mathcal{W}$ defined as

$$\mathcal{N}_\mathcal{W}(w) := \begin{cases} \{u \mid (v - w)^\top u \leq 0, \ \ \forall v \in \mathcal{W}\}, & \text{if } w \in \mathcal{W}, \\ \varnothing, & \text{otherwise.} \end{cases}$$

For any $\beta > 0$, we let

$$\mathscr{E}(w, \beta) = w - P_\mathcal{W}[w - \beta F(w)]. \tag{2.6}$$

Then, we have the following proposition.

Proposition 1 *The vector $w^* \in \mathcal{W}^*$ is a solution of VI (2.4) if and only if $\|\mathscr{E}(w^*, \beta)\| = 0$ for any $\beta > 0$.*

3 The Algorithm and Its Convergence Analysis

In this section, we first elaborate our proposed partially inertial CDRSM for solving (1.1). Then, we prove its global convergence under some mild conditions.

3.1 Algorithm

Before stating our new algorithm, we first recall the general scheme of DRSM for solving (2.5) (see [30, 32, 45]), which, for given the k-th iterate w^k, reads as

$$w^{k+1} = (I + \beta F)^{-1}\{w^k + \beta F(w^k) - \gamma \mathscr{E}(w^k, \beta)\}, \tag{3.1}$$

where $(I + \beta F)^{-1}$ represents the standard resolvent operator and $\gamma \in (0, 2)$ is a relaxation factor. Accordingly, by using the notations of F and w given in (2.4b), the iterative scheme (3.1) can be specified as finding $w^{k+1} \in \mathcal{W}$ and $\zeta^{k+1} \in \partial g(y^{k+1})$ such that

$$\begin{cases} x^{k+1} = \arg\min\limits_{x \in \mathbb{R}^{n_1}} \left\{ f(x) + \dfrac{1}{2\beta} \left\| x - \left[x^k + \beta \nabla f(x^k) - \beta A^\top(\lambda^k - \lambda^{k+1}) - \gamma \mathscr{E}_1(x^k, \lambda^k, \beta) \right] \right\|^2 \right\}, & (3.2a) \\ y^{k+1} = \arg\min\limits_{y \in \mathbb{R}^{n_2}} \left\{ g(y) + \dfrac{1}{2\beta} \left\| y - \left[y^k + \beta \zeta^k - \beta B^\top(\lambda^k - \lambda^{k+1}) - \gamma \mathscr{E}_2(y^k, \lambda^k, \beta) \right] \right\|^2 \right\}, & (3.2b) \\ \lambda^{k+1} = \lambda^k + \beta \left[A(x^k - x^{k+1}) + B(y^k - y^{k+1}) \right] - \gamma \mathscr{E}_3(x^k, y^k, \beta). & (3.2c) \end{cases}$$

Clearly, we can see from (3.2) that both (3.2a) and (3.2b) are unconstrained minimization problems, which are in general easier than ADMM's subproblems, i.e., (1.8a) and (1.8b), as long as the projections onto $\mathcal{X}$ and $\mathcal{Y}$ are simple enough with explicit representations. However, such an algorithm is not easy to implement due to the coupled variables $(x^{k+1}, y^{k+1}, \lambda^{k+1})$. In other words, the straightforward application of DRSM does not fully exploit the separability of the objective function. Actually, it can be easily seen from (3.2) that the main difficulty to implement such an algorithm is owing to the appearance of λ^{k+1} in (3.2a) and (3.2b). Hence, Han et al. [29] first made a prediction on λ, and slightly modified the updating schemes on x^{k+1}, y^{k+1} and λ^{k+1}. Consequently, they introduced the so-called CDRSM for (1.1). Inspired by the inertial acceleration idea, we in this paper propose a partially inertial CDRSM (denoted by ICDRSM) to accelerate the original CDRSM for solving (1.1). It is remarkable that our ICDRSM approaches the original CDRSM when the inertial parameters approaches zero.

To present our ICDRSM, for notational simplicity, we below denote

$$\bar{u}^k := (\tilde{x}^k, \bar{\lambda}^k, \beta), \quad \bar{v}^k := (y^k, \bar{\lambda}^k, \beta), \quad \bar{w}^k := (\tilde{x}^k, y^k, \bar{\lambda}^k, \beta), \tag{3.3}$$

and consequently,

$$\begin{cases} \mathscr{E}_x(\bar{u}^k) := \mathscr{E}_1(\tilde{x}^k, \bar{\lambda}^k, \beta), & (3.4a) \\ \mathscr{E}_y(\bar{v}^k) := \mathscr{E}_2(y^k, \bar{\lambda}^k, \beta), & (3.4b) \\ \mathscr{E}_\lambda(\bar{w}^k) := \tilde{\lambda}^k - \bar{\lambda}^k - \beta[A\mathscr{E}_x(\bar{u}^k) + B\mathscr{E}_y(\bar{v}^k)]. & (3.4c) \end{cases}$$

With the above preparations, details of the proposed ICDRSM for (1.1) can be described in Algorithm 1.

Algorithm 1 A partially inertial CDRSM for (1.1).

```
Choose starting point $(x^0, y^0, \lambda^0) = (x^1, y^1, \lambda^1) \in \mathcal{W}$. Set $\tau \in (0, 1)$, $\gamma \in \left(\frac{2\tau^2+2}{2\tau^2+\tau+1}, \frac{2\tau+2}{2\tau+1}\right)$ and
    $\beta \in (0, \frac{\sqrt{2}}{c})$ with $c := \max\{\|A\|, \|B\|\}$.
for $k = 1, 2, \ldots$ do
    Update $w^{k+1} \in \mathcal{W}$ via
        $\begin{pmatrix} \tilde{x}^k \\ \tilde{\lambda}^k \end{pmatrix} = \begin{pmatrix} x^k \\ \lambda^k \end{pmatrix} + \tau \begin{pmatrix} x^k - x^{k-1} \\ \lambda^k - \lambda^{k-1} \end{pmatrix};$   (3.5)
        $\bar{\lambda}^k = \tilde{\lambda}^k - \beta(A\tilde{x}^k + By^k - b);$   (3.6)
        $x^{k+1} = \arg\min_{x \in \mathbb{R}^{n_1}} \left\{ f(x) + \frac{1}{2\beta}\|x - \omega_x^k\|^2 \right\};$   (3.7)
        $y^{k+1} = \arg\min_{y \in \mathbb{R}^{n_2}} \left\{ g(y) + \frac{1}{2\beta}\|y - \omega_y^k\|^2 \right\};$   (3.8)
        $\lambda^{k+1} = \tilde{\lambda}^k - \gamma\alpha_k \mathscr{E}_\lambda(\bar{w}^k),$   (3.9)
    where $\omega_x^k := \tilde{x}^k + \beta\nabla f(\tilde{x}^k) - \gamma\alpha_k\mathscr{E}_x(\bar{u}^k)$, $\omega_y^k := y^k + \beta\zeta^k - \gamma\alpha_k\mathscr{E}_y(\bar{v}^k)$ with $\nabla f(\tilde{x}^k) = K^\top(K\tilde{x}^k - d)$, $\zeta^k \in \partial g(y^k)$ and
        $\alpha_k := \frac{\varphi(\bar{w}^k)}{\psi(\bar{w}^k)};$   (3.10)
        $\varphi(\bar{w}^k) := \|\mathscr{E}_x(\bar{u}^k)\|^2 + \|\mathscr{E}_y(\bar{v}^k)\|^2 + (\tilde{\lambda}^k - \bar{\lambda}^k)^\top \mathscr{E}_\lambda(\bar{w}^k);$   (3.11)
        $\psi(\bar{w}^k) := \|\mathscr{E}_x(\bar{u}^k)\|^2 + \|\mathscr{E}_y(\bar{v}^k)\|^2 + \|\mathscr{E}_\lambda(\bar{w}^k)\|^2.$   (3.12)
end for
```

3.2 Convergence Analysis

In this subsection, we establish the global convergence of Algorithm 1. For notational convenience, we let

$$z_1 := x + \beta\nabla f(x),\ z_2 := y + \beta\zeta,\ \mathbf{z} := \begin{pmatrix} z_1 \\ z_2 \\ \lambda \end{pmatrix} \text{ and } \mathscr{D}(\bar{w}^k) := \begin{pmatrix} \mathscr{E}_x(\bar{u}^k) \\ \mathscr{E}_y(\bar{v}^k) \\ \mathscr{E}_\lambda(\bar{w}^k) \end{pmatrix}, \tag{3.13}$$

where $\nabla f(x) = K^\top(Kx - d)$ and $\zeta \in \partial g(y)$. For convenience, we further denote

$$\tilde{z}_1 := \tilde{x} + \beta\nabla f(\tilde{x}),\ \tilde{\mathbf{z}} := \begin{pmatrix} \tilde{z}_1 \\ z_2 \\ \tilde{\lambda} \end{pmatrix}. \tag{3.14}$$

The following lemma clarifies the relationship between $\mathscr{D}(\bar{w}^k)$ and a solution of the VI problem (2.4).

Lemma 4 *If $\mathscr{D}(\bar{w}^k) = 0$ for any $\beta > 0$, then the vector $w^{k+1} = (x^{k+1}, y^{k+1}, \lambda^{k+1})$ is a solution of VI (2.4).*

Proof From the notation of $\mathscr{D}(\bar{w}^k)$ in (3.13), we know that $\mathscr{D}(\bar{w}^k) = 0$ implies

$$\|\mathscr{E}_x(\bar{u}^k)\| = \|\mathscr{E}_y(\bar{v}^k)\| = \|\mathscr{E}_\lambda(\bar{w}^k)\| = 0. \tag{3.15}$$

Then, by substituting (3.15) into (3.4) and combining with (3.6), we have

$$\mathscr{E}_3(\tilde{x}^k, y^k, \beta) = 0,$$

which immediately implies that

$$\mathscr{E}(\tilde{x}^k, y^k, \bar{\lambda}^k, \beta) = 0. \tag{3.16}$$

Hence, by invoking Proposition 1, the relation (3.16) means that $(\tilde{x}^k, y^k, \bar{\lambda}^k)$ is a solution of VI (2.4). Consequently, it follows from the optimality conditions of (3.7) and (3.8) and the update scheme (3.9) that $x^{k+1} = \tilde{x}^k$, $y^{k+1} = y^k$, and $\lambda^{k+1} = \tilde{\lambda}^k$, which, together with the fact that $(\tilde{x}^k, y^k, \bar{\lambda}^k)$ is a solution, means that $(x^{k+1}, y^{k+1}, \lambda^{k+1})$ is also a solution of VI (2.4). This completes the proof. □

Lemma 5 *Suppose that $w^* \in \mathcal{W}^*$ is an arbitrary solution of (2.4). Then the sequence $\{\tilde{\mathbf{z}}^k\}$ generated by Algorithm 1 satisfies*

$$(\tilde{\mathbf{z}}^k - \mathbf{z}^*)^\top \mathscr{D}(\bar{w}^k) \geq \varphi(\bar{w}^k), \quad k \geq 1,$$

where $\mathbf{z}^*$, $\tilde{\mathbf{z}}^k$, $\mathscr{D}(\bar{w}^k)$ and $\varphi(\bar{w}^k)$ are defined in (3.13),(3.14) and (3.11), respectively.

Proof Since $w^* = (x^*, y^*, \lambda^*) \in \mathcal{W}^*$ and $\zeta^* \in \partial g(y^*)$, it follows from (2.4) that

$$(x' - x^*)^\top \left(\nabla f(x^*) - A^\top \lambda^*\right) \geq 0, \quad \forall x' \in \mathcal{X}, \tag{3.17}$$

and

$$(y' - y^*)^\top \left(\zeta^* - B^\top \lambda^*\right) \geq 0, \quad \forall y' \in \mathcal{Y}.$$

Then, by setting $x' = P_{\mathcal{X}}[\tilde{x}^k - \beta(\nabla f(\tilde{x}^k) - A^\top \bar{\lambda}^k)] = \tilde{x}^k - \mathscr{E}_x(\bar{u}^k)$ in (3.17), we have

$$\left(\tilde{x}^k - \mathscr{E}_x(\bar{u}^k) - x^*\right)^\top \left(\nabla f(x^*) - A^\top \lambda^*\right) \geq 0. \tag{3.18}$$

On the other hand, by setting $\Omega := \mathcal{X}$, $u := \tilde{x}^k - \beta(\nabla f(\tilde{x}^k) - A^\top \bar{\lambda}^k)$, and $v := x^*$ in (2.2), we have

$$\left(\mathscr{E}_x(\bar{u}^k) - \beta(\nabla f(\tilde{x}^k) - A^\top \bar{\lambda}^k)\right)^\top \left(x^* - \tilde{x}^k + \mathscr{E}_x(\bar{u}^k)\right) \leq 0. \tag{3.19}$$

Multiplying inequality (3.18) by β and summing (3.18) and (3.19), it follows from $\nabla f(x) = K^\top(Kx - d)$ that

$$\left(\tilde{x}^k - x^* - \mathscr{E}_x(\bar{u}^k)\right)^\top \left(\mathscr{E}_x(\bar{u}^k) - \beta K^\top K(\tilde{x}^k - x^*) + \beta A^\top(\bar{\lambda}^k - \lambda^*)\right) \geq 0.$$

Rearranging the above inequality leads to

$$\begin{aligned}
&(\tilde{z}_1^k - z_1^*)^\top \mathscr{E}_x(\bar{u}^k) + \beta(\tilde{\lambda}^k - \lambda^*)^\top A \left(\tilde{x}^k - x^* - \mathscr{E}_x(\bar{u}^k)\right) \\
\geq &\|\mathscr{E}_x(\bar{u}^k)\|^2 + \beta(\tilde{x}^k - x^*)^\top K^\top K(\tilde{x}^k - x^*) + \beta(\tilde{\lambda}^k - \bar{\lambda}^k)^\top A \left(\tilde{x}^k - x^* - \mathscr{E}_x(\bar{u}^k)\right) \\
\geq &\|\mathscr{E}_x(\bar{u}^k)\|^2 + \beta(\tilde{\lambda}^k - \bar{\lambda}^k)^\top A \left(\tilde{x}^k - x^* - \mathscr{E}_x(\bar{u}^k)\right),
\end{aligned} \tag{3.20}$$

where the second inequality comes from the fact that $K^\top K$ is a positive semi-definite matrix.

Similarly, we can also obtain

$$(z_2^k - z_2^*)^\top \mathscr{E}_y(\bar{v}^k) + \beta(\tilde{\lambda}^k - \lambda^*)^\top B\left(y^k - y^* - \mathscr{E}_y(\bar{v}^k)\right) \\ \geq \|\mathscr{E}_y(\bar{v}^k)\|^2 + \beta(\tilde{\lambda}^k - \bar{\lambda}^k)^\top B\left(y^k - y^* - \mathscr{E}_y(\bar{v}^k)\right). \tag{3.21}$$

By adding (3.20) and (3.21), it follows from the notation of $\mathscr{E}_\lambda(\bar{w}^k)$ in (3.4c) that

$$\begin{pmatrix} \tilde{z}_1^k - z_1^* \\ z_2^k - z_2^* \\ \tilde{\lambda}^k - \lambda^* \end{pmatrix}^\top \begin{pmatrix} \mathscr{E}_x(\bar{u}^k) \\ \mathscr{E}_y(\bar{v}^k) \\ \mathscr{E}_\lambda(\bar{w}^k) \end{pmatrix} \geq \|\mathscr{E}_x(\bar{u}^k)\|^2 + \|\mathscr{E}_y(\bar{v}^k)\|^2 + (\tilde{\lambda}^k - \bar{\lambda}^k)^\top \mathscr{E}_\lambda(\bar{w}^k).$$

Consequently, we conclude the assertion of this lemma by using the notations $\mathbf{z}^*$, $\tilde{\mathbf{z}}^k$, $\mathscr{D}(\bar{w}^k)$ and $\varphi(\bar{w}^k)$ given by (3.11), (3.13) and (3.14), respectively. □

By invoking the first-order optimality condition of subproblems (3.7) and (3.8) with the notations z_1, z_2 in (3.13), we obtain

$$z_1^{k+1} = \tilde{z}_1^k - \gamma\alpha_k \mathscr{E}_x(\bar{u}^k) \quad \text{and} \quad z_2^{k+1} = z_2^k - \gamma\alpha_k \mathscr{E}_y(\bar{v}^k).$$

Note that, from the notations of $\mathbf{z}^{k+1}$, $\tilde{\mathbf{z}}^k$ and $\mathscr{D}(\bar{w}^k)$ defined in (3.13) and (3.14), the iterative scheme (3.7)-(3.9) then can be recast into the compact form:

$$\mathbf{z}^{k+1} = \tilde{\mathbf{z}}^k - \gamma\alpha_k \mathscr{D}(\bar{w}^k). \tag{3.22}$$

Thus, if $\varphi(\bar{w}^k)$ is nonnegative, Lemma 5 implies that $-\mathscr{D}(\bar{w}^k)$ is a descent direction of the distance function $\frac{1}{2}\|\mathbf{z} - \mathbf{z}^*\|^2$ at $\tilde{\mathbf{z}}^k$. Then, in this sense, α_k plays the role of "step size", and γ can be viewed as a relaxed factor. In the following lemma, we will prove the sequence $\{\alpha_k\}$ generated by Algorithm 1 is bounded below.

Lemma 6 *Assume that $\beta \in (0, \sqrt{2}/c)$. For the step size α_k given by (3.10), there exists a constant $\alpha_{\min} > 0$ such that*

$$\alpha_k \geq \alpha_{\min}, \quad \forall k \geq 0.$$

Proof For any two vectors $a, b \in \mathbb{R}^m$, we have the inequality

$$2a^\top b \leq \delta\|a\|^2 + 1/\delta\|b\|^2, \quad \delta > 0. \tag{3.23}$$

Hence, applying (3.23) to the last term of (3.11), for any $\delta > 0$, we have

$$\begin{aligned}(\tilde{\lambda}^k - \bar{\lambda}^k)^\top \mathscr{E}_\lambda(\bar{w}^k) &= \|\tilde{\lambda}^k - \bar{\lambda}^k\|^2 - \beta(\tilde{\lambda}^k - \bar{\lambda}^k)^\top \left(A\mathscr{E}_x(\bar{u}^k) + B\mathscr{E}_y(\bar{v}^k)\right) \\ &\geq (1-\delta)\|\tilde{\lambda}^k - \bar{\lambda}^k\|^2 - \frac{\beta^2}{2\delta}\left(\|A\mathscr{E}_x(\bar{u}^k)\|^2 + \|B\mathscr{E}_y(\bar{v}^k)\|^2\right) \\ &\geq (1-\delta)\|\tilde{\lambda}^k - \bar{\lambda}^k\|^2 - \frac{\beta^2c^2}{2\delta}\left(\|\mathscr{E}_x(\bar{u}^k)\|^2 + \|\mathscr{E}_y(\bar{v}^k)\|^2\right),\end{aligned} \tag{3.24}$$

where $c := \max\{\|A\|, \|B\|\}$. By substituting (3.24) into (3.11) and setting $\delta := \beta c/\sqrt{2}$, we deduce

$$\begin{aligned}\varphi(\bar{w}^k) &\geq (1-\delta)\|\tilde{\lambda}^k - \bar{\lambda}^k\|^2 + \frac{2\delta - \beta^2c^2}{2\delta}\left(\|\mathscr{E}_x(\bar{u}^k)\|^2 + \|\mathscr{E}_y(\bar{v}^k)\|^2\right) \\ &= \left(1 - \frac{\beta c}{\sqrt{2}}\right)\left(\|\tilde{\lambda}^k - \bar{\lambda}^k\|^2 + \|\mathscr{E}_x(\bar{u}^k)\|^2 + \|\mathscr{E}_y(\bar{v}^k)\|^2\right)\end{aligned}$$

$$\geq 0, \tag{3.25}$$

where the last inequality comes from the condition $\beta \in (0, \sqrt{2}/c)$.

On the other hand,

$$\begin{aligned}\psi(\bar{w}^k) &\leq \|\mathscr{E}_x(\bar{u}^k)\|^2 + \|\mathscr{E}_y(\bar{v}^k)\|^2 + 2\|\tilde{\lambda}^k - \bar{\lambda}^k\|^2 + 2\beta^2\left(\|A\mathscr{E}_x(\bar{u}^k)\|^2 + \|B\mathscr{E}_y(\bar{v}^k)\|^2\right)\\ &\leq c'\left(\|\tilde{\lambda}^k - \bar{\lambda}^k\|^2 + \|\mathscr{E}_x(\bar{u}^k)\|^2 + \|\mathscr{E}_y(\bar{v}^k)\|^2\right),\end{aligned} \tag{3.26}$$

where $c' := \max\{2,\ 1 + 4\beta^2\|A\|^2,\ 1 + 4\beta^2\|B\|^2\}$. By the definition of α_k given in (3.10), it immediately follows from (3.25) and (3.26) that

$$\alpha_k = \frac{\varphi(\bar{w}^k)}{\psi(\bar{w}^k)} \geq \frac{\sqrt{2} - \beta c}{\sqrt{2}c'} =: \alpha_{\min} > 0$$

holds under the condition $\beta \in (0, \sqrt{2}/c)$. The proof is completed. □

By the definitions of z_1 and $\tilde{z}_1$ in (3.13) and (3.14), it then follows from $\nabla f(x) = K^\top(Kx - d)$ that

$$\begin{aligned}\tilde{z}_1^k &= \tilde{x}^k + \beta K^\top\left(K\tilde{x}^k - d\right)\\ &= x^k + \beta K^\top\left(Kx^k - d\right) + \tau\left(I + \beta K^\top K\right)\left(x^k - x^{k-1}\right)\\ &= z_1^k + \tau\left(z_1^k - z_1^{k-1}\right).\end{aligned}$$

Consequently, by denoting $\boldsymbol{p} = (z_1, \lambda)$ as a subvector of $\mathbf{z} = (z_1, z_2, \lambda)$ for notational brevity, it then follows from the above equality and iterative scheme (3.5) that

$$\tilde{\boldsymbol{p}}^k = \boldsymbol{p}^k + \tau(\boldsymbol{p}^k - \boldsymbol{p}^{k-1}). \tag{3.27}$$

Theorem 1 *Suppose that $\beta \in (0, \sqrt{2}/c)$. Then the sequence $\{w^k\}$ generated by Algorithm 1 converges to a solution of VI (2.4).*

Proof It follows from (3.22) that

$$\begin{aligned}\|\mathbf{z}^{k+1} - \mathbf{z}^*\|^2 &= \|\tilde{\mathbf{z}}^k - \gamma\alpha_k\mathscr{D}(\bar{w}^k) - \mathbf{z}^*\|^2\\ &= \|\tilde{\mathbf{z}}^k - \mathbf{z}^*\|^2 - 2\gamma\alpha_k(\tilde{\mathbf{z}}^k - \mathbf{z}^*)^\top\mathscr{D}(\bar{w}^k) + \gamma^2\alpha_k^2\|\mathscr{D}(\bar{w}^k)\|^2\\ &\leq \|\tilde{\mathbf{z}}^k - \mathbf{z}^*\|^2 - 2\gamma\alpha_k\varphi(\bar{w}^k) + \gamma^2\alpha_k^2\|\mathscr{D}(\bar{w}^k)\|^2\\ &= \|\tilde{\mathbf{z}}^k - \mathbf{z}^*\|^2 - \gamma(2 - \gamma)\alpha_k^2\|\mathscr{D}(\bar{w}^k)\|^2\\ &= \|\tilde{\mathbf{z}}^k - \mathbf{z}^*\|^2 - \frac{2 - \gamma}{\gamma}\|\mathbf{z}^{k+1} - \tilde{\mathbf{z}}^k\|^2,\end{aligned} \tag{3.28}$$

where the first inequality is due to Lemma 5 and the third equality follows from (3.10), (3.12) and (3.13). By invoking (2.3), it follows from (3.27) that

$$\begin{aligned}\|\tilde{\boldsymbol{p}}^k - \boldsymbol{p}^*\|^2 &= \|\boldsymbol{p}^k - \boldsymbol{p}^* + \tau(\boldsymbol{p}^k - \boldsymbol{p}^{k-1})\|^2\\ &= (1 + \tau)\|\boldsymbol{p}^k - \boldsymbol{p}^*\|^2 - \tau\|\boldsymbol{p}^{k-1} - \boldsymbol{p}^*\|^2 + \tau(1 - \tau)\|\boldsymbol{p}^k - \boldsymbol{p}^{k-1}\|^2.\end{aligned} \tag{3.29}$$

For any $\rho > 0$, we have

$$\|\boldsymbol{p}^{k+1} - \tilde{\boldsymbol{p}}^k\|^2 = \|\boldsymbol{p}^{k+1} - \boldsymbol{p}^k - \tau(\boldsymbol{p}^k - \boldsymbol{p}^{k-1})\|^2$$

$$\geq \|\boldsymbol{p}^{k+1} - \boldsymbol{p}^k\|^2 + \tau^2\|\boldsymbol{p}^k - \boldsymbol{p}^{k-1}\|^2 + \tau\left(-\rho\|\boldsymbol{p}^{k+1} - \boldsymbol{p}^k\|^2 - \frac{1}{\rho}\|\boldsymbol{p}^k - \boldsymbol{p}^{k-1}\|^2\right), \tag{3.30}$$

where the last inequality follows from the fact that $2a^\top b \geq -\frac{1}{t}\|a\|^2 - t\|b\|^2$ for any $a, b \in \mathbb{R}^n$ and $t > 0$. Recalling the definition of $\tilde{\mathbf{z}}$ in (3.14) and substituting (3.29) and (3.30) into (3.28), we arrive at

$$\begin{aligned}&\|\boldsymbol{p}^{k+1} - \boldsymbol{p}^*\|^2 - (1+\tau)\|\boldsymbol{p}^k - \boldsymbol{p}^*\|^2 + \tau\|\boldsymbol{p}^{k-1} - \boldsymbol{p}^*\|^2 + \|z_2^{k+1} - z_2^*\|^2 - \|z_2^k - z_2^*\|^2\\ &\leq \frac{(2-\gamma)(\rho\tau - 1)}{\gamma}\|\boldsymbol{p}^{k+1} - \boldsymbol{p}^k\|^2 + s\|\boldsymbol{p}^k - \boldsymbol{p}^{k-1}\|^2 - \frac{2-\gamma}{\gamma}\|z_2^{k+1} - z_2^k\|^2, \end{aligned} \tag{3.31}$$

where

$$s := \tau(1+\tau) + \frac{\tau(2-\gamma)(1-\rho\tau)}{\gamma\rho}.$$

Setting $\rho = \frac{(2-2\gamma)\tau^2 - \gamma\tau - \gamma + 2}{2(2-\gamma)\tau}$, there exists $\bar{\tau} \in (0, 1)$ such that $\rho > 0$ for any $\tau \in (0, \bar{\tau})$ and $\rho\tau < 1$ by simple validation. Combining with $\gamma \in \left(\frac{2\tau^2+2}{2\tau^2+\tau+1}, \frac{2\tau+2}{2\tau+1}\right) \subset (1, 2)$, it is clear that $s \geq 0$ holds.

For convenience, we define the sequence $\{\Phi_k\}$ by

$$\Phi_k := \|\boldsymbol{p}^k - \boldsymbol{p}^*\|^2 - \tau\|\boldsymbol{p}^{k-1} - \boldsymbol{p}^*\|^2 + \|z_2^k - z_2^*\|^2 + s\|\boldsymbol{p}^k - \boldsymbol{p}^{k-1}\|^2$$

for all $k \geq 1$. Then, we have

$$\begin{aligned}\Phi_{k+1} - \Phi_k = &\|\boldsymbol{p}^{k+1} - \boldsymbol{p}^*\|^2 - (1+\tau)\|\boldsymbol{p}^k - \boldsymbol{p}^*\|^2 + \tau\|\boldsymbol{p}^{k-1} - \boldsymbol{p}^*\|^2\\ &+\|z_2^{k+1} - z_2^*\|^2 - \|z_2^k - z_2^*\|^2 + s\|\boldsymbol{p}^{k+1} - \boldsymbol{p}^k\|^2 - s\|\boldsymbol{p}^k - \boldsymbol{p}^{k-1}\|^2.\end{aligned}$$

Using (3.31) yields

$$\Phi_{k+1} - \Phi_k \leq \left(\frac{(2-\gamma)(\rho\tau - 1)}{\gamma} + s\right)\|\boldsymbol{p}^{k+1} - \boldsymbol{p}^k\|^2 - \frac{2-\gamma}{\gamma}\|z_2^{k+1} - z_2^k\|^2.$$

By the definitions of ρ and s, we have

$$\frac{(2-\gamma)(\rho\tau - 1)}{\gamma} + s = \frac{1+\tau}{2\gamma}\frac{[(2\tau+1)\gamma - 2(\tau+1)][(2\tau^2 - \tau + 1)\gamma - 2(1-\tau)^2]}{(2\tau^2+\tau+1)\gamma - 2(\tau^2+1)} < 0,$$

where the inequality is due to the selection of γ in Algorithm 1. Thus there exists a constant $\delta > 0$ such that $\delta < \min\left\{\frac{(2-\gamma)(1-\rho\tau)}{\gamma} - s, \frac{2-\gamma}{\gamma}\right\}$, then

$$\Phi_{k+1} - \Phi_k \leq -\delta\|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2. \tag{3.32}$$

Hence, the sequence $\{\Phi_k\}$ is monotonically nonincreasing, and in particular for all $k \geq 0$,

$$\|\boldsymbol{p}^k - \boldsymbol{p}^*\|^2 - \tau\|\boldsymbol{p}^{k-1} - \boldsymbol{p}^*\|^2 \leq \Phi_k \leq \Phi_1.$$

Consequently, it follows that

$$\|\boldsymbol{p}^k - \boldsymbol{p}^*\|^2 \leq \tau^k\|\boldsymbol{p}^0 - \boldsymbol{p}^*\|^2 + \Phi_1\sum_{i=0}^{k-1}\tau^i \leq \tau^k\|\boldsymbol{p}^0 - \boldsymbol{p}^*\|^2 + \frac{\Phi_1}{1-\tau}, \tag{3.33}$$

which implies that $\{\boldsymbol{p}^k\}$ is bounded due to the fact $\tau \in (0,1)$. Hence, the sequence $\{z^k\}$ is also bounded, where $\{z_2^k\}$ is bounded from the monotonicity of $\{\Phi_k\}$. Combining (3.32) and (3.33), we have

$$\delta \sum_{i=1}^{k} \|\mathbf{z}^{i+1} - \mathbf{z}^i\|^2 \leq \Phi_1 - \Phi_{k+1} \leq \Phi_1 + \tau \|\boldsymbol{p}^k - \boldsymbol{p}^*\|^2 \leq \tau^{k+1} \|\boldsymbol{p}^0 - \boldsymbol{p}^*\|^2 + \frac{\Phi_1}{1-\tau},$$

which shows that

$$\sum_{k=0}^{+\infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2 < +\infty. \tag{3.34}$$

Thus we have $\lim_{k\to+\infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$. By the definition of $\tilde{\mathbf{z}}^k$, (3.27) and triangle inequality, we have

$$\|\mathbf{z}^{k+1} - \tilde{\mathbf{z}}^k\| \leq \|\mathbf{z}^{k+1} - \mathbf{z}^k\| + \tau \|\boldsymbol{p}^k - \boldsymbol{p}^{k-1}\|,$$

which means that $\lim_{k\to+\infty} \|\mathbf{z}^{k+1} - \tilde{\mathbf{z}}^k\| = 0$. Recalling (3.22), we further obtain

$$\lim_{k\to+\infty} \mathscr{D}(\bar{w}^k) = 0.$$

Then, using Lemma 4 leads to

$$\lim_{k\to+\infty} \mathscr{E}(w^k, \beta) = 0. \tag{3.35}$$

On the other hand, the sequence $\{w^k\}$ is bounded due to the boundedness of $\{\mathbf{z}^k\}$. Then the sequence $\{w^k\}$ has at least one cluster point $w^\infty = (x^\infty, y^\infty, \lambda^\infty)$ and $\{w^{k_j} = (x^{k_j}, y^{k_j}, \lambda^{k_j})\}$ be the corresponding subsequence converging to w^∞. Thus, taking limit along this subsequence in (3.35) together with the continuity of $\mathscr{E}(w,\beta)$ with respect to w, we immediately obtain

$$\|\mathscr{E}(w^\infty, \beta)\|^2 = \|\mathscr{E}(\lim_{j\to\infty} w^{k_j}, \beta)\|^2 = \lim_{j\to\infty} \|\mathscr{E}(w^{k_j}, \beta)\|^2 = 0.$$

According to Proposition 1, the above fact means that w^∞ is a solution of VI (2.4). Finally, applying Lemma 3 with setting $\mathcal{D} = \mathcal{W}^*$, we conclude that the sequence $\{w^k\}$ generated by Algorithm 1 converges to w^∞, a solution of VI (2.4). The proof is completed. □

4 Numerical Experiments

In this section, we shall conduct the numerical performance of the proposed ICDRSM (i.e., Algorithm 1) to verify that the inertial technique can really improve the efficiency of solving (1.1), when comparing with some state-of-the-art benchmark solvers, including the classical ADMM (see (1.8)), inertial ADMM (denoted by "IADMM") in [15], the primal-dual method in [13] (denoted by "PDM") and inertial PDM (denoted by "IPDM") in [14], and the original CDRSM proposed in [29]. All the codes are written by MATLAB R2021a and all the numerical experiments are conducted on a 64-bit PC with an Intel(R) Core(TM) i5-8265U CPU @ 1.6GHz with 8GB RAM.

4.1 Constrained Lasso

In this subsection, we conduct the numerical performance of the proposed ICDRSM for solving the following ball-constrained Lasso problem

$$\min_{\|x\|\leq 1} \left\{ F(x) := \frac{1}{2}\|Kx-d\|^2 + \mu\|x\|_1 \right\}, \tag{4.1}$$

which is a special case of (1.4). Here, we consider the reformulation given by (1.5). Therefore, applying our ICDRSM (i.e., Algorithm 1) to (1.5), the x- and y-subproblems (i.e., (3.7) and (3.8)) can be specified as

$$\begin{cases} x^{k+1} = \left(I+\beta K^\top K\right)^{-1}\left(\omega_x^k + \beta K^\top d\right), \\ y^{k+1} = \text{Shrink}_{\mu\beta}\left(\omega_y^k\right), \end{cases}$$

where 'Shrink$_\beta$ $(\cdot)$' is the soft-shrinkage operator given by

$$\text{Shrink}_\beta(\cdot) := \max\{\|\cdot\| - \beta, 0\}\frac{\cdot}{\|\cdot\|}. \tag{4.2}$$

Here, we should notice that the subgradient ζ^{k+1} of $\mu\|y\|_1$ at y^{k+1} can be updated by

$$\zeta^{k+1} := \frac{1}{\beta}\left(\omega_y^k - y^{k+1}\right).$$

Note that CDRSM shares similar subproblems with ICDRSM, we omit its details here for brevity. When applying ADMM to (1.5), the iterative scheme (1.8) reads as

$$\begin{cases} x^{k+1} = \arg\min\left\{\frac{1}{2}\|Kx-d\|^2 + \frac{\beta}{2}\|x-y^k-\frac{1}{\beta}\lambda^k\|^2 \mid \|x\|\leq 1\right\}, \\ y^{k+1} = \arg\min\left\{\mu\|y\|_1 + \frac{\beta}{2}\|x^{k+1}-y-\frac{1}{\beta}\lambda^k\|^2\right\} = \text{Shrink}_{\mu/\beta}\left(x^{k+1}-\frac{1}{\beta}\lambda^k\right), \\ \lambda^{k+1} = \lambda^k - \gamma\beta(x^{k+1}-y^{k+1}). \end{cases} \tag{4.3}$$

To implement the PDM [13] to solve (4.1), we first reformulate (4.1) as a saddle point problem, i.e.,

$$\min_{\|x\|\leq 1}\max_{y\in\mathcal{Y}_\infty}\left\{y^\top x + \frac{1}{2\mu}\|Kx-d\|^2\right\}, \tag{4.4}$$

where $\mathcal{Y}_\infty := \{y \mid \|y\|_\infty \leq 1\}$. Then, the iterative scheme of PDM solving (4.4) is specified as

$$\begin{cases} y^{k+1} = \arg\max\left\{\sigma y^\top x^k - \frac{1}{2}\|y-y^k\|^2 \mid y\in\mathcal{Y}_\infty\right\} = P_{\mathcal{Y}_\infty}[y^k+\sigma x^k], \\ \bar{y}^{k+1} = y^{k+1} + \theta(y^{k+1}-y^k), \\ x^{k+1} = \arg\min\left\{\frac{\nu}{2\mu}\|Kx-d\|^2 + \frac{1}{2}\|x-x^k+\nu\bar{y}^{k+1}\|^2 \mid \|x\|\leq 1\right\}. \end{cases} \tag{4.5}$$

Apparently, the main difficulties of ADMM and PDM for solving (4.1) come from their x-subproblems, i.e., (4.1) and (4.1), where the nonnegative constraint makes their x-subproblems lose closed-form solutions. As suggested by [29], we employ the projected Barzilai-Borwein method (PBBM) in [4, 19] to find approximate solutions of (4.1) and (4.1).

Before the formal simulation, we investigated the numerical sensitivity of the maximum inner iterations of PBBM to the performance of those algorithms that need PBBM, where we considered five cases by setting the largest number of inner iterations as $\{10, 20, 30, 40, 50\}$. The results showed that ADMM and PDM equipped with 20 performed slightly better than the other settings in many cases. Therefore, we empirically choose 20 as the maximum number of inner iterations for the following experiments.

Below, we conduct the numerical performance of the above six algorithms on synthetic data. In our experiments, we test four scenarios for $(m, n) = \{(1000, 3000), (3000, 5000), (5000, 8000), (8000, 10000)\}$. Here, we first generate $K \in \mathbb{R}^{m\times n}$ in a random way such that its components satisfy independent and identical standard normal distribution. Then, we construct the observed vector d via $d = K\hat{x} + \epsilon$, where $\epsilon \in N(0, 0.01 \cdot I_m)$ and $\hat{x}$ in ℓ_2 unit ball is a sparse random vector with $0.2n$ nonzeros. For the trade-off parameter μ, we take it as 0.001. Now, we specify the choices of parameters to implement these algorithms. We empirically set $\tau = 0.3$ for ICDRSM for the four scenarios. For IADMM and IPDM, we set inertial parameters $\tau = 0.1$ for the first scenario, $\tau = 0.4$ for the second and third scenarios and $\tau = 0.6$ for the last scenario. We set $\gamma = \frac{1}{2}\left(\frac{2(1+\tau^2)}{2\tau^2+\tau+1} + \frac{2+2\tau}{1+2\tau}\right)$ and $\beta = 0.0005$ for ICDRSM, $\gamma = 1.6$ and $\beta = 0.0005$ for CDRSM, $\beta = 0.1$ and $\gamma = 1.6$ for ADMM and IADMM, $\nu = 0.5$, $\sigma = \frac{1}{\nu}$ and $\theta = 1$ for PDM and IPDM. The initial iterates for all tested methods are zero vectors $x^0 = y^0 = \lambda^0 = \mathbf{0}_{n\times 1}$. Finally, we adopt the following residual as the stopping criterion for all methods

$$\text{resi} = |F(x^k) - F(x^*)| \leq 5 \cdot 10^{-3}, \tag{4.6}$$

where $F(x^*)$ is the objective value of termination point obtained by running CDRSM to satisfy

$$\|F(x^k) - F(x^{k-1})\| \leq 10^{-12}.$$

In Table 1, we report the computing time in seconds (‘CPU’) and the number of iterations (‘It.’). Note that ADMM and PDM require an inner loop to find approximate solutions of their subproblems, we accordingly report the total inner iterations (‘InIt.’) for solving x-subproblems. For each scenario, we test 10 times and report the averaged performance. The symbol ‘–’ in Table 1 means that the computing time exceeds 20 s. It can be seen from Table 1 that all inertial methods take fewer iterations and less computing time than those without inertial acceleration. Both CDRSM and ICDRSM take more iterations than ADMM, PDM and their inertial variants. However, owing to the inner iterations for solving x-subproblems, the total computing time of CDRSM and ICDRSM is less than ADMM-like and PDM-like methods. Comparing ICDRSM with CDRSM, we can see from Table 1 that ICDRSM takes fewer iterations and less computing time than CDRSM, which support the idea of this paper that inertial technique can further improve the performance of CDRSM for separable problem (1.1).

To show more details of their numerical behaviors, we show box plots with respect to computing time of these 10 times in Fig. 1, which reports the minimum, median, and maximum computing time of four methods for solving 10 groups of random problems. We can see from Fig. 1 that CDRSM and ICDRSM are relatively stable and more efficient than ADMM and PDM. Moreover, the convergence curves shown in Figs. 2 and 3 demonstrate that ICDRSM is the fastest solver for solving ball-constrained Lasso (4.1).

As we all know, inertial parameter is important for inertial-type algorithms. Therefore, we are further concerned with the numerical sensitivity of the inertial parameter to the three inertial-type algorithms. In our experiments, we consider four scenarios and conduct behaviors on CPU time of the three inertial-type algorithms equipped with $\tau =$

Table 1 Numerical results for box constrained Lasso

n	ADMM		IADMM		PDM		IPDM		CDRSM		ICDRSM	
	CPU	It./InIt.	CPU	It./InIt.	CPU	It./InIt.	CPU	It./InIt.	CPU	It.	CPU	It.
3000	0.32	10/178	0.24	9/160	0.31	10/175	0.23	9/160	0.18	28.7	**0.11**	17.1
5000	3.49	19/380	2.15	12.1/242	3.49	19/380	2.2	12.1/242	1.06	58.2	**0.68**	31.2
8000	10.78	22/440	5.78	13/260	10.8	22.1/440	5.88	13/260	2.09	44.9	**1.29**	25.4
10,000	–	–	16.6	21.7/434	–	–	16.9	21.7/434	7.82	96.2	**5.73**	64.9

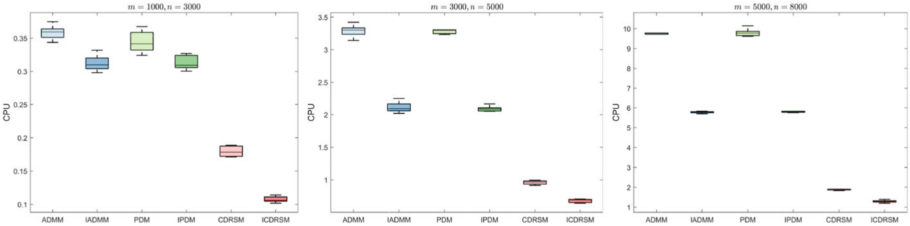

Fig. 1 Box plots with respect to CPU of the six methods for solving constrained Lasso

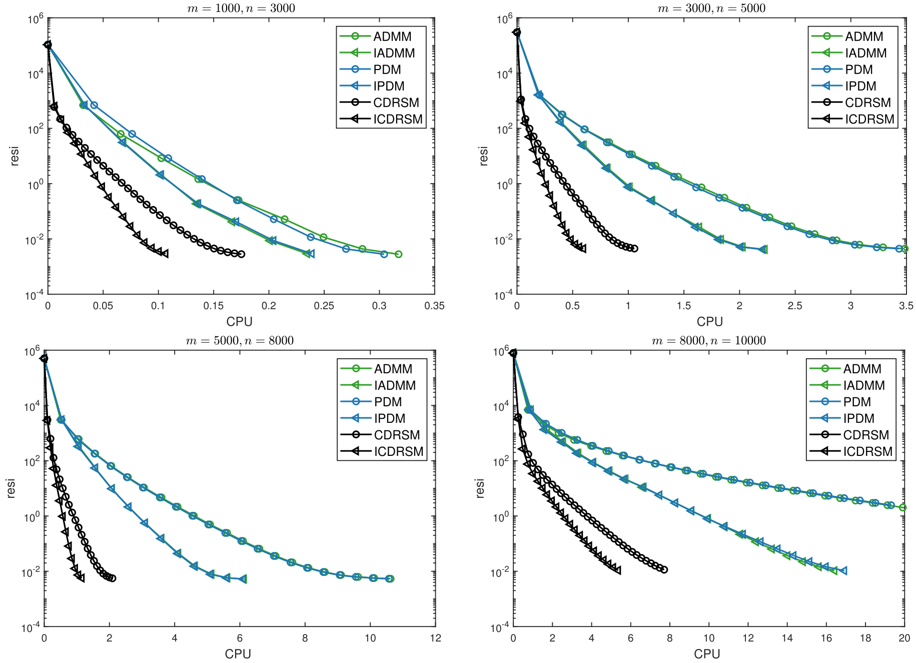

Fig. 2 Evolution of relative errors with respect to computing time for constrained Lasso

$\{0.0, 0.1, 0.2, \ldots, 0.9, 0.999\}$. The bar plots are listed in Fig. 4, where the algorithmic parameters and stopping criterion are the same as the previous experiments. We can observe from Fig. 4 that, with the increase of $\tau \in [0, 1)$, the CPU time required by the three algorithms decreases at first and then increases. Comparatively, ICDRSM equipped with $\tau \in [0.3, 0.5]$ is more reliable, and the inertial parameter should be larger as the dimension of the problem increases for IADMM and IPDM. This experience also supports our settings of the inertial parameters in the previous experiments.

4.2 Constrained TV-Based Image Restoration

In this subsection, we conduct the performance of the proposed ICDRSM on solving constrained TV-based image restoration problem (1.2). Here, we also consider the separable form (1.3) of (1.2). So, applying ICDRSM to model (1.3), both subproblems (3.7) and (3.8)

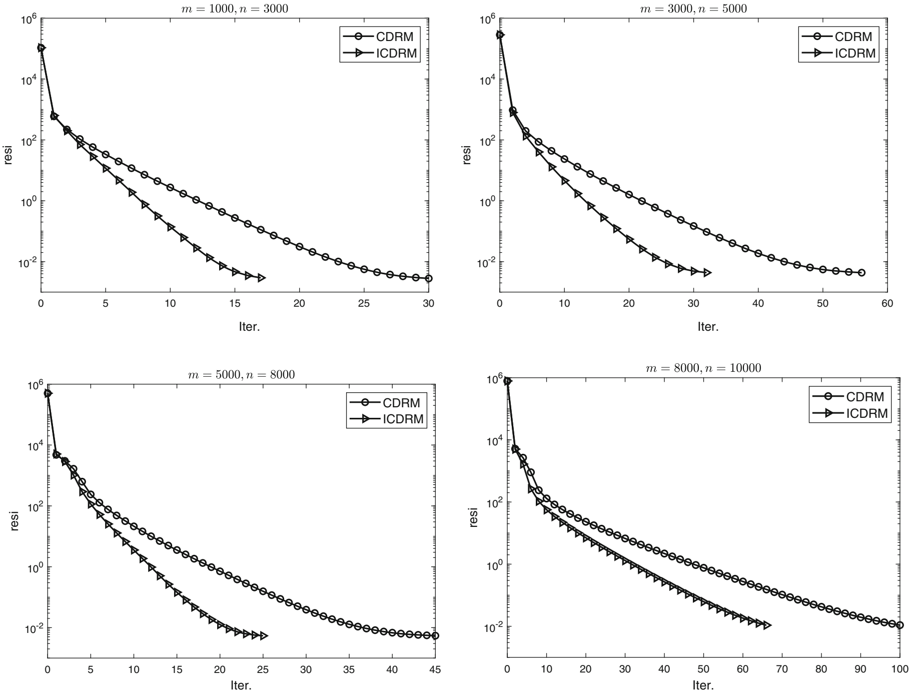

Fig. 3 Evolution of relative errors with respect to iterations for constrained Lasso

of ICDRSM are specified as

$$\begin{cases} \left(I + \beta K^{\top} K\right) x^{k+1} = \left(I + \beta K^{\top} K\right) \tilde{x}^{k} - \gamma \alpha_k \mathscr{E}_x(\bar{u}^k), \\ y^{k+1} = \text{Shrink}_{\mu\beta}\left(\omega_y^k\right), \end{cases} \tag{4.7}$$

where $\mathscr{E}_x(\bar{u}^k)$ and ω_y^k are defined in Algorithm 1. Theoretically, the x-subproblem (4.2) has an explicit solution. However, as shown in [31, p.43], the linear system (4.2) is often solved by the fast Fourier transform (FFT) or discrete cosine transform. In our experiments, we employ FFT to find the solution of (4.2). When applying ADMM to (1.3), the main subproblems are specified as

$$\begin{cases} x^{k+1} = \arg\min_{x \in \mathcal{B}} \left\{ \frac{1}{2}\|Kx - d\|^2 + \frac{\beta}{2}\|\boldsymbol{D}x - y^k - \frac{1}{\beta}\lambda^k\|^2 \right\}, \\ y^{k+1} = \text{Shrink}_{\mu/\beta}\left(\boldsymbol{D}x^{k+1} - \frac{1}{\beta}\lambda^k\right). \end{cases} \tag{4.8}$$

Like the saddle point reformulation of (1.4), we can also reformulate (1.2) as the following min-max problem

$$\min_{x \in \mathcal{B}} \max_{y \in \mathcal{Y}_\infty} \left\{ y^{\top} \boldsymbol{D}x + \frac{1}{2\mu}\|Kx - d\|^2 \right\}.$$

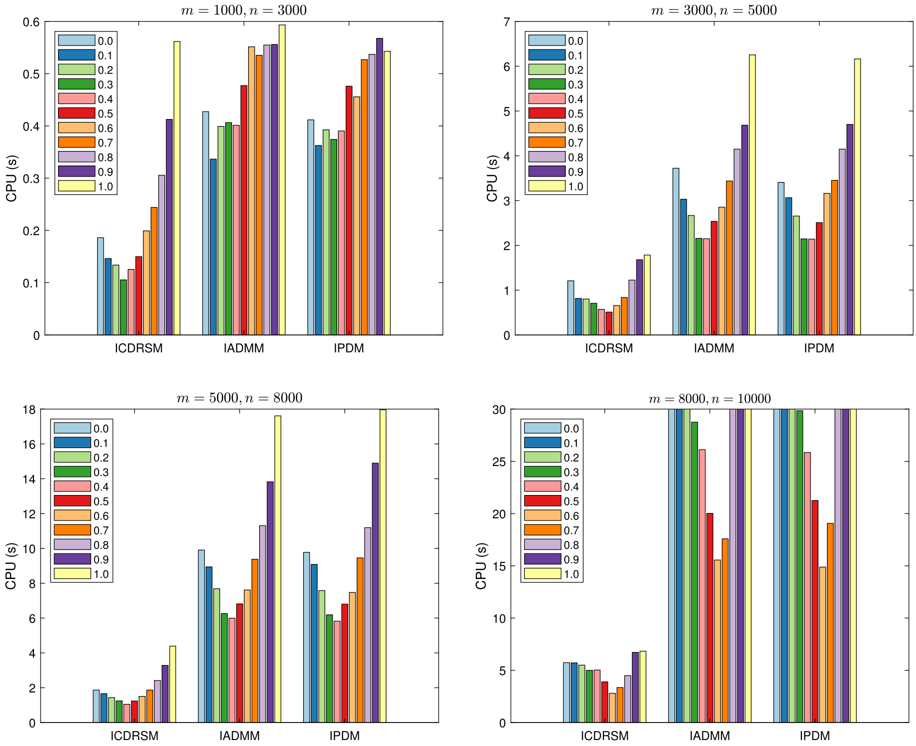

Fig. 4 Numerical sensitivity of inertial parameter $\tau = \{0.0, 0.1, 0.2, \ldots, 0.9, 0.999\}$ in terms of CPU time for the three inertial algorithms

Consequently, the iterative scheme of PDM reads as

$$\begin{cases} y^{k+1} = P_{\mathcal{Y}_\infty}\left[y^k + \sigma \boldsymbol{D} x^k\right], \\ \bar{y}^{k+1} = y^{k+1} + \theta(y^{k+1} - y^k), \\ x^{k+1} = \arg\min\limits_{x \in \mathcal{B}} \left\{ \dfrac{\nu}{2\mu}\|Kx - d\|^2 + \dfrac{1}{2}\left\|x - x^k + \nu \boldsymbol{D}^\top \bar{y}^{k+1}\right\|^2 \right\}. \end{cases} \tag{4.9}$$

Clearly, both x-subproblems (4.2) and (4.2) have no closed-form solution. So we also employ the aforementioned to solve it approximately. Like the previous experiments, we also investigate the numerical sensitivity of the maximum number of inner iterations to the algorithms that need to recall the PBBM. According to our numerical simulation, we empirically choose 10 as the maximum number of inner iterations for the following experiments.

In our experiments, we consider four well-tested images in the literature, i.e., ‘Cameraman.tif (256 × 256)’, ‘Columbia.tiff (480 × 480)’, ‘Crowd.tiff (512 × 512)’ and ‘Man.pgm (1024 × 1024)’. All images are first corrupted by the blurring operator with a 25 × 25 uniform kernel and by adding the zero-white Gaussian noise with the standard derivation 0.001. In Fig. 5, we list the original and degraded images.

For the algorithmic parameters, we take $\tau = 0.7, 0.4, 0.5$ for ICDRSM, IADMM and IPDM, respectively. And we set $\gamma = \frac{1}{2}\left(\frac{2(1+\tau^2)}{2\tau^2+\tau+1} + \frac{2+2\tau}{1+2\tau}\right)$ and $\beta = 0.35$ for ICDRSM, $\gamma = 1.6$ and $\beta = 0.3$ for CDRSM, $\beta = 1$ and $\gamma = 1.6$ for ADMM and IADMM, $\nu = 0.03$,

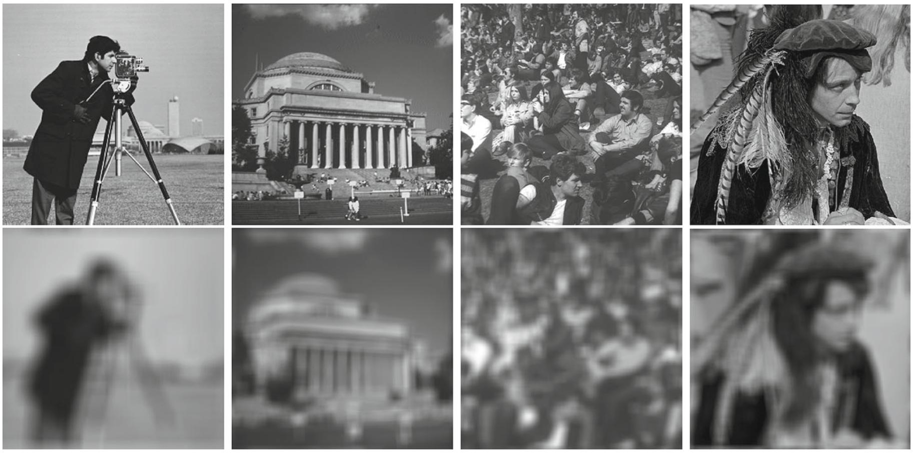

Fig. 5 From left to right: Cameraman.tif, Columbia.tiff, Crowd.tiff, Man.pgm. From top to bottom: original images and degraded images, respectively

$\sigma = \frac{1}{8\nu}$ and $\theta = 1$ for PDM and IPDM. The starting points for all methods are set as zeros $x^0 = y^0 = \lambda^0 = \mathbf{0}$. As used in the literature, we employ the signal-to-noise ratio (SNR) in the unit of dB to measure the quality of restored images. The SNR is defined by

$$\mathrm{SNR} = 10 \log_{10} \frac{\|x^*\|^2}{\|x - x^*\|^2}, \tag{4.10}$$

where x^* is the original image and x represents the restored image.

In Fig. 6, we graphically show the evolution of SNR values with respect to CPU time. It is clear from Fig. 6 that ICDRSM outperforms the other five methods in terms of taking less time to recover images with higher quality.

To see more detailed performance of those methods, we also report the number of iterations, computing time, and SNR values in Table 2, where images are corrupted by blurring operator with a 21 × 21 uniform kernel, and we take pre-set SNR values as stopping criterion. The symbol '–' in Table 2 means that the number of total iterations exceeds 1000 or the computing time exceeds 20 s. As shown in Table 1, both ICDRSM and CDRSM require more iterations than ADMM-like and PDM-like methods. However, ICDRSM and CDRSM take less computing time to achieve higher SNR values than other methods. The computational results in Table 2 further verify that ICDRSM equipped with an inertial step runs faster than CDRSM for solving constrained TV-based image restoration problems.

Finally, we also investigate the numerical sensitivity of the inertial parameter to these inertial-type algorithms for solving constrained TV-based image restoration problems. Here, we follow the settings used in Fig. 4, i.e., $\{\tau = 0.0, 0.1, 0.2, \ldots, 0.9, 0.999\}$. The numerical results are summarized in Fig. 7, where the algorithmic parameters are the same as the previous experiments and the stopping rules are taken as some preset SNR values, i.e., 19.5 dB, 22 dB, 19.5 dB, and 22 dB for Cameraman, Columbia, Crowd, and Man, respectively. It can be observed from Fig. 7 that $\tau = 0.7$ is suitable for ICDRSM, $\tau = 0.4$ is suitable for IADMM and $\tau = 0.5$ is suitable for IPDM, which also supports our settings of the inertial parameters in the previous experiments.

Table 2 Numerical results for constrained TV-based image restoration

Pre-set SNR		Camereman (12.7dB)			Columbia (11.3dB)			Crowd (11.5dB)			Man (14.0dB)		
		19.0	19.5	20.0	21.0	22.0	23.0	19.0	19.5	20.0	20.0	21.0	22.0
ADMM	Iter	40	85	–	30	55	–	45	60	–	15	20	36
	InIter	386	809	–	292	526	–	432	578	–	150	200	360
	CPU	1.11	2.04	–	3.07	5.42	–	5.15	6.74	–	8.45	11.26	20.0
	SNR	19.03	19.53	19.67	21.22	22.02	22.32	19.19	19.56	19.97	20.63	21.15	22.02
IADMM	Iter	40	85	–	30	60	–	35	50	–	15	20	35
	InIter	383	801	–	276	561	–	350	499	–	150	200	350
	CPU	1.05	2.03	–	2.86	5.89	–	4.04	5.83	–	8.54	11.28	19.74
	SNR	19.00	19.52	19.67	21.01	22.01	22.31	19.03	19.53	19.97	20.38	21.13	22.10
PDM	Iter	55	130	–	25	55	–	45	70	–	10	15	–
	InIter	490	1166	–	225	499	–	411	642	–	93	137	–
	CPU	1.40	3.23	–	2.52	6.17	–	5.16	8.12	–	5.77	8.38	–
	SNR	19.05	19.50	19.67	21.14	22.09	22.40	19.09	19.54	19.89	20.61	21.12	21.98
IPDM	Iter	45	105	–	25	45	–	40	60	–	10	15	30
	InIter	404	949	–	234	424	–	389	583	–	95	145	291
	CPU	1.22	2.68	–	2.79	5.25	–	4.87	7.27	–	5.75	8.83	17.78
	SNR	19.07	19.52	19.67	21.31	22.02	22.37	19.11	19.54	19.94	20.51	21.21	22.04
CDRSM	Iter	120	160	220	100	160	220	120	140	160	40	60	80
	CPU	0.75	0.85	1.10	2.77	4.43	6.02	3.80	4.48	5.15	5.55	8.29	11.14
	SNR	19.20	19.62	20.01	21.08	22.37	23.06	19.32	19.70	20.04	20.48	21.41	22.10
ICDRSM	Iter	81	121	181	81	101	161	101	101	121	41	41	61
	CPU	0.44	0.61	0.93	2.19	2.74	4.42	3.13	3.13	3.78	5.69	5.77	8.54
	SNR	19.10	19.61	20.02	21.63	22.25	23.19	19.62	19.62	20.06	21.66	21.66	22.52

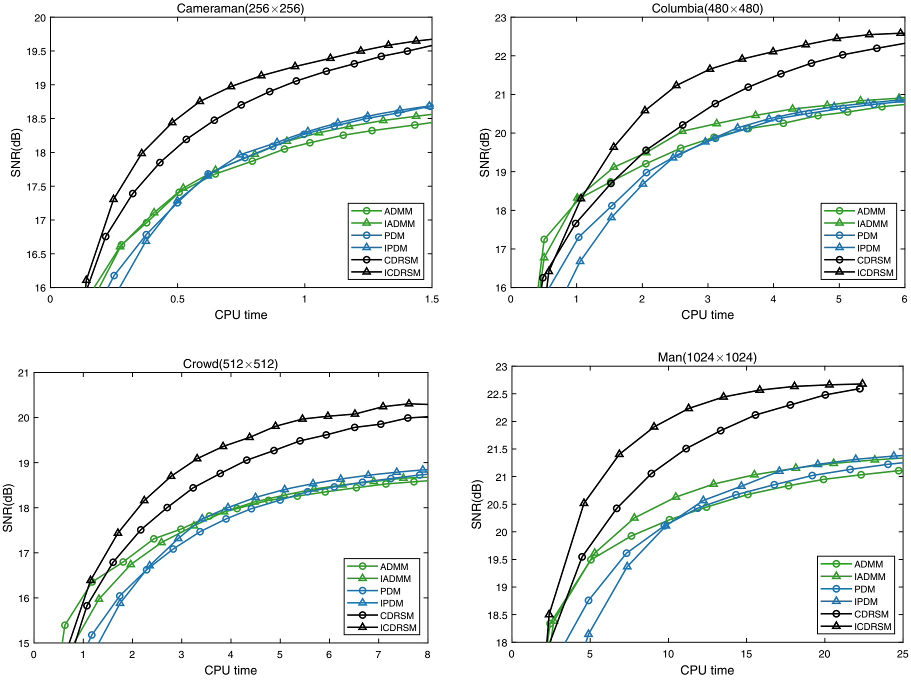

Fig. 6 Evolution of SNR with respect to computing time for solving constrained TV-based image restoration problem

5 Conclusion

In this paper, we considered a class of structured convex optimization problem, which has a separable objective function and a structured constraint. Such problems have many applications in statistical learning, machine learning, and image processing. Due to the appearance of two simple convex sets in the structured constraint, a direct application of the popular ADMM yields two constrained subproblems, thereby failing to exploit the simple structure of the two simple convex sets. Comparatively, although the so-called customized DRSM (CDRSM) proposed in [29] enjoys easier subproblems than ADMM, it does not fully make use of the quadratic term in the objective function, since such a quadratic term dominates the main minimization task. Hence, we introduced the inertial acceleration technique to the quadratic term and proposed a partially inertial CDRSM for the problem under consideration. Under mild conditions, we established its global convergence. Some preliminary numerical results also supported the idea of this paper. In the future, we will pay our attention to the theoretical convergence rate and its extension to the cases where the objective function is nonconvex or nonquadratic.

Acknowledgements The authors would like to thank the two anonymous referees for their valuable comments, which helped us improve the quality of this paper. H. He was supported in part by the National Natural Science Foundation of China (NSFC) No. 12371303 and Ningbo Natural Science Foundation (No. 2023J014). D. Han was supported by NSFC Nos. 12126603 and 12131004 and Ministry of Science and Technology of China (Grant No. 2021YFA1003600).

Data Availability Enquiries about data availability should be directed to the authors.

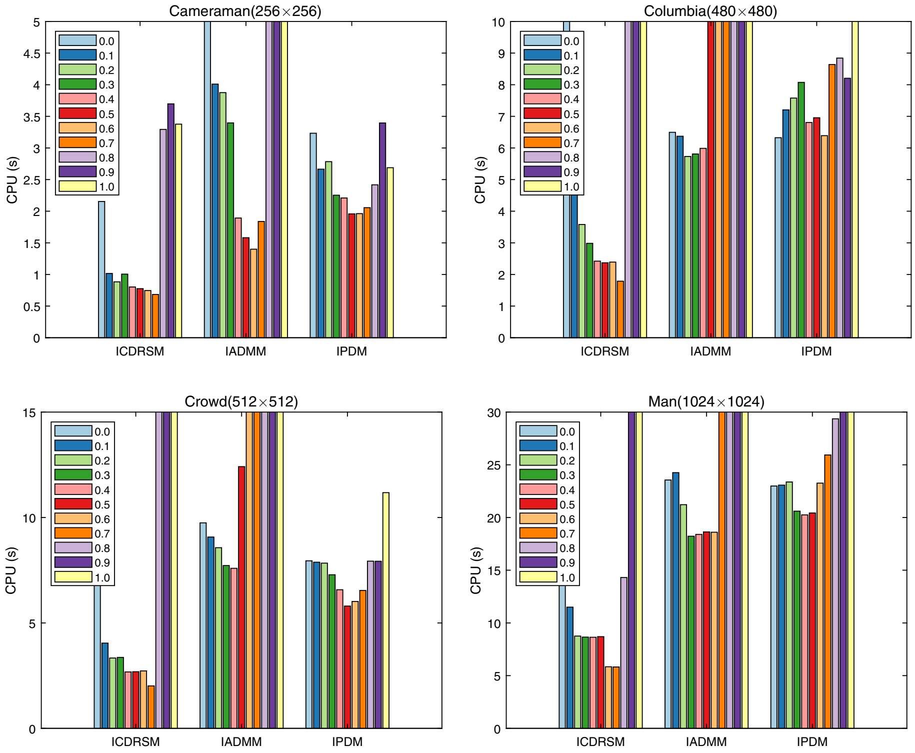

Fig. 7 Numerical sensitivity of inertial parameter $\tau = \{0.0, 0.1, 0.2, \ldots, 0.9, 0.999\}$ in terms of CPU time for the three inertial algorithms

Declarations

Conflict of interest The authors have not disclosed any competing interests.

References

1. Alvarez, F.: Weak convergence of a relaxed and inertial hybrid projection-proximal point algorithm for maximal monotone operators in Hilbert space. SIAM J. Optim. **14**(3), 773–782 (2004)
2. Alves, M.M., Eckstein, J., Geremia, M., Melo, J.G.: Relative-error inertial-relaxed inexact versions of Douglas–Rachford and ADMM splitting algorithms. Comput. Optim. Appl. **75**(2), 389–422 (2020)
3. Attouch, H., Peypouquet, J., Redont, P.: A dynamical approach to an inertial forward-backward algorithm for convex minimization. SIAM J. Optim. **24**(1), 232–256 (2014)
4. Barzilai, J., Borwein, J.: Two-point step size gradient methods. IMA J. Numer. Anal. **8**, 141–148 (1988)
5. Bennett, J., Lanning, S.: The Netflix prize. In Proceedings of KDD Cup and Workshop, vol. 2007, p. 35. New York (2007)
6. Bertsekas, D., Tsitsiklis, J.: Parallel and Distributed Computation. Numerical Methods. Prentice-Hall, Englewood Cliffs (1989)
7. Bot, R.I., Csetnek, E.R.: An inertial alternating direction method of multipliers. Minimax Theory Appl. **1**, 29–49 (2016)
8. Boţ, R.I., Csetnek, E.R., Hendrich, C.: Inertial Douglas–Rachford splitting for monotone inclusion problems. Appl. Math. Comput. **256**, 472–487 (2015)
9. Boţ, R.I., Csetnek, E.R., László, S.C.: An inertial forward–backward algorithm for the minimization of the sum of two nonconvex functions. EURO J. Comput. Optim. **4**(1), 3–25 (2016)

10. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**, 1–122 (2010)
11. Briceno-Arias, L.M., Combettes, P.L.: A monotone+ skew splitting model for composite monotone inclusions in duality. SIAM J. Optim. **21**(4), 1230–1250 (2011)
12. Cai, X., Guo, K., Jiang, F., Wang, K., Wu, Z., Han, D.: The developments of proximal point algorithms. J. Oper. Res. Soc. China **10**(2), 197–239 (2022)
13. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. J. Math. Imaging Vis. **40**, 120–145 (2011)
14. Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal-dual algorithm. Math. Program. **159**(1–2), 253–287 (2016)
15. Chen, C., Chan, R.H., Ma, S., Yang, J.: Inertial proximal ADMM for linearly constrained separable convex optimization. SIAM J. Imaging Sci. **8**(4), 2239–2267 (2015)
16. Chen, C., He, B., Yuan, X.: Matrix completion via alternating direction method. IMA J. Numer. Anal. **32**, 227–245 (2012)
17. Chen, Y., Nasrabadi, N.M., Tran, T.D.: Hyperspectral image classification using dictionary-based sparse representation. IEEE Trans. Geosci. Remote sens. **49**(10), 3973–3985 (2011)
18. Cheng, H., Liu, Z., Yang, L., Chen, X.: Sparse representation and learning in visual recognition: theory and applications. Signal Process. **93**(6), 1408–1425 (2013)
19. Dai, Y., Fletcher, R.: Projected Barzilai–Borwein methods for large-scale box-constrained quadratic programming. Numer. Math. **100**, 21–47 (2005)
20. Deng, Z., Liu, S.: Inertial generalized proximal Peaceman–Rachford splitting method for separable convex programming. Calcolo **58**(1), 1–30 (2021)
21. Dou, M., Li, H., Liu, X.: An inertial proximal Peaceman–Rachford splitting method. Sci. Sin. Math. **47**(2), 333–348 (2016)
22. Eaves, B.: On the basic theorem of complementarity. Math. Program. **1**, 68–75 (1971)
23. Facchinei, F., Pang, J.: Finite-Dimensional Variational Inequalities and Complementarity Problems. Springer, New York (2003)
24. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximations. Comput. Math. Appl. **2**, 16–40 (1976)
25. Gaines, B.R., Kim, J., Zhou, H.: Algorithms for fitting the constrained Lasso. J. Comput. Graph. Stat. **27**(4), 861–871 (2018)
26. Glowinski, R., Marrocco, A.: Approximation par éléments finis d'ordre un et résolution par pénalisation-dualité d'une classe de problèmes non linéaires. R.A.I.R.O. **R2**, 41–76 (1975)
27. Goldberg, K., Roeder, T., Gupta, D., Perkins, C.: Eigentaste: a constant time collaborative filtering algorithm. Inf. Retr. J. **4**(2), 133–151 (2001)
28. Han, D.: A survey on some recent developments of alternating direction method of multipliers. J. Oper. Res. Soc. China **10**, 1–52 (2022)
29. Han, D., He, H., Yang, H., Yuan, X.: A customized Douglas–Rachford splitting algorithm for separable convex minimization with linear constraints. Numer. Math. **127**, 167–200 (2014)
30. Han, D., Xu, W., Yang, H.: An operator splitting method for variational inequalities with partially unknown mappings. Numer. Math. **111**, 207–237 (2008)
31. Hansen, P.C., Nagy, J.G., O'leary, D.P.: Deblurring images: matrices, spectra, and filtering. SIAM (2006)
32. He, B., Liao, L., Wang, S.: Self-adaptive operator splitting methods for monotone variational inequalities. Numer. Math. **94**, 715–737 (2003)
33. He, H., Cai, X., Han, D.: A fast splitting method tailored for Dantzig selectors. Comput. Optim. Appl. **62**, 347–372 (2015)
34. He, H., Xu, H.K.: Splitting methods for split feasibility problems with application to dantzig selectors. Inverse Probl. **33**(5), 055003 (2017)
35. Hu, L., Zhang, W., Cai, X., Han, D.: A parallel operator splitting algorithm for solving constrained total-variation retinex. Inverse Probl. Imaging **14**(6), 1135 (2020)
36. Iordache, M.D., Bioucas-Dias, J.M., Plaza, A.: Sparse unmixing of hyperspectral data. IEEE Trans. Geosci. Remote. Sens. **49**(6), 2014–2039 (2011)
37. Liu, Y.J., Sun, D., Toh, K.C.: An implementable proximal point algorithmic framework for nuclear norm minimization. Math. Program. **133**(1), 399–436 (2012)
38. Lorenz, D.A., Pock, T.: An inertial forward-backward algorithm for monotone inclusions. J. Math. Imaging Vis. **51**(2), 311–325 (2015)
39. Morini, S., Porcelli, M., Chan, R.: A reduced Newton method for constrained linear least squares problems. J. Comput. Appl. Math. **233**, 2200–2212 (2010)
40. Polyak, B.T.: Some methods of speeding up the convergence of iteration methods. USSR Comput. Math. Math. Phys. **4**(5), 1–17 (1964)

41. Rockafellar, R.: On the maximal monotonicity of subdifferential mappings. Pac. J. Math. **33**(1), 209–216 (1970)
42. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Phys. D **60**, 227–238 (1992)
43. Tibshirani, R.: Regression shrinkage and selection via the Lasso. J. R. Stat. Soc. Ser. B **58**, 267–288 (1996)
44. Toh, K., Yun, S.: An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems. Pac. J. Optim. **6**, 615–640 (2010)
45. Varga, R.: Matrix Iterative Analysis. Prentice-Hall, Englewood Cliffs (1966)
46. Wang, X., Yuan, X.: The linearized alternating direction method of multipliers for Dantzig selector. SIAM J. Sci. Comput. **34**, A2792–A2811 (2012)
47. Wu, L., Yang, Y., Liu, H.: Nonnegative-lasso and application in index tracking. Comput. Stat. Data Anal. **70**, 116–126 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.