J. Nonlinear Var. Anal. 9 (2025), No. 6, pp. 927-945
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.9.2025.6.06

SIMPLIFIED PRIMAL-DUAL FORWARD-BACKWARD SPLITTING ALGORITHM FOR SOLVING STRUCTURED MONOTONE INCLUSION WITH APPLICATIONS

WENLI HUANG[1], HAIYANG LI[1,*], JIGEN PENG[1], YUCHAO TANG[1,2], HUIMIN HE[1]

[1] *School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China*
[2] *Department of Mathematics, Nanchang University, Nanchang 330031, China*

Abstract. This paper addresses the challenge of finding a zero of a structured monotone inclusion, which is closely related to convex minimization problems in signal and image processing. By defining a suitable product space, the monotone inclusion problem is transformed into the sum of two maximally monotone operators, one of which is cocoercive. Based on the preconditioned forward-backward splitting algorithm, we propose a new primal-dual splitting algorithm with a simple structure and prove its convergence with appropriate parameter conditions. In contrast to existing primal-dual forward-backward splitting algorithms, the proposed algorithm uses fewer variables and employs a reduced amount of parameters. Furthermore, we apply the algorithm to solve a class of convex minimization problems. Numerical experiments demonstrate the effectiveness and robustness of the proposed algorithm for image denoising problems.

Keywords. Cocoercive operator; Infimal convolution; Monotone inclusion; Primal-dual algorithm.

1. Introduction

In recent years, monotone inclusion problems have become increasingly important in the study of various convex minimization problems. The forward-backward splitting algorithm [1, 2], the Douglas-Rachford splitting algorithm [3, 4], the forward-backward-forward splitting algorithm [5, 6], the forward-reflected-backward splitting algorithm [7], and the reflected forward-backward splitting algorithm [8] are among the most widely employed methods for addressing monotone inclusions that involve the sum of two maximally monotone operators. To further improve the efficiency of the splitting algorithms, a new parameter-selection step, called Halpern-type extrapolation technique, was introduced into these traditional splitting algorithms and developed some Halpern-type algorithms [9, 10], in which the step sizes are self-adaptively chosen without the prior knowledge of Lipschitz constant of the cocoercive (or monotone Lipschitz continuous) operators. Very recently, Tan and Qin [11] introduced an inertial Halpern-type forward-backward splitting algorithm by combining Halpern type extrapolation, inertial accelerated strategy, and forward-backward splitting algorithm. As demonstrated in [12–14], driven by a variety of applications-including Huber total variation image restoration problems [15, 16], infimal convolution total variation image restoration problems [17, 18], and total generalized

[*]Corresponding author.
E-mail address: fplihaiyang@126.com (H. Li).
Received 4 June 2024; Accepted 20 January 2025; Published online 10 August 2025.

variation image restoration problems [19, 20]-highly structured monotone inclusion problems have gained significant attention.

In this paper, we focus on solving the following structured monotone inclusion.

Problem 1.1. Let $\mathscr{H}$ be a real Hilbert space, and let $m > 0$ be an integer. Let $A : \mathscr{H} \to 2^{\mathscr{H}}$ be maximally monotone operator, and let $C : \mathscr{H} \to \mathscr{H}$ be an μ^{-1}-cocoercive operator, for some $\mu > 0$. For every $i = 1, \ldots, m$, let $\mathscr{G}_i, \mathscr{X}_i, \mathscr{Y}_i$ be real Hilbert spaces, let $B_i : \mathscr{X}_i \to 2^{\mathscr{X}_i}$ and $D_i : \mathscr{Y}_i \to 2^{\mathscr{Y}_i}$ be maximally monotone operators, and let $L_i : \mathscr{H} \to \mathscr{G}_i, K_i : \mathscr{G}_i \to \mathscr{X}_i$ and $M_i : \mathscr{G}_i \to \mathscr{Y}_i$ be nonzero bounded linear operators.

$$\text{find } x \in H \text{ such that } z \in Ax + \sum_{i=1}^{m} L_i^* \left((K_i^* \circ B_i \circ K_i) \square (M_i^* \circ D_i \circ M_i) \right) (L_i x - r_i) + Cx, \tag{1.1}$$

together with its dual inclusion

$$\text{find} \quad \begin{cases} p_i \in \mathscr{X}_i, i = 1, \ldots, m, \\ q_i \in \mathscr{Y}_i, i = 1, \ldots, m, \\ y_i \in \mathscr{G}_i, i = 1, \ldots, m, \end{cases} \quad \text{such that} \quad \exists x \in \mathscr{H} : \begin{cases} -\sum_{i=1}^{m} L_i^* K_i^* p_i \in Ax + Cx, \\ K_i (L_i x - y_i) \in B_i^{-1} p_i, i = 1, \ldots, m, \\ M_i y_i \in D_i^{-1} q_i, i = 1, \ldots, m, \\ K_i^* p_i = M_i^* q_i, i = 1, \ldots, m. \end{cases} \tag{1.2}$$

Examining some special cases of (1.1) also presents interesting problems. In the following, we introduce some relevant work concerning problems (1.1)-(1.2). Let I denote the identity operator.

- Let $L_i^*(K_i^* \circ B_i \circ K_i) \square (M_i^* \circ D_i \circ M_i) L_i = B_i$. Davis and Yin [21] introduced a so-called three-operator splitting algorithm when $m = 1$. Later, Zong et al. [22] developed a four-operator splitting algorithm to solve $0 \in Ax + B_1 x + B_2 x + Cx$, which combines the Davis-Yin three-operator splitting algorithm and Ryu three-operator splitting algorithm [23].
- Let $K_i = I$ and $M_i = I$ for any $i = 1, 2, \cdots, m$. The preconditioned forward-backward splitting algorithm was introduced independently by Vu [24] and Condat [25]. Moreover, Boţ and Csetnek [26] presented a comprehensive survey on primal-dual splitting algorithms, exploring various intriguing applications including image denoising and deblurring, portfolio optimization, and clustering.
- Let $L_i = I$ for any $i = 1, 2, \cdots, m$. Becker and Combettes [27] proposed a primal-dual splitting algorithm to solve (1.1)-(1.2), which is derived from the inexact forward-backward-forward splitting algorithm [28]. The key idea is to transform the original monotone inclusion into a simplified formulation in a product Hilbert space which encourages a wealth of primal-dual splitting algorithms in solving diverse monotone inclusion problems; see, e.g., [29–32].
- For the general monotone inclusion (1.1), Boţ and Hendrich [33] proposed two different types of primal-dual splitting algorithms to solve (1.1)-(1.2) based on the forward-backward splitting algorithm and the forward-backward-forward splitting algorithm. Recently, Chen et al. [34] relaxed the parameters for the primal-dual forward-backward splitting algorithm [33] and designed another new algorithm for solving (1.1)-(1.2).

This new proposed algorithm is derived from the forward-backward-half-forward splitting algorithm [35].

The purpose of this paper is to introduce a new primal-dual splitting algorithm for solving (1.1)-(1.2). We reformulate monotone inclusions (1.1)-(1.2) into a sum of two maximally monotone operators in a suitable product space, where one of them is cocoercive. We develop a completely splitting algorithm derived from the preconditioned forward-backward splitting algorithm. Compared to the primal-dual forward-backward splitting algorithms studied in [33, 34], this new algorithm requires fewer variables, which can save computational storage space. At the same time, the proposed algorithm has fewer parameters.

The remainder of the paper is organized as follows. In Section 2, we review some fundamental elements of monotone operators and convex analysis. In Section 3, we present the main algorithm and prove its convergence. As an application, we discuss a convex minimization problem. In Section 4, we perform numerical experiments on image denoising problems to demonstrate the efficiency and effectiveness of the proposed algorithm. Finally, we draw some conclusions in Section 5.

2. PRELIMINARIES

Let $\mathscr{H}$ be a real Hilbert space with inner product $\langle\cdot,\cdot\rangle_{\mathscr{H}}$ and associated norm $\|\cdot\|_{\mathscr{H}} = \sqrt{\langle\cdot,\cdot\rangle}_{\mathscr{H}}$. Let $L:\mathscr{H}\to\mathscr{G}$ be a nonzero bounded linear operator, where $\mathscr{G}$ is a real Hilbert space. The adjoint operator of L is denoted by $L^*:\mathscr{G}\to\mathscr{H}$, which is defined by $\langle L^*y,x\rangle_{\mathscr{H}} = \langle y,Lx\rangle_{\mathscr{G}}, \forall x\in\mathscr{H}, y\in\mathscr{G}$. We denote by $\Gamma_0(\mathscr{H})$ the collection of all proper lower semi-continuous convex functions from $\mathscr{H}$ to $(-\infty,+\infty]$. Most of definitions are taken from [36].

Let $A:\mathscr{H}\to 2^{\mathscr{H}}$ be a set-valued operator. We denote by $graA = \{(x,u)\in\mathscr{H}\times\mathscr{H}|u\in Ax\}$ its graph, by $domA = \{x\in\mathscr{H}|Ax\neq\emptyset\}$ its domain, and by A^{-1} its inverse operator. A is said to be monotone if $\langle x-y,u-v\rangle\geq 0, \forall(x,u),(y,v)\in graA$. Furthermore, A is said to be maximally monotone, if there exists no other monotone operator B such that its graph properly contains $graA$. The parallel sum of two operators $A,B:\mathscr{H}\to 2^{\mathscr{H}}$ is defined by $A\Box B = (A^{-1}+B^{-1})^{-1}$.

The resolvent of A is defined by $J_A = (I+A)^{-1}$. If A is maximally monotone, then J_A is single-valued and firmly nonexpansive. Let $x\in\mathscr{H}$ and $\lambda>0$. Note that

$$J_{\lambda A}x + \lambda J_{\lambda^{-1}A^{-1}}(\frac{1}{\lambda}x) = x.$$

Let $B:\mathscr{H}\to\mathscr{H}$ be a single-valued operator. Recall that B is said to be μ^{-1}-cocoercive if $\langle x-y,Bx-By\rangle_{\mathscr{H}} \geq \frac{1}{\mu}\|Bx-By\|^2_{\mathscr{H}}$ for all $x,y\in\mathscr{H}$.

In the following, we recall some elements of convex analysis. Let $f:\mathscr{H}\to(-\infty,+\infty]$. The conjugate function of f is defined by $f^*(u) = \sup_{x\in\mathscr{H}}\{\langle u,x\rangle_{\mathscr{H}} - f(x)\}$. The subdifferential of a convex function f at $x\in\mathscr{H}$ is the set $\partial f(x) = \{v\in\mathscr{H}|f(y)\geq f(x)+\langle v,y-x\rangle_{\mathscr{H}}, \forall y\in\mathscr{H}\}$. If $f\in\Gamma_0(\mathscr{H})$, then ∂f is maximally monotone and $(\partial f)^{-1} = \partial f^*$. Let $f,g\in\Gamma_0(\mathscr{H})$. The infimal convolution is defined by $(f\Box g)(x) = \inf_{y\in\mathscr{H}}\{f(y)+g(x-y)\}$.

Let $f\in\Gamma_0(\mathscr{H})$, $x\in\mathscr{H}$, and $\lambda>0$. We denote by $prox_{\lambda f}$ the proximity operator of λf at x, which is defined by

$$prox_{\lambda f}(x) = \arg\min_{x\in\mathscr{H}}\left\{\frac{1}{2\lambda}\|y-x\|_2^2 + f(y)\right\}.$$

It follows from the first-order optimal condition of the proximity operator that $J_{\lambda\partial f} = prox_{\lambda f}$. The Moreau's decomposition shows that the relationship between the proximity operator of λf

and $\frac{1}{\lambda}f^*$, that is,

$$prox_{\lambda f}(x) + \lambda prox_{\frac{1}{\lambda}f^*}(\frac{1}{\lambda}x) = x,\ \forall x \in \mathscr{H}.$$

3. Simplified Primal-Dual Forward-Backward Splitting Algorithm

In this section, we present the main results. Firstly, we establish a lemma that offers an equivalent characterization of (1.1)-(1.2). Then, based on the lemma, we propose a primal-dual forward-backward splitting algorithm for solving (1.1)-(1.2) and prove the convergence of the proposed algorithm. Finally, we apply the obtained results to solve a class of convex minimization problems.

Lemma 3.1. *Let $\mathscr{H}$, $\mathscr{X}_i$, $\mathscr{Y}_i$, $\mathscr{G}_i$, A, C, B_i, D_i, L_i, K_i, M_i, $i = 1, \cdots, m$ be defined as in (1.1)-(1.2), and let*

$$\begin{aligned}
&\mathscr{X} := \mathscr{X}_1 \oplus \cdots \oplus \mathscr{X}_m, \mathscr{Y} := \mathscr{Y}_1 \oplus \cdots \oplus \mathscr{Y}_m, \mathscr{G} := \mathscr{G}_1 \oplus \cdots \oplus \mathscr{G}_m, \mathscr{K} := \mathscr{H} \oplus \mathscr{X} \oplus \mathscr{Y} \oplus \mathscr{G},\\
&\boldsymbol{p} = (p_1, \ldots, p_m) \in \mathscr{X}, \boldsymbol{q} = (q_1, \ldots, q_m) \in \mathscr{Y}, \boldsymbol{y} = (y_1, \ldots, y_m) \in \mathscr{G}, \boldsymbol{r} = (r_1, \ldots, r_m) \in \mathscr{G}\\
&B : \mathscr{X} \to 2^{\mathscr{X}} : \boldsymbol{p} \mapsto (B_1 p_1, \cdots, B_m p_m), D : \mathscr{Y} \to 2^{\mathscr{Y}}, \boldsymbol{q} \mapsto (D_1 q_1, \cdots, D_m q_m),\\
&\widetilde{M} : \mathscr{G} \to \mathscr{Y}, \boldsymbol{y} \mapsto (M_1 y_1, \cdots, M_m y_m), \widetilde{K} : \mathscr{G} \to \mathscr{X}, \boldsymbol{y} \mapsto (K_1 y_1, \cdots, K_m y_m),\\
&\boldsymbol{M} : \mathscr{K} \to 2^{\mathscr{K}}, (x, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{y}) \mapsto (-z + Ax) \times (B^{-1}\boldsymbol{p} + \widetilde{K}\boldsymbol{r}) \times D^{-1}\boldsymbol{q} \times 0\\
&L : \mathscr{H} \to \mathscr{G} : x \mapsto (L_1 x, \cdots, L_m x)\\
&\boldsymbol{S} : \mathscr{K} \to \mathscr{K}, (x, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{y}) \mapsto \left(L^*\widetilde{K}^*\boldsymbol{p}, -\widetilde{K}Lx + \widetilde{K}\boldsymbol{y}, -\widetilde{M}\boldsymbol{y}, -\widetilde{K}^*\boldsymbol{p} + \widetilde{M}^*\boldsymbol{q}\right)\\
&\boldsymbol{Q} : \mathscr{K} \to \mathscr{K}, (x, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{y}) \mapsto (Cx, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0})
\end{aligned}$$

Then the following conclusions hold:
(i) *$\boldsymbol{M}$ is maximally monotone.*
(ii) *$\boldsymbol{S}$ is monotone and l-Lipschitzian, where*

$$l = (\max\{2\sum_{i=1}^{m}\|K_i L_i\|^2, \sum_{i=1}^{m}\|K_i L_i\|^2 + 2\max_j \|K_j\|^2, 2\max_j \|M_j\|^2, \max_j \|M_j\|^2 + 2\max_j \|K_j\|^2\})^{\frac{1}{2}}.$$

(iii) *$\boldsymbol{Q}$ is μ^{-1}-cocoercive.*
(iv) *For any $\bar{x} \in \mathscr{H}$, $\bar{x}$ is a solution to* Problem 1.1 *if and only if there exists $(\bar{\boldsymbol{p}}, \bar{\boldsymbol{q}}, \bar{\boldsymbol{y}}) \in \mathscr{X} \oplus \mathscr{Y} \oplus \mathscr{G}$ such that $(\bar{x}, \bar{\boldsymbol{p}}, \bar{\boldsymbol{q}}, \bar{\boldsymbol{y}}) \in \operatorname{zer}(\boldsymbol{M} + \boldsymbol{S} + \boldsymbol{Q})$.*

Proof. (i) Since A, $\boldsymbol{B}$ and $\boldsymbol{D}$ are maximally monotone, it follows from [36, Proposition 20.22 and Proposition 20.23] that set-valued operator $\boldsymbol{M}$ is maximally monotone.

(ii) By taking $\boldsymbol{x} = (x, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{y}), \hat{\boldsymbol{x}} = (\hat{x}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{q}}, \hat{\boldsymbol{y}}) \in \mathscr{K}$, we obtain

$$\begin{aligned}
\langle \boldsymbol{x} - \hat{\boldsymbol{x}}, \boldsymbol{S}\boldsymbol{x} - \boldsymbol{S}\hat{\boldsymbol{x}} \rangle_{\mathscr{K}} =& \sum_{i=1}^{m} \langle (x - \hat{x}, L_i^* K_i^* (p_i - \hat{p}_i)\rangle_{\mathscr{H}} + \sum_{i=1}^{m} \langle p_i - \hat{p}_i, -K_i L_i (x - \hat{x}) + K_i (y_i - \hat{y}_i)\rangle_{\mathscr{X}_i}\\
&+ \sum_{i=1}^{m} \langle (q_i - \hat{q}_i), M_i (y_i - \hat{y}_i)\rangle_{\mathscr{Y}_i} + \sum_{i=1}^{m} \langle y_i - \hat{y}_i, K_i^* (p_i - \hat{p}_i) - M_i^* (q_i - \hat{q}_i)\rangle_{\mathscr{G}_i}\\
=& 0,
\end{aligned}$$

which means that $\boldsymbol{S}$ is monotone. It follows from the Cauchy-Schwarz inequality that

$$\begin{aligned}
&\|\boldsymbol{S}\boldsymbol{x}-\boldsymbol{S}\hat{\boldsymbol{x}}\|_{\mathscr{K}}\\
=&(\|\sum_{i=1}^{m}L_i^*K_i^*(p_i-\hat{p}_i)\|_{\mathscr{H}}^2+\sum_{i=1}^{m}\|K_iL_i(x-\hat{x})-K_i(y_i-\hat{y}_i)\|_{\mathscr{X}_i}^2+\sum_{i=1}^{m}\|M_i(y_i-\hat{y}_i)\|_{\mathscr{Y}_i}^2\\
&+\sum_{i=1}^{m}\|K_i^*(p_i-\hat{p}_i)-M_i^*(q_i-\hat{q}_i)\|_{\mathscr{G}_i}^2)^{\frac{1}{2}}\\
\leq&((\sum_{i=1}^{m}\|K_iL_i\|^2)\sum_{i=1}^{m}\|p_i-\hat{p}_i\|_{\mathscr{X}_i}^2+2\sum_{i=1}^{m}\|K_iL_i\|^2\|x-\hat{x}\|_{\mathscr{H}}^2+2\sum_{i=1}^{m}\|K_i\|^2\|y_i-\hat{y}_i\|_{\mathscr{G}_i}^2\\
&+\sum_{i=1}^{m}\|M_i\|^2\|y_i-\hat{y}_i\|_{\mathscr{G}_i}^2+2\sum_{i=1}^{m}\|K_i\|^2\|p_i-\hat{p}_i\|_{\mathscr{X}_i}^2+2\sum_{i=1}^{m}\|M_i\|^2\|q_i-\hat{q}_i\|_{\mathscr{Y}_i}^2)^{\frac{1}{2}}\\
\leq& l(\|x-\hat{x}\|_{\mathscr{H}}^2+\sum_{i=1}^{m}\|p_i-\hat{p}_i\|_{\mathscr{X}_i}^2+\sum_{i=1}^{m}\|q_i-\hat{q}_i\|_{\mathscr{Y}_i}^2+\sum_{i=1}^{m}\|y_i-\hat{y}_i\|_{\mathscr{G}_i}^2)^{\frac{1}{2}}\\
=& l\|\boldsymbol{x}-\hat{\boldsymbol{x}}\|_{\mathscr{K}}.
\end{aligned}$$

Hence, $\boldsymbol{S}$ is monotone and l-Lipschitzian.

(iii) Let $\boldsymbol{x}=(x,\boldsymbol{p},\boldsymbol{q},\boldsymbol{y})\in\mathscr{K}$ and $\hat{\boldsymbol{x}}=(\hat{x},\hat{\boldsymbol{p}},\hat{\boldsymbol{q}},\hat{\boldsymbol{y}})\in\mathscr{K}$. Since C is μ^{-1}-cocoercive, we have

$$\langle\boldsymbol{x}-\hat{\boldsymbol{x}},\boldsymbol{Q}\boldsymbol{x}-\boldsymbol{Q}\hat{\boldsymbol{x}}\rangle_{\mathscr{K}}=\langle x-\hat{x},Cx-C\hat{x}\rangle_{\mathscr{H}}\geq\mu^{-1}\|Cx-C\hat{x}\|_{\mathscr{H}}^2=\mu^{-1}\|\boldsymbol{Q}\boldsymbol{x}-\boldsymbol{Q}\hat{\boldsymbol{x}}\|_{\mathscr{K}}^2.$$

Thus $\boldsymbol{Q}$ is μ^{-1}-cocoercive.

(iv) Fixing $\bar{x}\in\mathscr{H}$, we have that

$$\bar{x}\text{ solves (1.1)}\Leftrightarrow\exists(\bar{x},\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{y}})\in\mathscr{H}\oplus\mathscr{X}\oplus\mathscr{Y}\oplus\mathscr{G}:\begin{cases}z-\sum_{i=1}^{m}L_i^*K_i^*\bar{p}_i\in A\bar{x}+C\bar{x},\\ K_i(L_i\bar{x}-\bar{y}_i-r_i)\in B_i^{-1}\bar{p}_i,i=1,\ldots,m,\\ M_i\bar{y}_i\in D_i^{-1}\bar{q}_i,i=1,\ldots,m,\\ K_i^*\bar{p}_i=M_i^*\bar{q}_i,i=1,\ldots,m.\end{cases}$$

$$\Leftrightarrow\exists(\bar{x},\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{y}})\in\operatorname{zer}(\boldsymbol{M}+\boldsymbol{S}+\boldsymbol{Q}).$$

Therefore, if $(\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{y}})$ is a solution of (1.2), then there exists $\bar{x}\in\mathscr{H}$ such that $(\bar{x},\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{y}})$ is a primal-dual solution to Problem 1.1. □

3.1. **Main algorithm.** In this subsection, we present the main algorithm to solve (1.1)-(1.2) and establish its convergence.

Theorem 3.1. *In monotone inclusion (1.1)-(1.2), suppose that*

$$z\in\operatorname{ran}\left(A+\sum_{i=1}^{m}L_i^*\left((K_i^*\circ B_i\circ K_i)\,\square\,(M_i^*\circ D_i\circ M_i)\right)(L_i\cdot-r_i)+C\right).$$

For any $i=1,\cdots,m$, let $\tau,\theta_{1,i},\theta_{2,i}$ and γ_i be strictly positive real numbers and $\{\lambda_n\}\subseteq[0,2-\frac{1}{2\beta}]$ satisfying the following conditions:

(i) $2\beta>1$, *where*

$$\beta=\mu^{-1}\left(\frac{1}{\tau}-\sum_{i=1}^{m}(\frac{1}{\theta_{1,i}}-(\frac{1}{\gamma_i}-\theta_{2,i}\|M_i\|^2)^{-1}\|K_i\|^2)^{-1}\|K_iL_i\|^2\right);$$

(ii)

$$(1-\alpha)\min_{i=1,\ldots,m}\left\{\frac{1}{\tau},\frac{1}{\theta_{1,i}},\frac{1}{\theta_{2,i}},\frac{1}{\gamma_i}\right\}>0,$$

where α is defined by

$$\alpha=\max_{j=1,2,\cdots,m}\left\{\sqrt{\tau\sum_{i=1}^{m}\theta_{1,i}\|K_iL_i\|^2}+\sqrt{\theta_{2,j}\gamma_j\|K_j\|^2},\sqrt{\theta_{2,j}\gamma_i\|M_j\|^2}+\sqrt{\theta_{2,j}\gamma_j\|K_j\|^2}\right\}.$$

(iii)

$$\sum_{n=0}^{+\infty}\lambda_n(2-\frac{1}{2\beta}-\lambda_n)=+\infty.$$

Let $x_0\in\mathscr{H}$, and for any $i=1,\ldots,m$, and let $p_{i,0}\in\mathscr{X}_i, q_{i,0}\in\mathscr{Y}_i$ and $y_{i,0}\in\mathscr{G}_i$. Set

$$(\forall n\geq 0)\left\lfloor\begin{array}{l}\widetilde{x}_n=J_{\tau A}(x_n-\tau(Cx_n+\sum_{i=1}^{m}L_i^*K_i^*p_{i,n}-z))\\ x_{n+1}=x_n+\lambda_n(\widetilde{x}_n-x_n)\\ \textit{For } i=1,\ldots,m\\ \left\lfloor\begin{array}{l}\widetilde{p}_{i,n}=J_{\theta_{1,i}B_i^{-1}}(\theta_{1,i}K_iL_i(2\widetilde{x}_n-x_n)-\theta_{1,i}K_ir_i+p_{i,n}-\theta_{1,i}K_iy_{i,n})\\ \widetilde{q}_{i,n}=J_{\theta_{2,i}D_i^{-1}}(q_{i,n}+\theta_{2,i}M_iy_{i,n})\\ \widetilde{y}_{i,n}=\gamma_iK_i^*(2\widetilde{p}_{i,n}-p_{i,n})-\gamma_iM_i^*(2\widetilde{q}_{i,n}-q_{i,n})+y_{i,n}-\gamma_iK_ir_i\\ p_{i,n+1}=p_{i,n}+\lambda_n(\widetilde{p}_{i,n}-p_{i,n})\\ q_{i,n+1}=q_{i,n}+\lambda_n(\widetilde{q}_{i,n}-q_{i,n})\\ y_{i,n+1}=y_{i,n}+\lambda_n(\widetilde{y}_{i,n}-y_{i,n})\end{array}\right.\end{array}\right. \tag{3.1}$$

Then there exists a primal-dual solution $(\bar{x},\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{y}})$ of (1.1)-(1.2) such that $x_n\rightharpoonup\bar{x}, p_{i,n}\rightharpoonup\bar{p}_i, q_{i,n}\rightharpoonup\bar{q}_i$, and $y_{i,n}\rightharpoonup\bar{y}_i$ for any $i=1,\ldots,m$ as $n\to+\infty$.

Proof. Let the real Hilbert space $\mathscr{K}=\mathscr{H}\oplus\mathscr{X}\oplus\mathscr{Y}\oplus\mathscr{G}$ and

$$\left\{\begin{array}{l}\boldsymbol{p}=(p_1,\ldots,p_m)\\ \boldsymbol{q}=(q_1,\ldots,q_m)\\ \boldsymbol{y}=(y_1,\ldots,y_m)\end{array}\right.\text{ and }\left\{\begin{array}{l}\boldsymbol{z}=(z_1,\ldots,z_m)\\ \boldsymbol{r}=(r_1,\ldots,r_m).\end{array}\right.$$

Define

$$\boldsymbol{V}:\mathscr{K}\to\mathscr{K},(x,\boldsymbol{p},\boldsymbol{q},\boldsymbol{y})\mapsto\left(\frac{x}{\tau},\frac{\boldsymbol{p}}{\theta_1},\frac{\boldsymbol{q}}{\theta_2},\frac{\boldsymbol{y}}{\gamma}\right)+\left(-L^*\widetilde{K}^*\boldsymbol{p},-\widetilde{K}Lx-\widetilde{K}\boldsymbol{y},\widetilde{M}\boldsymbol{y},-\widetilde{K}^*\boldsymbol{p}+\widetilde{M}^*\boldsymbol{q}\right).$$

Further, for positive real values $\tau,\theta_{1,i},\theta_{2,i},\gamma_i, i=1,\ldots,m$, we define the notations

$$\left\{\begin{array}{l}\frac{p}{\theta_1}=\left(\frac{p_1}{\theta_{1,1}},\ldots,\frac{p_m}{\theta_{1,m}}\right)\\ \frac{q}{\theta_2}=\left(\frac{q_1}{\theta_{2,1}},\ldots,\frac{q_m}{\theta_{2,m}}\right)\end{array}\right.,\quad\left\{\frac{y}{\gamma}=\left(\frac{y_1}{\gamma_1},\ldots,\frac{y_m}{\gamma_m}\right).\right.$$

Let

$$\left\{\begin{array}{l}\boldsymbol{p}_n=(p_{1,n},\ldots p_{m,n})\in\mathscr{X}\\ \boldsymbol{q}_n=(q_{1,n},\ldots,q_{m,n})\in\mathscr{Y}\\ \boldsymbol{y}_n=(y_{1,n},\ldots,y_{m,n})\in\mathscr{G}\end{array}\right.\quad\left\{\begin{array}{l}\widetilde{\boldsymbol{p}}_{\boldsymbol{n}}=(\widetilde{p}_{1,n},\ldots\widetilde{p}_{m,n})\in\mathscr{X}\\ \widetilde{\boldsymbol{q}}_{\boldsymbol{n}}=(\widetilde{q}_{1,n},\ldots,\widetilde{q}_{m,n})\in\mathscr{Y}\\ \widetilde{\boldsymbol{y}}_n=(\widetilde{y}_{1,n},\ldots,\widetilde{y}_{m,n})\in\mathscr{G}\end{array}\right.$$

and

$$\begin{cases} \boldsymbol{x}_n = (x_n, \boldsymbol{p}_n, \boldsymbol{q}_n, \boldsymbol{y}_n) \in \mathscr{K} \\ \widetilde{\boldsymbol{x}}_n = (\widetilde{x}_n, \widetilde{\boldsymbol{p}}_{\boldsymbol{n}}, \widetilde{\boldsymbol{q}}_{\boldsymbol{n}}, \widetilde{\boldsymbol{y}}_{\boldsymbol{n}}) \in \mathscr{K}. \end{cases}$$

We se that the iteration scheme in (3.1) is equivalent to

$$(\forall n \geq 0) \left\lfloor \begin{array}{l} \boldsymbol{V}(\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n) - \boldsymbol{Q}\boldsymbol{x}_n \in (\boldsymbol{M}+\boldsymbol{S})\widetilde{\boldsymbol{x}}_{\boldsymbol{n}} \\ \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \lambda_n(\widetilde{\boldsymbol{x}}_n - \boldsymbol{x}_n). \end{array} \right. \tag{3.2}$$

We introduce the notations $\boldsymbol{A}_{\mathscr{K}} := \boldsymbol{V}^{-1}(\boldsymbol{M}+\boldsymbol{S})$ and $\boldsymbol{B}_{\mathscr{K}} := \boldsymbol{V}^{-1}\boldsymbol{Q}$. Then, for any $n \geq 0$,

$$\begin{aligned} &\boldsymbol{V}(\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n) - \boldsymbol{Q}\boldsymbol{x}_n \in (\boldsymbol{M}+\boldsymbol{S})\widetilde{\boldsymbol{x}}_{\boldsymbol{n}} \\ \Leftrightarrow &\boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{Q}\boldsymbol{x}_n \in (\boldsymbol{V}+\boldsymbol{M}+\boldsymbol{S})\widetilde{\boldsymbol{x}}_{\boldsymbol{n}} \\ \Leftrightarrow &\boldsymbol{x}_n - \boldsymbol{V}^{-1}\boldsymbol{Q}\boldsymbol{x}_n \in \left(\mathrm{Id} + \boldsymbol{V}^{-1}(\boldsymbol{M}+\boldsymbol{S})\right)\widetilde{\boldsymbol{x}}_n \\ \Leftrightarrow &\widetilde{\boldsymbol{x}}_n = \left(\mathrm{Id} + \boldsymbol{V}^{-1}(\boldsymbol{M}+\boldsymbol{S})\right)^{-1}\left(\boldsymbol{x}_n - \boldsymbol{V}^{-1}\boldsymbol{Q}\boldsymbol{x}_n\right) \\ \Leftrightarrow &\widetilde{\boldsymbol{x}}_n = (\mathrm{Id} + \boldsymbol{A}_{\mathscr{K}})^{-1}(\boldsymbol{x}_n - \boldsymbol{B}_{\mathscr{K}}\boldsymbol{x}_n), \end{aligned}$$

which can be written as $\widetilde{\boldsymbol{x}}_{\boldsymbol{n}} = J_{\boldsymbol{A}_{\mathscr{K}}}(\boldsymbol{x}_n - \boldsymbol{B}_{\mathscr{K}}\boldsymbol{x}_n)$. Thus the iterative scheme in (3.2) becomes

$$(\forall n \geq 0) \left\lfloor \begin{array}{l} \widetilde{\boldsymbol{x}}_n = J_{\boldsymbol{A}_{\mathscr{K}}}(\boldsymbol{x}_n - \boldsymbol{B}_{\mathscr{K}}\boldsymbol{x}_n) \\ \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \lambda_n(\widetilde{\boldsymbol{x}}_{\boldsymbol{n}} - \boldsymbol{x}_n). \end{array} \right. \tag{3.3}$$

On the other hand, we have

$$\begin{aligned} &\langle \boldsymbol{x}, \boldsymbol{V}\boldsymbol{x}\rangle_{\mathscr{K}} \\ =&\frac{1}{\tau}\|x\|^2_{\mathscr{H}} + \sum_{i=1}^m \frac{1}{\theta_{1,i}}\|p_i\|^2_{\mathscr{X}_i} + \sum_{i=1}^m \frac{1}{\theta_{2,i}}\|q_i\|^2_{\mathscr{Y}_i} + \sum_{i=1}^m \frac{1}{\gamma_i}\|y_i\|^2_{\mathscr{G}_i} - 2\sum_{i=1}^m \langle x, L_i^* K_i^* p_i\rangle_{\mathscr{H}} \\ &\quad - 2\sum_{i=1}^m \langle p_i, K_i y_i\rangle_{\mathscr{X}_i} + 2\sum_{i=1}^m \langle q_i, M_i y_i\rangle_{\mathscr{G}_i} \\ \geq&\frac{1}{\tau}\|x\|^2_{\mathscr{H}} + \sum_{i=1}^m \frac{1}{\theta_{1,i}}\|p_i\|^2_{\mathscr{X}_i} + \sum_{i=1}^m \frac{1}{\theta_{2,i}}\|q_i\|^2_{\mathscr{Y}_i} + \sum_{i=1}^m \frac{1}{\gamma_i}\|y_i\|^2_{\mathscr{G}_i} - \frac{1}{\tau}\sqrt{\sum_{i=1}^m \tau\theta_{1,i}\|K_iL_i\|^2}\|x\|^2_{\mathscr{H}} \\ &- \sum_{j=1}^m \frac{1}{\theta_{1,j}}\left(\sqrt{\sum_{i=1}^m \tau\theta_{1,i}\|K_iL_i\|^2} + \sqrt{\gamma_j\theta_{1,j}\|K_j\|^2}\right)\|p_i\|^2_{\mathscr{X}_j} - \sum_{i=1}^m \frac{1}{\theta_{2,i}}\sqrt{\sum_{i=1}^m \gamma_i\theta_{2,i}\|M_i\|^2}\|q_i\|^2_{\mathscr{Y}_i} \\ &- \sum_{i=1}^m \frac{1}{\gamma_i}\left(\sqrt{\gamma_i\theta_{2,i}\|K_i\|^2} + \sqrt{\gamma_i\theta_{2,i}\|M_i\|^2}\right)\|_i\|^2_{\mathscr{G}_j} \\ \geq&(1-\alpha)\left(\frac{1}{\tau}\|x\|^2_{\mathscr{H}} + \sum_{i=1}^m \frac{1}{\theta_{1,i}}\|p_i\|^2_{\mathscr{X}_i} + \sum_{i=1}^m \frac{1}{\theta_{2,i}}\|q_i\|^2_{\mathscr{Y}_i} + \sum_{i=1}^m \frac{1}{\gamma_i}\|y_i\|^2_{\mathscr{G}_i}\right) \\ \geq&(1-\alpha)\min_{i=1,\dots,m}\left\{\frac{1}{\tau}, \frac{1}{\theta_{1,i}}, \frac{1}{\theta_{2,i}}, \frac{1}{\gamma_i}\right\}\|\boldsymbol{x}\|^2_{\mathscr{K}}. \end{aligned}$$

Define the Hilbert space $(\mathscr{K}_{\boldsymbol{V}}, \langle \cdot,\cdot \rangle_{\mathscr{K}_{\boldsymbol{V}}})$ as: For $\boldsymbol{x},\hat{\boldsymbol{x}} \in \mathscr{K}$,

$$\langle \boldsymbol{x},\hat{\boldsymbol{x}} \rangle_{\mathscr{K}_{\boldsymbol{V}}} = \langle \boldsymbol{x},\boldsymbol{V}\hat{\boldsymbol{x}} \rangle_{\mathscr{K}} \text{ and } \|\boldsymbol{x}\|_{\mathscr{K}_{\boldsymbol{V}}} = \sqrt{\langle \boldsymbol{x},\boldsymbol{V}\boldsymbol{x} \rangle_{\mathscr{K}}}.$$

Since $\boldsymbol{V}$ is self-adjoint and strongly positive, one can easily see that weak and strong convergence in $\mathscr{K}_{\boldsymbol{V}}$ are equivalent with weak and strong convergence in $\mathscr{K}$, respectively. In the following, we prove that $\boldsymbol{B}_{\mathscr{K}}$ is β-cocoercive on $\mathscr{K}_{\boldsymbol{V}}$. In fact, letting $\boldsymbol{x},\hat{\boldsymbol{x}} \in \mathscr{K}_{\boldsymbol{V}}$, we have

$$\begin{aligned}
&\|\boldsymbol{B}_{\mathscr{K}}\boldsymbol{x} - \boldsymbol{B}_{\mathscr{K}}\hat{\boldsymbol{x}}\|^2_{\mathscr{K}_{\boldsymbol{V}}} \\
&= \langle \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\hat{\boldsymbol{x}}, \boldsymbol{V}^{-1}\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{V}^{-1}\boldsymbol{Q}\hat{\boldsymbol{x}} \rangle_{\mathscr{K}} \\
&= \langle Cx - C\hat{x}, (\frac{1}{\tau}Id - L^*\widetilde{K}^*(\frac{1}{\theta_1}Id - \widetilde{K}(\frac{1}{\gamma}Id - \theta_2\widetilde{M}^*\widetilde{M})^{-1}\widetilde{K}^*)^{-1}\widetilde{K}L)^{-1}(Cx - C\hat{x})\rangle_{\mathscr{H}} \\
&\le (\frac{1}{\tau} - \sum_{i=1}^m (\frac{1}{\theta_{1,i}} - (\frac{1}{\gamma_i} - \theta_{2,i}\|M_i\|^2)^{-1}\|K_i\|^2)^{-1}\|K_iL_i\|^2)^{-1}\langle Cx - C\hat{x}, Cx - C\hat{x}\rangle_{\mathscr{H}} \\
&= (\frac{1}{\tau} - \sum_{i=1}^m (\frac{1}{\theta_{1,i}} - (\frac{1}{\gamma_i} - \theta_{2,i}\|M_i\|^2)^{-1}\|K_i\|^2)^{-1}\|K_iL_i\|^2)^{-1}\|Cx - C\hat{x}\|^2_{\mathscr{H}}.
\end{aligned}$$

It follows from the above inequality that

$$\begin{aligned}
\langle \boldsymbol{x} - \hat{\boldsymbol{x}}, \boldsymbol{B}_{\mathscr{K}}\boldsymbol{x} - \boldsymbol{B}_{\mathscr{K}}\hat{\boldsymbol{x}} \rangle_{\mathscr{K}_{\boldsymbol{V}}} &= \langle \boldsymbol{x} - \hat{\boldsymbol{x}}, \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\hat{\boldsymbol{x}} \rangle_{\mathscr{K}} \\
&= \langle x - \hat{x}, Cx - C\hat{x} \rangle_{\mathscr{H}} \\
&\ge \mu^{-1}\|Cx - C\hat{x}\|^2_{\mathscr{H}} \\
&= \beta\, \|\boldsymbol{B}_{\mathscr{K}}\boldsymbol{x} - \boldsymbol{B}_{\mathscr{K}}\hat{\boldsymbol{x}}\|^2_{\mathscr{K}_{\boldsymbol{V}}},
\end{aligned}$$

where

$$\beta = \mu^{-1}(\frac{1}{\tau} - \sum_{i=1}^m (\frac{1}{\theta_{1,i}} - (\frac{1}{\gamma_i} - \theta_{2,i}\|M_i\|^2)^{-1}\|K_i\|^2)^{-1}\|K_iL_i\|^2).$$

Since $2\beta > 1$, then iteration scheme (3.3) could be viewed as a special case of the forward-backward splitting algorithm. By [36, Corollary 28.9], we see that iterative sequences $\{\boldsymbol{x}_n\}$ converges weakly to a point $\bar{\boldsymbol{x}} = (\bar{x}, \overline{\boldsymbol{p}}, \overline{\boldsymbol{q}}, \overline{\boldsymbol{y}})$ in $\operatorname{zer}(\boldsymbol{A}_{\mathscr{K}} + \boldsymbol{B}_{\mathscr{K}})$. It is observed that

$$\operatorname{zer}(\boldsymbol{A}_{\mathscr{K}} + \boldsymbol{B}_{\mathscr{K}}) = \operatorname{zer}(\boldsymbol{M} + \boldsymbol{S} + \boldsymbol{Q}).$$

Then, we obtain that $x_n \rightharpoonup \bar{x}$, $p_{i,n} \rightharpoonup \bar{p}_i$, $q_{i,n} \rightharpoonup \bar{q}_i$ and $y_{i,n} \rightharpoonup \bar{y}_i$ for $i = 1,\dots,m$ as $n \to +\infty$. This completes the proof. □

Remark 3.1. The following primal-dual forward-backward splitting algorithm was proposed in [33] and [34].

$$(\forall n \geq 0) \quad \left\lfloor \begin{array}{l} \widetilde{x}_n = J_{\tau A}(x_n - \tau(Cx_n + \sum_{i=1}^{m} L_i^* v_{i,n} - z)) \\ \text{for } i = 1,\ldots,m \\ \left\lfloor \begin{array}{l} \widetilde{p}_{i,n} = J_{\theta_{1,i}B_i^{-1}}(p_{i,n} + \theta_{1,i}K_i z_{i,n}) \\ \widetilde{q}_{i,n} = J_{\theta_{2,i}D_i^{-1}}(q_{i,n} + \theta_{2,i}M_i y_{i,n}) \\ u_{1,i,n} = z_{i,n} + \gamma_{1,i}(K_i^*(p_{i,n} - 2\widetilde{p}_{i,n}) + v_{i,n} + \sigma_i(L_i(2\widetilde{x}_n - x_n) - r_i)) \\ u_{2,i,n} = y_{i,n} + \gamma_{2,i}(M_i^*(q_{i,n} - 2\widetilde{q}_{i,n}) + v_{i,n} + \sigma_i(L_i(2\widetilde{x}_n - x_n) - r_i)) \\ \widetilde{z}_{i,n} = \dfrac{1+\sigma_i\gamma_{2,i}}{1+\sigma_i(\gamma_{1,i}+\gamma_{2,i})}(u_{1,i,n} - \dfrac{\sigma_i\gamma_{1,i}}{1+\sigma_i\gamma_{2,i}}u_{2,i,n}) \\ \widetilde{y}_{i,n} = \dfrac{1}{1+\sigma_i\gamma_{2,i}}(u_{2,i,n} - \sigma_i\gamma_{2,i}\widetilde{z}_{i,n}) \\ \widetilde{v}_{i,n} = v_{i,n} + \sigma_i(L_i(2\widetilde{x}_n - x_n) - r_i - \widetilde{z}_{i,n} - \widetilde{y}_{i,n}) \end{array} \right. \\ x_{n+1} = x_n + \lambda_n(\widetilde{x}_n - x_n) \\ \text{for } i = 1,\ldots,m \\ \left\lfloor \begin{array}{l} p_{i,n+1} = p_{i,n} + \lambda_n(\widetilde{p}_{i,n} - p_{i,n}) \\ q_{i,n+1} = q_{i,n} + \lambda_n(\widetilde{q}_{i,n} - q_{i,n}) \\ z_{i,n+1} = z_{i,n} + \lambda_n(\widetilde{z}_{i,n} - z_{i,n}) \\ y_{i,n+1} = y_{i,n} + \lambda_n(\widetilde{y}_{i,n} - y_{i,n}) \\ v_{i,n+1} = v_{i,n} + \lambda_n(\widetilde{v}_{i,n} - v_{i,n}), \end{array} \right. \end{array} \right. \tag{3.4}$$

where, for any $i = 1,\ldots,m$, $\tau, \theta_{1,i}, \theta_{2,i}, \gamma_{1,i}, \gamma_{2,i}$, and σ_i are strictly positive real numbers, and $\lambda_n > 0$. Iterative algorithm (3.1) requires fewer iterations to update variables and uses fewer iterative parameters than (3.4). Therefore, (3.1) is simpler than (3.4).

3.2. **Applications to convex minimization problems.** In this subsection, we apply the proposed algorithms to solve the following convex minimization problem.

Problem 3.1. Let $\mathscr{H}$ be a real Hilbert space, let $z \in \mathscr{H}$ and $h : \mathscr{H} \to R$ be differentiable with μ-Lipschitzian gradient for some $\mu > 0$. Let $f \in \Gamma_0(\mathscr{H})$. For every $i = 1,\cdots,m$, let $\mathscr{G}_i, \mathscr{X}_i, \mathscr{Y}_i$ be real Hilbert spaces, $r_i \in \mathscr{G}_i$, $g_i \in \Gamma_0(\mathscr{X}_i)$ and $l_i \in \Gamma_0(\mathscr{Y}_i)$ and consider the nonzero linear bounded operators $L_i : \mathscr{H} \to \mathscr{G}_i, K_i : \mathscr{G}_i \to \mathscr{X}_i$ and $M_i : \mathscr{G}_i \to \mathscr{Y}_i$. The primal convex minimization problem is

$$\min_{x\in\mathscr{H}} \left\{ f(x) + \sum_{i=1}^{m} ((g_i \circ K_i) \square (l_i \circ M_i))(L_i x - r_i) + h(x) - \langle x, z\rangle \right\}, \tag{3.5}$$

together with its conjugate dual problem

$$\max_{(\boldsymbol{p},\boldsymbol{q})\in\mathscr{X}\oplus\mathscr{Y}, K_i^* p_i = M_i^* q_i, i=1,\ldots,m} \left\{ -(f^* \square h^*)\left(z - \sum_{i=1}^{m} L_i^* K_i^* p_i\right) - \sum_{i=1}^{m} [g_i^*(p_i) + l_i^*(q_i) + \langle p_i, K_i r_i\rangle] \right\}. \tag{3.6}$$

Let $(\bar{x},\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{y}}) \in \mathscr{H} \oplus \mathscr{X} \oplus \mathscr{Y} \oplus \mathscr{G}$ be a solution to the following primal-dual system of monotone inclusions

$$\begin{aligned} & z - \sum_{i=1}^{m} L_i^* K_i^* \bar{p}_i \in \partial f(\bar{x}) + \nabla h(\bar{x}) \\ & \text{and } K_i\left(L_i\bar{x} - \bar{y}_i - r_i\right) \in \partial g_i^*\left(\bar{p}_i\right), M_i\bar{y}_i \in \partial l_i^*\left(\bar{q}_i\right), K_i^*\bar{p}_i = M_i^*\bar{q}_i, i = 1,\ldots,m, \end{aligned} \tag{3.7}$$

which means that $\bar{x}$ is an optimal solution to (3.5) and $(\overline{\boldsymbol{p}},\overline{\boldsymbol{q}})$ is an optimal solution to (3.6).

For primal-dual system (3.7), the iterative sequence proposed in (3.1) read as:

Algorithm 3.1. Let $x_0 \in \mathscr{H}$, and for any $i = 1,\ldots,m$, let $p_{i,0} \in \mathscr{X}_i, q_{i,0} \in \mathscr{Y}_i$ and $y_{i,0} \in \mathscr{G}_i$. Define

$$(\forall n \geq 0) \left\lfloor \begin{array}{l} \widetilde{x}_n = \operatorname{prox}_{\tau f}\left(x_n - \tau\left(\nabla h(x_n) + \sum_{i=1}^{m} L_i^* K_i^* p_{i,n} - z\right)\right) \\ x_{n+1} = x_n + \lambda_n\left(\widetilde{x}_n - x_n\right) \\ \text{For } i = 1,\ldots,m \\ \left\lfloor \begin{array}{l} \widetilde{p}_{i,n} = \operatorname{prox}_{\theta_{1,i} g_i^*}\left(\theta_{1,i}K_iL_i(2\widetilde{x}_n - x_n) - \theta_{1,i}K_ir_i + p_{i,n} - \theta_{1,i}K_iy_{i,n}\right) \\ \widetilde{q}_{i,n} = \operatorname{prox}_{\theta_{2,i} l_i^*}\left(q_{i,n} + \theta_{2,i}M_iy_{i,n}\right) \\ \tilde{y}_{i,n} = \gamma_i K_i^*(2\widetilde{p}_{i,n} - p_{i,n}) - \gamma_i M_i^*(2\widetilde{q}_{i,n} - q_{i,n}) + y_{i,n} - \gamma K_i r_i \\ p_{i,n+1} = p_{i,n} + \lambda_n\left(\widetilde{p}_{i,n} - p_{i,n}\right) \\ q_{i,n+1} = q_{i,n} + \lambda_n\left(\widetilde{q}_{i,n} - q_{i,n}\right) \\ y_{i,n+1} = y_{i,n} + \lambda_n\left(\widetilde{y}_{i,n} - y_{i,n}\right) \end{array} \right. \end{array} \right.$$

The convergence of Algorithm 3.1 is presented in the following theorem.

Theorem 3.2. *For convex minimization problem (3.5), suppose that*

$$z \in \operatorname{ran}\left(\partial f + \sum_{i=1}^{m} L_i^*\left(\left(K_i^* \circ \partial g_i \circ K_i\right) \square \left(M_i^* \circ \partial l_i \circ M_i\right)\right)\left(L_i \cdot - r_i\right) + \nabla h\right)$$

and consider the sequences generated by Algorithm 3.1. For any $i = 1,\ldots,m$, let $\tau, \theta_{1,i}, \theta_{2,i}, \gamma_{1,i}$, $\gamma_{2,i}$, and σ_i be strictly positive real numbers and $\{\lambda_n\}$ satisfy the conditions in Theorem 3.1. Then there exists an optimal solution $\bar{x}$ to (3.5) and optimal solution $(\overline{\boldsymbol{p}},\overline{\boldsymbol{q}})$ to (3.6) such that $x_n \rightharpoonup \bar{x}$ and for $i = 1,\ldots,m, p_{i,n} \rightharpoonup \bar{p}_i, q_{i,n} \rightharpoonup \bar{q}_i$ as $n \to +\infty$.

Proof. In Theorem 3.1, let

$$A = \partial f, C = \nabla h, \text{ and } B_i = \partial g_i, D_i = \partial l_i, i = 1,\ldots,m. \tag{3.8}$$

According to Theorem 20.25 of [36], the operators in (3.8) are maximally monotone.

On the other hand, we have $B_i^{-1} = \partial g_i^*$ and $D_i^{-1} = \partial l_i^*$ for $i = 1,\ldots,m$. Moreover, by Baillon-Haddad theorem, $C = \nabla h$ is μ^{-1}-cocoercive. By Theorem 3.1, we have $x_n \rightharpoonup \bar{x}$ and for $i = 1,\ldots,m, p_{i,n} \rightharpoonup \bar{p}_i, q_{i,n} \rightharpoonup \bar{q}_i$. □

Remark 3.2. Let $\mathscr{H}$ be $\widetilde{n}-$dimensional real Euclidean space, $\mathscr{X}_i$ be c_i-dimensional real Euclidean space, $\mathscr{Y}_i$ be d_i-dimensional real Euclidean space, $\mathscr{G}_i$ be m_i-dimensional real Euclidean space, the multiplication computation of Algorithm 3.1 be $m(2c_i\widetilde{n} + 2c_im_i + 2d_im_i)$, and of (3.4) [34] be $m(c_i\widetilde{n} + 2c_im_i + 2d_im_i + m_i\widetilde{n})$. Therefore, the actual computational complexity of the proposed algorithm and Algorithm 3.4 can only be compared in the context of specific practical problems.

4. Numerical Experiments

In this section, we present some numerical experiments to demonstrate the effectiveness of the proposed algorithm for image denoising problems. We compare the proposed algorithm with (3.4) by using the parameter conditions from [34], which we refer to as FB_CLTD. All numerical experiments are implemented on MATLAB R2017a on a personal computer with Intel Core i7-10870H CPU 2.21GHz and 16 GB memory. The code for this paper is available for download at the GitHub repository: https://github.com/hhaaoo1331/Simpli-ed-PDFB.

We mainly focus on the following constrained image denoising models:

$$(\ell_2-\text{IC}) \quad \min_{x\in R^{kl}} \left\{ \frac{1}{2}\|x-b\|^2 + ((\alpha_1\|\cdot\|_1 \circ \mathscr{D}_1)\,\square\,(\alpha_2\|\cdot\|_1\circ\mathscr{D}_2))\,(x) \right\}, \ s.t. \quad x\in C, \tag{4.1}$$

and

$$(\ell_2-\text{MIC}) \quad \min_{x\in^{kl}} \left\{ \frac{1}{2}\|x-b\|^2 + ((\alpha_1\|\cdot\|_1)\,\square\,(\alpha_2\|\cdot\|_1\circ L_1))\,(\mathscr{D}_1 x) \right\}, \ s.t. \quad x\in C, \tag{4.2}$$

where $\alpha_1 > 0, \alpha_2 > 0$ are the regularization parameters, and $C = \{x \in R^{kl} | 0 \le x_j \le 255, j = 1,2,\cdots,kl\}$. For detailed definitions of (4.1) and (4.2), we refer to [33, 34]. It is easy to check that (4.1) and (4.2) are special cases of convex minimization problem (3.5). For example, for the ℓ_2-IC and ℓ_2-MIC, let $h(x) = \frac{1}{2}\|x-b\|^2$. Then $\nabla h(x) = x-b$ and $\mu = 1$.

We use the peak-signal-to-noise (PSNR) and the structural similarity index (SSIM) [37] to evaluate the quality of the restored images, which are estimated as follows:

$$PSNR = 20\log_{10}\frac{255\sqrt{kl}}{\|x-y\|},$$

and

$$SSIM = \frac{(2\mu_1\mu_2+c_1)(2\sigma_{12}+c_2)}{(2\mu_1^2\mu_2^2+c_1)(\sigma_1^2+\sigma_2^2+c_2)},$$

where $x \in R^{kl}$ is the column vector converted from the original image $\bar{x}$ with size of $k\times l$, $y \in R^{kl}$ is the column vector converted from the restored image $\bar{y}$, $c_1 > 0$ and $c_2 > 0$ are small constants, μ_1 and μ_2 are the mean values of $\bar{x}$ and $\bar{y}$, respectively; σ_1 and σ_2 are the variances of $\bar{x}$ and $\bar{y}$, respectively; σ_{12} is the covariance of $\bar{x}$ and $\bar{y}$.

In the following experiments, we select two images as test images, shown in Figure 1, and add Gaussian noise with mean zero and standard deviation of σ to the original image. The criterion for stopping all algorithms is that the relative error of two consecutive iterations satisfies the following inequality

$$\frac{\|x_{n+1}-x_n\|}{\|x_n\|} < \varepsilon,$$

where $\varepsilon > 0$ is a given positive constant. In the whole experiments, we choose $\varepsilon = 10^{-5}$. The regularization parameters α_1 and α_2 are listed in Table 1.

TABLE 1. Selection of the regularization parameters α_1 and α_2.

Image	Model	$\sigma = 15$		$\sigma = 25$		$\sigma = 50$	
		α_1	α_2	α_1	α_2	α_1	α_2
Castle	$\ell_2 - \text{IC}$	7.7	20.7	14.8	30.2	35.8	123.5
	$\ell_2 - \text{MIC}$	7.7	21.5	14.8	51.2	35.7	115.9
Building	$\ell_2 - \text{IC}$	6.1	24.8	12.5	32.6	31.6	87.8
	$\ell_2 - \text{MIC}$	6.1	27.4	12.5	49	31.8	139.8

In the first experiment, we discuss the influence of the selection of iterative parameters on the convergence speed of the proposed algorithm, and the standard deviation of Gaussian noise σ is set to 15. The numerical results are shown in Table 2. According to the convergence criteria of Algorithm 3.1, we provide the range of parameter values that need to be satisfied in Table 2. At the same time, we list some specific parameter combinations in Table 3 to meet the requirement.

TABLE 2. The range of parameter values of Algorithm 3.1.

Model	Parameter
$\ell_2 - \text{IC}$	$\lambda_n \in (0, 2 - \frac{1}{2\beta}), 2\beta > 1, \beta = \frac{1}{\tau} - (\frac{1}{\theta_1} - (\frac{1}{\gamma} - \theta_2 \times 5.6133^2)^{-1} \times 2.8072^2)^{-1} \times 2.8072^2$, $\alpha = \max\{2.8072\sqrt{\tau\theta_1} + 2.8072\sqrt{\gamma\theta_2}, 5.6133\sqrt{\theta_2\gamma} + 2.8072\sqrt{\theta_2\gamma}\} < 1$
$\ell_2 - \text{MIC}$	$\lambda_n \in (0, 2 - \frac{1}{2\beta}), 2\beta > 1, \beta = \frac{1}{\tau} - (\frac{1}{\theta_1} - (\frac{1}{\gamma} - \theta_2 \times 1.9926^2)^{-1})^{-1} \times 2.8072^2$, $\alpha = \max\{2.8072\sqrt{\tau\theta_1} + \sqrt{\gamma\theta_2}, 1.9926\sqrt{\theta_2\gamma} + \sqrt{\theta_2\gamma}\} < 1$

TABLE 3. The parameters selection of Algorithm 3.1, where $\lambda_{max} = 2 - \frac{1}{2\beta}$.

Model	Case	θ_1	θ_2	τ	γ	λ_{max}
$\ell_2 - \text{IC}$	1	0.1	0.1	0.2	0.1	1.87
	2	0.1	0.1	0.4	0.1	1.68
	3	0.2	0.1	0.1	0.1	1.93
	4	0.3	0.1	0.1	0.1	1.92
	5	0.4	0.1	0.1	0.1	1.87
	6	0.5	0.1	0.1	0.1	1.30
$\ell_2 - \text{MIC}$	1	0.1	0.4	0.6	0.1	1.42
	2	0.1	0.3	0.3	0.3	1.80
	3	0.1	0.5	0.4	0.2	1.70
	4	0.1	0.7	0.6	0.1	1.42
	5	0.2	0.2	0.4	0.2	1.40
	6	0.4	0.5	0.1	0.1	1.92

It can be seen from Table 4 that, under the given parameter selection, the PSNR and SSIM values of the restored images by Algorithm 3.1 is almost consistently, and the difference in terms of the number of iterations required for the algorithm is not significant. Therefore, in practical applications, we can easily choose appropriate parameters to ensure the convergence speed of the algorithm. In the following experiment, for the ℓ_2-IC model, we select the parameter combination of Case 4, and for the ℓ_2-MIC model, we selected the parameter combination of Case 3, respectively.

(a) (b)

FIGURE 1. Test images. (a) 481×321 "Castle" image, (b) 493×517 "Building" image.

TABLE 4. Numerical results of Algorithm 3.1 with different parameters in terms of the PSNR, SSIM, and number of iterations (Iter).

Model	Case	Castle			Building		
		PSNR	SSIM	Iter	PSNR	SSIM	Iter
ℓ_2-IC	1	30.5400	0.8409	612	28.3654	0.8405	656
	2	30.5401	0.8409	620	28.3651	0.8405	670
	3	30.5400	0.8408	585	28.3652	0.8405	615
	4	30.5400	0.8408	581	28.3652	0.8405	610
	5	30.5400	0.8408	585	28.3652	0.8405	614
	6	30.5400	0.8409	640	28.3660	0.8405	699
ℓ_2-MIC	1	30.5449	0.8411	304	28.3671	0.8405	320
	2	30.5458	0.8412	379	28.3676	0.8405	416
	3	30.5464	0.8412	292	28.3676	0.8405	298
	4	30.5454	0.8413	334	28.3680	0.8405	368
	5	30.5427	0.8410	377	28.3662	0.8405	401
	6	30.5478	0.8413	317	28.3681	0.8405	340

In the second experiment, we compare the proposed Algorithm 3.1 with the FB_CLTD for solving ℓ_2-IC and ℓ_2-MIC. The test images are added by Gaussian noise with mean zero and standard deviation of $\sigma = 15, 25$ and 50, respectively. The obtained results in terms of PSNR, SSIM, the number of iteration and CPU time are presented in Table 5. We can observe

TABLE 5. Numerical results of the compared algorithms in terms of the PSNR, SSIM, number of iterations (Iter) and CPU time (in seconds).

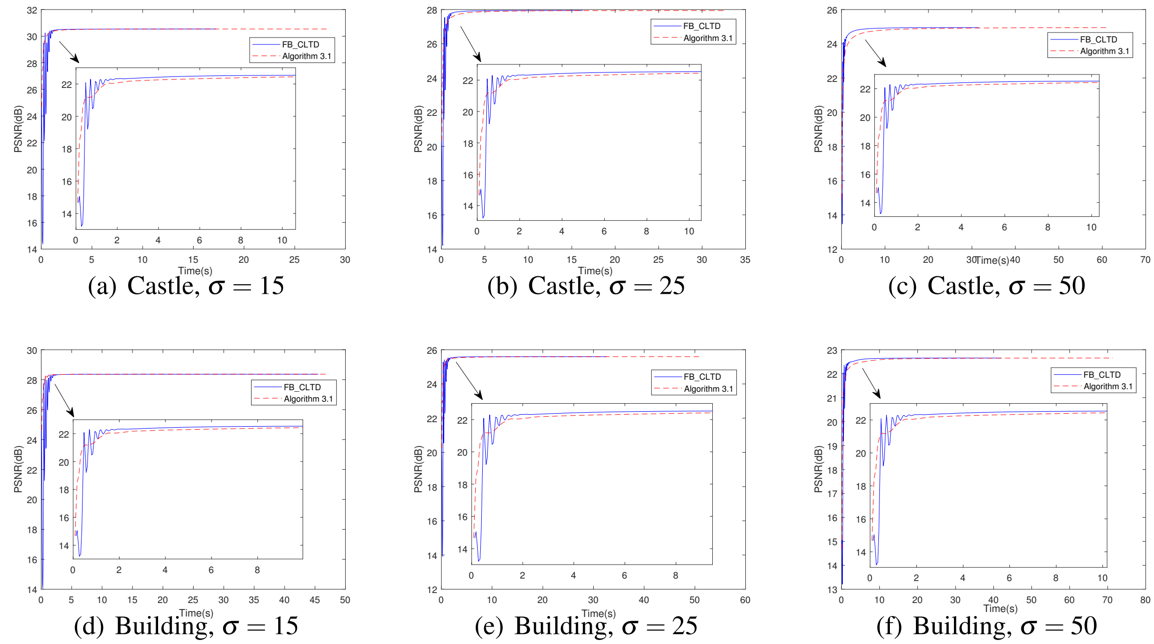
Image	Model	σ	FB_CLTD				Algorithm 3.1			
			PSNR	SSIM	Iter	Time	PSNR	SSIM	Iter	Time
Castle	ℓ_2-IC	15	30.5393	0.8409	590	17.2	30.5400	0.8408	581	28.1
		25	27.9452	0.7812	551	16.3	27.9422	0.7812	669	32.5
		50	24.9487	0.7039	1044	31.6	24.9441	0.7047	1252	61.5
	ℓ_2-MIC	15	30.5449	0.8411	403	12.9	30.5464	0.8412	292	10.1
		25	27.9400	0.7799	604	19.0	27.9457	0.7803	457	15.9
		50	24.9418	0.7017	1085	34.2	24.9501	0.7021	824	28.8
Building	ℓ_2-IC	15	28.3634	0.8404	903	45.6	28.3652	0.8405	610	47.2
		25	25.5958	0.7331	705	32.6	25.5965	0.7331	680	50.9
		50	22.6542	0.5552	850	42.1	22.6535	0.5552	939	71.4
	ℓ_2-MIC	15	28.3669	0.8405	421	21.1	28.3676	0.8405	298	16.1
		25	25.6005	0.7332	551	27.9	25.6019	0.7333	415	22.6
		50	22.6731	0.5565	1057	53.1	22.6752	0.5566	836	45.4

that the quality evaluation indicators PSNR and SSIM of the restored images by FB_CLTD and Algorithm 3.1 are almost the same. For ℓ_2-IC, Algorithm 3.1 requires more time than FB_CLTD regardless of the number of iterations. For ℓ_2-MIC, Algorithm 3.1 exhibits fewer iterations and less total computation time than FB_CLTD. It has been observed that there is a proportional relationship between the time consumed in each iteration of Algorithm 3.1 and that of FB_CLTD, i.e.,

$$\frac{Time_{Algorithm3.1}}{Iter_{Algorithm3.1}} \propto \frac{Time_{FB\_CLTD}}{Iter_{FB\_CLTD}}.$$

This phenomenon can be attributed to the fact that, as discussed in Remark 3.2: For ℓ_2-IC, Algorithm 3.1 performs $12k^2l^2$ multiplications per iteration, which is $4k^2l^2$ more than FB_CLTD, leading to longer computation time; and for ℓ_2-MIC, in each iteration, the multiplication computation of both algorithm is $8k^2l^2$. Therefore, in each iteration, when the evaluations of the proximity and the multiplication with the identity matrix I are negligible, with solving the IC model, the time of Algorithm 3.1 for multiplication calculation is about 1.5 times that of FB_CLTD; when solving the MIC model, the multiplication computation time is nearly identical between FB_CLTD and Algorithm 3.1.

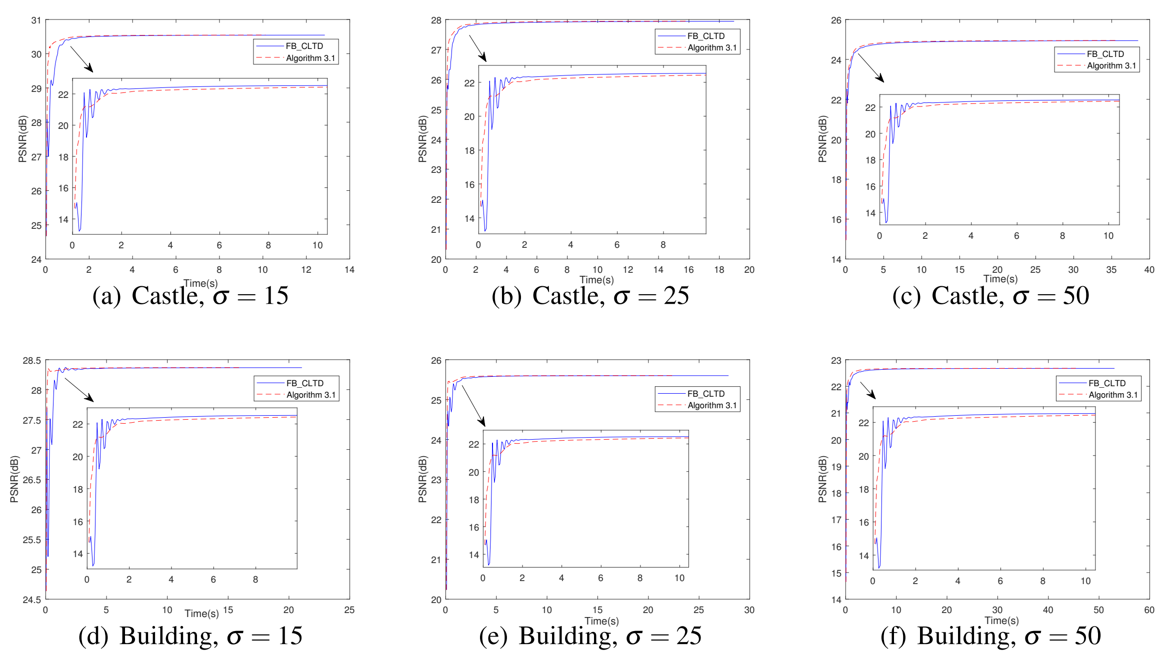
We plot the PSNR performance with CPU time performance of FB_CLTD and Algorithm 3.1 in Figure 2 and Figure 3, respectively. It can be observed that the SSIM of both algorithms converge with almost equal values, respectively. Especially the zoomed-in images, Algorithm 3.1 is more stable than FB_CLTD in the early iteration, i.e., the convergence of Algorithm 3.1 is more robust than FB_CLTD. Furthermore, we present the restored images in Figure 4 and Figure 5. As shown in Table 5, the PSNR values of the images restored by Algorithm 3.1 and FB_CLTD demonstrate minimal variation. Consequently, there is no obvious visual difference between the images restored by the two algorithms.



(a) Castle, $\sigma = 15$ (b) Castle, $\sigma = 25$ (c) Castle, $\sigma = 50$

(d) Building, $\sigma = 15$ (e) Building, $\sigma = 25$ (f) Building, $\sigma = 50$

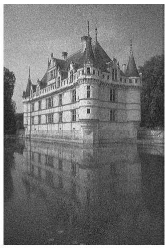
FIGURE 2. For $\ell_2 - \text{IC}$ (4.1), PSNR vs CPU time.



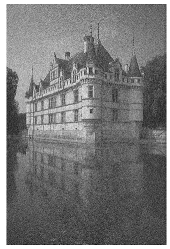
(a) Castle, $\sigma = 15$ (b) Castle, $\sigma = 25$ (c) Castle, $\sigma = 50$

(d) Building, $\sigma = 15$ (e) Building, $\sigma = 25$ (f) Building, $\sigma = 50$

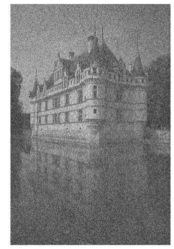
FIGURE 3. For $\ell_2 - \text{MIC}$ (4.2), PSNR vs CPU time.



(a) $\sigma = 15$

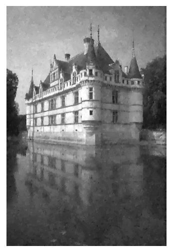


(b) $\sigma = 25$

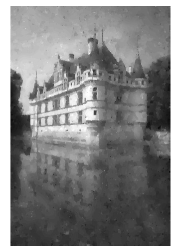


(c) $\sigma = 50$

(d) $\ell_2 -$ IC/FB_CLTD

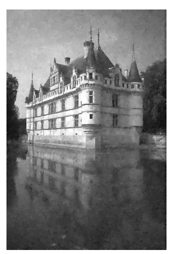


(e) $\ell_2 -$ IC/FB_CLTD

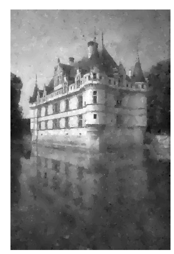


(f) $\ell_2 -$ IC/FB_CLTD

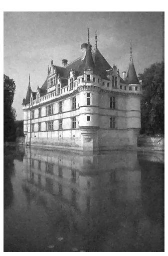
(g) $\ell_2 -$ IC/Algorithm 3.1



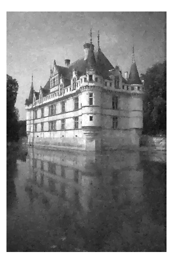
(h) $\ell_2 -$ IC/Algorithm 3.1



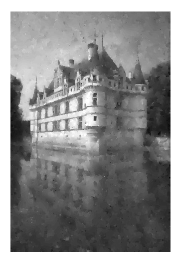
(i) $\ell_2 -$ IC/Algorithm 3.1



(j) $\ell_2 -$ MIC/FB_CLTD

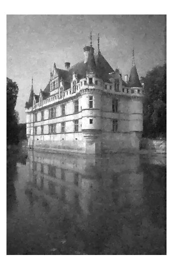


(k) $\ell_2 -$ MIC/FB_CLTD

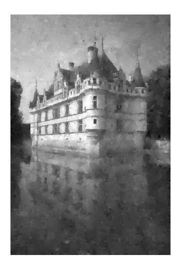


(l) $\ell_2 -$ MIC/FB_CLTD

(m) $\ell_2 -$ MIC/Algorithm 3.1

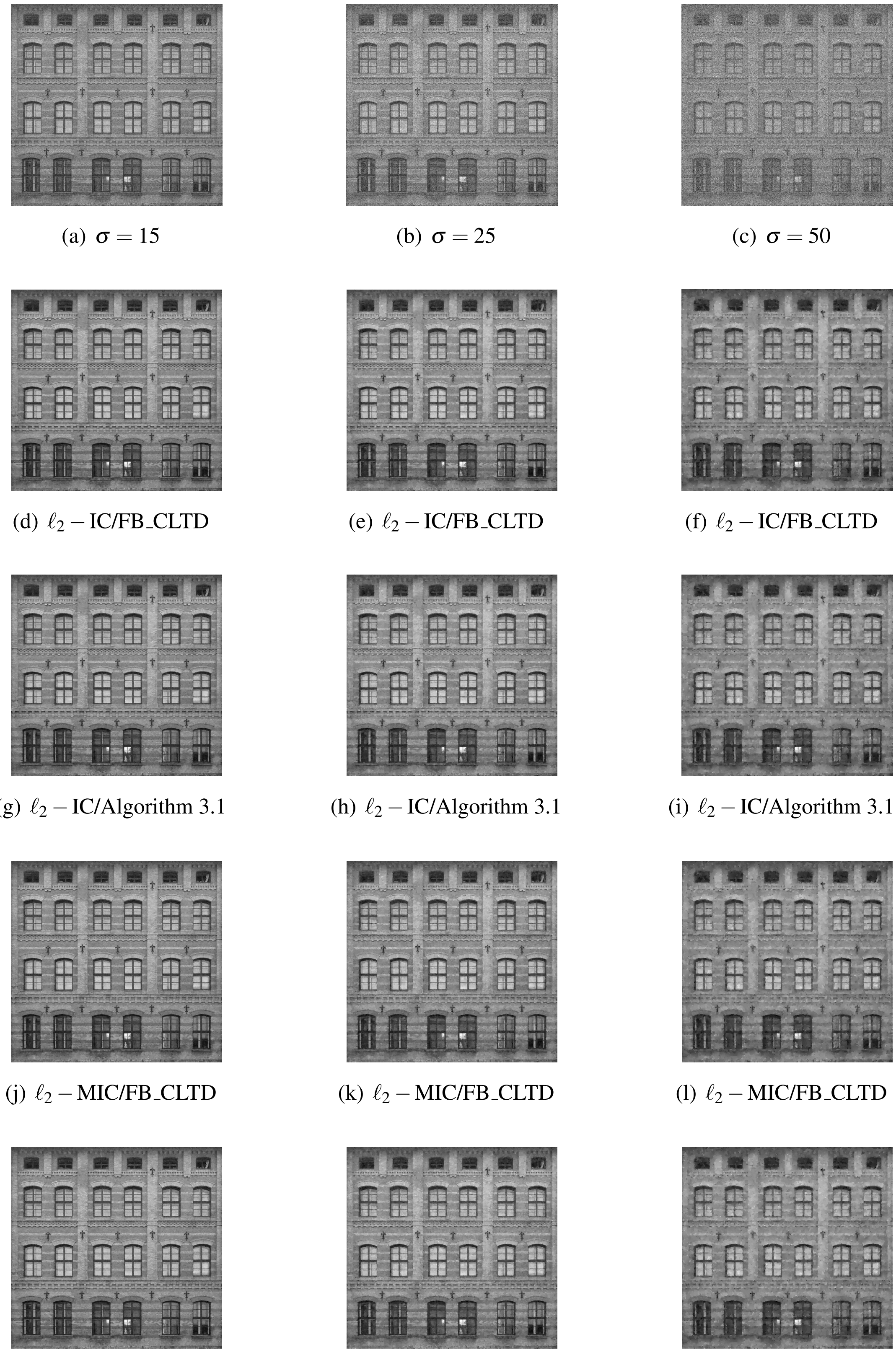


(n) $\ell_2 -$ MIC/Algorithm 3.1



(o) $\ell_2 -$ MIC/Algorithm 3.1

FIGURE 4. Noisy and restored "Castle" images.



(a) $\sigma = 15$ (b) $\sigma = 25$ (c) $\sigma = 50$

(d) $\ell_2 -$ IC/FB_CLTD (e) $\ell_2 -$ IC/FB_CLTD (f) $\ell_2 -$ IC/FB_CLTD

(g) $\ell_2 -$ IC/Algorithm 3.1 (h) $\ell_2 -$ IC/Algorithm 3.1 (i) $\ell_2 -$ IC/Algorithm 3.1

(j) $\ell_2 -$ MIC/FB_CLTD (k) $\ell_2 -$ MIC/FB_CLTD (l) $\ell_2 -$ MIC/FB_CLTD

(m) $\ell_2 -$ MIC/Algorithm 3.1 (n) $\ell_2 -$ MIC/Algorithm 3.1 (o) $\ell_2 -$ MIC/Algorithm 3.1

FIGURE 5. Noisy and restored "Building" images.

5. Conclusions

In this paper, we developed a new primal-dual splitting algorithm to solve monotone inclusion problem (1.1)-(1.2), which is strongly related to convex minimization problem (3.5)-(3.6). Firstly, we transformed the monotone inclusion into the sum of two maximally monotone operators under a proper product space. Based on the preconditioned forward-backward splitting algorithm, we proved the convergence of the proposed algorithm with appropriate parameter conditions. The proposed algorithm has a simpler form than (3.4). Additionally, we employed the proposed algorithm to solve a class of convex minimization problems. To verify the advantages of the proposed algorithm, we used it to solve image denoising models (4.1) and (4.2). The numerical results showed that the proposed algorithm demonstrates a reduction in the number of iterations and the CPU time when solving (4.2) compared to (3.4).

Acknowledgements

This work was supported by the National Natural Science Foundations of China (12271117, 12031003, 12061045, 12401559), the Guangzhou Education Scientific Research Project 2024 (202315829), the Guangzhou University Research Projects (RC2023061), and the Jiangxi Provincial Natural Science Foundation (20224ACB211004). We would like to express our thanks to the reviewers and editors for their comments and suggestions provided during the review process, which significantly improved the quality of this paper.

References

[1] George H.G. Chen and R.T. Rockafellar, Convergence rates in forward-backward splitting, SIAM J. Optim. 7 (1997), 421-444.

[2] P.L. Combettes and V.R. Wajs, Signal recovery by proximal forward-backward splitting, Multiscale Model. Simul. 4 (2005), 1168-1200.

[3] J. Eckstein and D. Bertsekas, On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators, Math. Program. 55 (1992), 293-318.

[4] P.L. Combettes and J.-C. Pesquet, A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery, IEEE J. Sel. Top. Signal Process 1 (2007), 564-574.

[5] P. Tseng, A modified forward-backward splitting method for maximal monotone mappings, SIAM J. Control Optim. 38 (2000), 431-446.

[6] B.C. Vũ, A variable metric extension of the forward-backward-forward algorithm for monotone operators, Numer. Funct. Anal. Optim. 34 (2013), 1050-1065.

[7] Y. Malitsky and M.K. Tam, A forward-backward splitting method for monotone inclusions without cocoercivity, SIAM J. Optim. 30 (2020), 1451-1472.

[8] V. Cevher and B.C. Vũ, A reflected forward-backward splitting method for monotone inclusions involving lipschitzian operators, Set-valued Vari. Anal. 29 (2021), 163-174.

[9] D. Kitkuan, P. Kumam, and J. Martínez-Moreno, Generalized halpern-type forward–backward splitting methods for convex minimization problems with application to image restoration problems, Optimization 69 (2020), 1557-1581.

[10] C. Izuchukwu, S. Reich, Y. Shehu, and A. Taiwo, Strong convergence of forward-reflected-backward splitting methods for solving monotone inclusions with applications to image restoration and optimal control, J. Sci. Comput. 94 (2023) 73.

[11] B. Tan and X. Qin, On relaxed inertial projection and contraction algorithms for solving monotone inclusion problems, Adv. Comput. Math. 50 (2024), 59.

[12] P. Latafat and P. Patrinos, Asymmetric forward–backward–adjoint splitting for solving monotone inclusions involving three operators, Comput. Optim. Appl. 68 (2017), 57-93.

[13] F.J. Aragõn-Artacho, Y. Malitsky, M.K. Tam, and D. Torregrosa-Belẽn, Distributed forward-backward methods for ring networks, Comput. Optim. Appl. 86 (2023), 845-870.
[14] Y. Cao, Y.H. Wang, H. ur Rehman, Y. Shehu, and J. Yao, Convergence analysis of a new forward-reflected-backward algorithm for four operators without cocoercivity, J. Optim. Theory Appl. 203 (2024), 256–284.
[15] P. J. Huber, Robust regression: asymptotics, conjectures and monte carlo, Ann. Statist. 1 (1973), 799-821.
[16] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, An iterative regularization method for total variation based image restoration, SIAM Multiscale Model. Sim. 4 (2005), 460-489.
[17] A. Chambolle and P.L. Lions, Image recovery via total variation minimizaing and related problems, Numer. Math. 76 (1997), 167-188.
[18] S. Setzer, G. Steidl, and T. Teuber, Infimal convolution regularization with discrete l1-type functionals, Commun. Math. Sci. 9 (2011), 797-827.
[19] K. Bredies, K. Kunisch, and T. Pock, Total generalized variation, SIAM J. Imaging Sci. 3 (2010), 492-526.
[20] R.W. Liu, L. Shi, S. C.H. Yu, and D. Wang, Box-constrained second-order total generalized variation minimization with a combined $l^{1,2}$ data-fidelity for image reconstruction, J. Electron. Imaging, 24 (2015), 033026.
[21] D. Davis and W.T. Yin, A three-operator splitting scheme and its optimization applications, Set-valued Var. Anal. 25 (2017), 829-858.
[22] C.X. Zong, Y.C. Tang, and G.F. Zhang, Solving monotone inclusions involving the sum of three maximally monotone operators and a cocoercive operator with applications, Set-valued Var. Anal. 31 (2023), 16.
[23] E.K. Ryu, Uniqueness of drs as the 2 operator resolvent-splitting and impossibility of 3 operator resolvent-splitting, Math. Program. 182 (2020), 233-273.
[24] B.C. Vũ, A splitting algorithm for dual monotone inclusions involving cocoercive operators, Adv. Comput. Math. 38 (2013), 667-681.
[25] L. Condat, A primal-dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms, J. Optim. Theory Appl. 158 (2013), 460-479.
[26] R.I. Bot, E. R. Csetnek, and C. Hendrich, Recent developments on primal-dual splitting methods with applications to convex minimization, Springer, New York, 2014.
[27] S.R. Becker and P.L. Combettes, An algorithm for splitting parallel sums of linearly composed monotone operatos with applications to signal recovery, J.Nonlinear Convex Anal. 15 (2014), 137-159.
[28] P.L. Combettes, Systems of structured monotone inclusions: Duality, algorithms, and applications, SIAM J. Optim. 23 (2013), 2420-2447.
[29] R.I. Boţ and C. Hendrich, A douglas-rachford type primal-dual method for solving inclusions with mixtures of composite and parallel -sum type monotone operators, SIAM J. Optim. 4 (2013), 2541-2565.
[30] Y.X. Yang, Y.C. Tang, M. Wen, and T.Y. Zeng, Preconditioned douglas-rachford type primal-dual method for solving composite monotone inclusion problems with applications, Inverse Probl. Imaging 15 (2021), 787-825.
[31] Y.C. Tang, M. Wen, and T.Y. Zeng, Preconditioned three-operator splitting algorithm with applications to image restoration, J. Sci. Comput. 92 (2022), 106.
[32] V.C. Bang, D. Papadimitriou, V.X. Nham, A primal-dual backward reflected forward splitting algorithm for structured monotone inclusions, Acta Math. Vietnam 49 (2024), 159-172.
[33] R.I. Boţ and C. Hendrich, Solving monotone inclusions involving parallel sums of linearly composed maximally monotone operators, Inverse Probl. Imaging 10 (2016), 617-640.
[34] J.J. Chen, X.Y. Luo, Y.C. Tang, and Q.L. Dong, Primal-dual splitting algorithms for solving structured monotone inclusion with applications, Symmetry, 13 (2021), 2415.
[35] L.M. Briceño-Arias and D. Davis, Forward-backward-half forward algorithm for solving monotone inclusions, SIAM J. Optim. 28 (2018), 2839-2871.
[36] H.H. Bauschke and P.L. Combettes, Convex Analysis and Monotone Operator Theory in Hilbert Spaces, second edition, Springer, London, 2017.
[37] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Trans. Image Process. 13 (2004), 600-612.