symmetry

Article

Primal-Dual Splitting Algorithms for Solving Structured Monotone Inclusion with Applications

Jinjian Chen [1,†], Xingyu Luo [1,†], Yuchao Tang [1,*] and Qiaoli Dong [2]

[1] Department of Mathematics, Nanchang University, Nanchang 330031, China; 400603319001@email.ncu.edu.cn (J.C.); 5503118023@email.ncu.edu.cn (X.L.)
[2] Tianjin Key Laboratory for Advanced Signal Processing and College of Science, Civil Aviation University of China, Tianjin 300300, China; dongql@lsec.cc.ac.cn
* Correspondence: yctang@ncu.edu.cn
† Jinjian Chen and Xingyu Luo are the co-first authors of this paper.

Abstract: This work proposes two different primal-dual splitting algorithms for solving structured monotone inclusion containing a cocoercive operator and the parallel-sum of maximally monotone operators. In particular, the parallel-sum is symmetry. The proposed primal-dual splitting algorithms are derived from two approaches: One is the preconditioned forward–backward splitting algorithm, and the other is the forward–backward–half-forward splitting algorithm. Both algorithms have a simple calculation framework. In particular, the single-valued operators are processed via explicit steps, while the set-valued operators are computed by their resolvents. Numerical experiments on constrained image denoising problems are presented to show the performance of the proposed algorithms.

Keywords: primal-dual algorithm; monotone inclusion; cocoercive operator; infimal convolution

Citation: Chen, J.; Luo, X.; Tang, Y.; Dong, Q. Primal-Dual Splitting Algorithms for Solving Structured Monotone Inclusion with Applications. *Symmetry* **2021**, *13*, 2415. https://doi.org/10.3390/sym13122415

Academic Editor: Mihai Postolache

Received: 15 November 2021
Accepted: 8 December 2021
Published: 13 December 2021

1. Introduction

In the last decade, there has been great interest in primal-dual splitting algorithms for solving structured monotone inclusion. The reason is that many convex minimization problems arising in image processing, statistical learning, and economic management can be modelled by such monotone inclusion problems. Based on the perspective of operator splitting algorithms, these primal-dual splitting algorithms can be roughly divided into four categories: (i) Forward–backward splitting type [1–4]; (ii) Douglas-Rachford splitting type [5–7]; (iii) Forward–backward–forward splitting type [8–12]; and (iv) Projective splitting type [13–17]. In 2014, Becker and Combettes [11] first studied the following structured monotone inclusion problem:

Problem 1. *Let $\mathcal{H}$ be a real Hilbert space, $z \in \mathcal{H}$, and $m > 0$ be an integer. Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be maximally monotone and $C : \mathcal{H} \to \mathcal{H}$ be monotone and L-Lipschitzian continuous, for some $L > 0$. For every integer $i = 1, \ldots, m$, let $\mathcal{X}_i$ and $\mathcal{G}_i$ be real Hilbert spaces, let $B_i : \mathcal{X}_i \to 2^{\mathcal{X}_i}$ and $D_i : \mathcal{Y}_i \to 2^{\mathcal{Y}_i}$ be maximally monotone operators, and let $K_i : \mathcal{H} \to \mathcal{X}_i$ and $M_i : \mathcal{H} \to \mathcal{Y}_i$ be nonzero linear bounded operators. The problem is to solve the primal inclusion*

$$\text{find } x \in \mathcal{H} \quad \text{such that } z \in Ax + \sum_{i=1}^{m}((K_i^* \circ B_i \circ K_i)\Box(M_i^* \circ D_i \circ M_i))x + Cx, \tag{1}$$

together with its dual inclusion

$$\text{find} \quad \begin{cases} p_i \in \mathcal{X}_i, i = 1, \ldots, m, \\ q_i \in \mathcal{Y}_i, i = 1, \ldots, m, \\ y_i \in \mathcal{H}, i = 1, \ldots, m, \end{cases} \quad \text{such that} \quad \exists x \in \mathcal{H} : \begin{cases} z - \sum_{i=1}^{m} L_i^* K_i^* p_i \in Ax + Cx, \\ K_i x - K_i y_i \in B_i^{-1} p_i, i = 1, \ldots, m, \\ M_i y_i \in D_i^{-1} q_i, i = 1, \ldots, m, \\ K_i^* p_i = M_i^* q_i, i = 1, \ldots, m. \end{cases} \tag{2}$$

Based on the method of [10], they proposed a primal-dual splitting algorithm to solve (1) and (2). Moreover, they applied the algorithm to solve an image restoration model with infimal convolution terms mixing first- and second-order total variation, which was initially studied in [18] and further explored in [19]. The advantage of this model is that it can reduce the staircase effects caused by the first-order total variation. Furthermore, Boţ and Hendrich [4] studied a more general monotone inclusion problem as follows.

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

$$\text{find } x \in \mathcal{H} \quad \text{such that } z \in Ax + \sum_{i=1}^{m} L_i^* ((K_i^* \circ B_i \circ K_i) \square (M_i^* \circ D_i \circ M_i))(L_i x - r_i) + Cx, \tag{3}$$

together with its dual inclusion

$$\text{find} \quad \begin{cases} p_i \in \mathcal{X}_i, i = 1, \ldots, m, \\ q_i \in \mathcal{Y}_i, i = 1, \ldots, m, \\ y_i \in \mathcal{G}_i, i = 1, \ldots, m, \end{cases} \quad \text{such that} \quad \exists x \in \mathcal{H} : \begin{cases} z - \sum_{i=1}^{m} L_i^* K_i^* p_i \in Ax + Cx, \\ K_i(L_i x - y_i - r_i) \in B_i^{-1} p_i, i = 1, \ldots, m, \\ M_i y_i \in D_i^{-1} q_i, i = 1, \ldots, m, \\ K_i^* p_i = M_i^* q_i, i = 1, \ldots, m. \end{cases} \tag{4}$$

It is easy to see that Problem 1 could be viewed as a special case of Problem 2 by letting $L_i = I$ and $r_i = 0$, for any $i = 1, \cdots, m$. They proposed two different primal-dual algorithms for solving the primal-dual pair of monotone inclusions (3) and (4). The first algorithm is a forward–backward splitting type, which is defined as follows. Let $x_0 \in \mathcal{H}$, and for any $i = 1, \ldots, m$, $p_{i,0} \in \mathcal{X}_i$, $q_{i,0} \in \mathcal{Y}_i$ and $z_{i,0}, y_{i,0}, v_{i,0} \in \mathcal{G}_i$, and set

$$(\forall n \geq 0) \left\lfloor \begin{array}{l} \widetilde{x}_n = J_{\tau A}(x_n - \tau(Cx_n + \sum_{i=1}^{m} L_i^* v_{i,n} - z)) \\ \text{for } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} \widetilde{p}_{i,n} = J_{\theta_{1,i} B_i^{-1}}(p_{i,n} + \theta_{1,i} K_i z_{i,n}) \\ \widetilde{q}_{i,n} = J_{\theta_{2,i} D_i^{-1}}(q_{i,n} + \theta_{2,i} M_i y_{i,n}) \\ u_{1,i,n} = z_{i,n} + \gamma_{1,i}(K_i^*(p_{i,n} - 2\widetilde{p}_{i,n}) + v_{i,n} + \sigma_i(L_i(2\widetilde{x}_n - x_n) - r_i)) \\ u_{2,i,n} = y_{i,n} + \gamma_{2,i}(M_i^*(q_{i,n} - 2\widetilde{q}_{i,n}) + v_{i,n} + \sigma_i(L_i(2\widetilde{x}_n - x_n) - r_i)) \\ \widetilde{z}_{i,n} = \dfrac{1 + \sigma_i \gamma_{2,i}}{1 + \sigma_i(\gamma_{1,i} + \gamma_{2,i})}(u_{1,i,n} - \dfrac{\sigma_i \gamma_{1,i}}{1 + \sigma_i \gamma_{2,i}} u_{2,i,n}) \\ \widetilde{y}_{i,n} = \dfrac{1}{1 + \sigma_i \gamma_{2,i}}(u_{2,i,n} - \sigma_i \gamma_{2,i} \widetilde{z}_{i,n}) \\ \widetilde{v}_{i,n} = v_{i,n} + \sigma_i(L_i(2\widetilde{x}_n - x_n) - r_i - \widetilde{z}_{i,n} - \widetilde{y}_{i,n}) \end{array} \right. \\ x_{n+1} = x_n + \lambda_n(\widetilde{x}_n - x_n) \\ \text{for } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} p_{i,n+1} = p_{i,n} + \lambda_n(\widetilde{p}_{i,n} - p_{i,n}) \\ q_{i,n+1} = q_{i,n} + \lambda_n(\widetilde{q}_{i,n} - q_{i,n}) \\ z_{i,n+1} = z_{i,n} + \lambda_n(\widetilde{z}_{i,n} - z_{i,n}) \\ y_{i,n+1} = y_{i,n} + \lambda_n(\widetilde{y}_{i,n} - y_{i,n}) \\ v_{i,n+1} = v_{i,n} + \lambda_n(\widetilde{v}_{i,n} - v_{i,n}), \end{array} \right. \end{array} \right. \tag{5}$$

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 such that

$$2\mu^{-1}(1-\bar{\alpha}) \min_{i=1,\ldots,m} \left\{ \frac{1}{\tau}, \frac{1}{\theta_{1,i}}, \frac{1}{\theta_{2,i}}, \frac{1}{\gamma_{1,i}}, \frac{1}{\gamma_{2,i}}, \frac{1}{\sigma_i} \right\} > 1 \tag{6}$$

for

$$\bar{\alpha} = \max\left\{ \sqrt{\tau \sum_{i=1}^{m} \sigma_i \|L_i\|^2}, \max_{j=1,\ldots,m} \left\{ \sqrt{\theta_{1,j}\gamma_{1,j}\|K_j\|^2}, \sqrt{\theta_{2,j}\gamma_{2,j}\|M_j\|^2} \right\} \right\}. \tag{7}$$

In addition, $\{\lambda_n\} \subseteq [\varepsilon, 1]$, where $\varepsilon \in (0,1)$. The second algorithm is a forward–backward–forward splitting type. Let $x_0 \in \mathcal{H}$, and for any $i = 1, \ldots, m$, let $p_{i,0} \in \mathcal{X}_i$, $q_{i,0} \in \mathcal{Y}_i$, $z_{i,0}, y_{i,0}, v_{i,0} \in \mathcal{G}_i$, and set

$$(\forall n \geq 0) \left\lfloor \begin{array}{l} \tilde{x}_n = J_{\gamma_n A}(x_n - \gamma_n(Cx_n + \sum_{i=1}^{m} L_i^* v_{i,n} - z)) \\ \text{for } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} \widetilde{p}_{i,n} = J_{\gamma_n B_i^{-1}}(p_{i,n} + \gamma_n K_i z_{i,n}) \\ \widetilde{q}_{i,n} = J_{\gamma_n D_i^{-1}}(q_{i,n} + \gamma_n M_i y_{i,n}) \\ u_{1,i,n} = z_{i,n} - \gamma_n(K_i^* p_{i,n} - v_{i,n} - \gamma_n(L_i x_n - r_i)) \\ u_{2,i,n} = y_{i,n} - \gamma_n(M_i^* q_{i,n} - v_{i,n} - \gamma_n(L_i x_n - r_i)) \\ \widetilde{z}_{i,n} = \dfrac{1+\gamma_n^2}{1+2\gamma_n^2}(u_{1,i,n} - \dfrac{\gamma_n^2}{1+\gamma_n^2} u_{2,i,n}) \\ \widetilde{y}_{i,n} = \dfrac{1}{1+\gamma_n^2}\left(u_{2,i,n} - \gamma_n^2 \widetilde{z}_{i,n}\right) \\ \widetilde{v}_{i,n} = v_{i,n} + \gamma_n(L_i x_n - r_i - \widetilde{z}_{i,n} - \widetilde{y}_{i,n}) \end{array} \right. \\ x_{n+1} = \widetilde{x}_n + \gamma_n(Cx_n - C\tilde{x}_n + \sum_{i=1}^{m} L_i^*(v_{i,n} - \widetilde{v}_{i,n})) \\ \text{for } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} p_{i,n+1} = \widetilde{p}_{i,n} - \gamma_n(K_i(z_{i,n} - \widetilde{z}_{i,n})) \\ q_{i,n+1} = \widetilde{q}_{i,n} - \gamma_n(M_i(y_{i,n} - \widetilde{y}_{i,n})) \\ z_{i,n+1} = \widetilde{z}_{i,n} + \gamma_n(K_i^*(p_{i,n} - \widetilde{p}_{i,n})) \\ y_{i,n+1} = \widetilde{y}_{i,n} + \gamma_n(M_i^*(q_{i,n} - \widetilde{q}_{i,n})) \\ v_{i,n+1} = \widetilde{v}_{i,n} - \gamma_n(L_i(x_n - \widetilde{x}_n)), \end{array} \right. \end{array} \right. \tag{8}$$

where $\{\gamma_n\} \subseteq \left[\varepsilon, \frac{1-\varepsilon}{\beta}\right]$ with $\varepsilon \in \left(0, \frac{1}{\beta+1}\right)$ and

$$\beta = \mu + \sqrt{\max\left\{ \sum_{i=1}^{m} \|L_i\|^2, \max_{j=1,\ldots,m} \left\{ \|K_j\|^2, \|M_j\|^2 \right\} \right\}}. \tag{9}$$

The first algorithm (5) could be viewed as a preconditioned forward–backward splitting algorithm [20]. While the second algorithm (8) is an instance of the forward-backward-forward splitting algorithm proposed by Tseng [21]. We can see that the operators B_i^{-1} and D_i^{-1} are symmetry in both of algorithms. We call the first algorithm (5) the FB_BH algorithm and the second algorithm, the FBF_BH algorithm.

In this paper, we continue studying primal-dual splitting algorithms for solving the structured monotone inclusion (3) and (4). First, we establish a new convergence theorem for the primal-dual forward–backward splitting type algorithm (5). We relax the limitation of the iteration parameters as well as expand the selection range of the relaxation parameter. Since the primal-dual forward–backward–forward splitting type algorithm (8) does not make full use of the cocoercive operator, we introduce a new primal-dual splitting algorithm for solving (3) and (4), which is based on the forward–backward–half-forward splitting algorithm [22]. This new algorithm is not only less computationally expensive than the original algorithm but also provides a larger range of parameter selection. To show the advantages of the proposed algorithms, we apply them to solve image denoising problems.

This paper is organized as follows. In Section 2, we recall some preliminary results in monotone operator theory. In Section 3, we present the main results. We study the convergence of two primal-dual splitting algorithms for solving (3) and (4). Furthermore, we employ the obtained algorithms to solve convex minimization problems. In Section 4, we perform numerical experiments on image denoising problems. Finally, we present conclusions.

2. Preliminaries

Throughout this paper, $\mathcal{H}$ is a real Hilbert space, which is equipped with inner product $\langle \cdot, \cdot \rangle$ and associated norm $\| \cdot \| = \sqrt{\langle \cdot, \cdot \rangle}$. Let $2^{\mathcal{H}}$ be the power set of $\mathcal{H}$. Let $L : \mathcal{H} \to \mathcal{G}$ be a linear bounded operator, where $\mathcal{G}$ is another real Hilbert space, the operator $L^* : \mathcal{G} \to \mathcal{H}$ is said to be its adjoint if $\langle Lx, y \rangle = \langle x, L^* y \rangle$ holds for all $x \in \mathcal{H}$ and all $y \in \mathcal{G}$. Most of the definitions are taken from [23].

Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be a set-valued operator. Let zer $A = \{x \in \mathcal{H} : 0 \in Ax\}$ be the set of its zeros, ran $A = \{u \in \mathcal{H} : \exists x \in \mathcal{H}, u \in Ax\}$ its range, and gra $A = \{(x, u) \in \mathcal{H} \times \mathcal{H} : u \in Ax\}$ its graph. The inverse of A is defined by $A^{-1} : \mathcal{H} \to 2^{\mathcal{H}}, u \mapsto \{x \in \mathcal{H} : u \in Ax\}$.

Definition 1. *Let $A : \mathcal{H} \to 2^{\mathcal{H}}$ be a set-valued operator. A is said to be monotone, if $\langle x - y, u - v \rangle \geq 0, \quad \forall (x, u), (y, v) \in \operatorname{gra} A$. Furthermore, A is said to be maximally monotone if it is monotone, and there exists no monotone operator $B : \mathcal{H} \to 2^{\mathcal{H}}$ such that gra B properly contains gra A.*

Definition 2. *Let $T : \mathcal{H} \to \mathcal{H}$ be a single-valued operator.*

(i) T is said to be L-Lipschitzian, for some $L > 0$, if $\|Tx - Ty\| \leq L\|x - y\|, \quad \forall x, y \in \mathcal{H}$.

(ii) T is said to be μ-cocoercive, for some $\mu > 0$, if $\langle x - y, Tx - Ty \rangle \geq \mu \|Tx - Ty\|^2$, $\forall x, y \in \mathcal{H}$.

Definition 3. *Let $A : \mathcal{H} \to 2^{\mathcal{H}}$, the resolvent of A with index $\lambda > 0$ is defined by*

$$J_{\lambda A} = (Id + \lambda A)^{-1}, \tag{10}$$

where Id denotes the identity operator on $\mathcal{H}$.

Definition 4. *Let $A_1, A_2 : \mathcal{H} \to 2^{\mathcal{H}}$ be two set-valued operators. The sum and the parallel sum of A_1 and A_2 are defined as follows:*

$$\begin{aligned} &A_1 + A_2 : \mathcal{H} \to 2^{\mathcal{H}}, (A_1 + A_2)(x) = A_1(x) + A_2(x) \\ &A_1 \square A_2 : \mathcal{H} \to 2^{\mathcal{H}}, A_1 \square A_2 = \left(A_1^{-1} + A_2^{-1} \right)^{-1}. \end{aligned} \tag{11}$$

Let $f : \mathcal{H} \to \overline{\mathbb{R}} := \mathbb{R} \cup \{\pm\infty\}$. We denote its effective domain as dom $f := \{x \in \mathcal{H} : f(x) < +\infty\}$, f is said to be proper if dom $f \neq \varnothing$ and $f(x) > -\infty$ for all $x \in \mathcal{H}$. Furthermore, we define $\Gamma_0(\mathcal{H}) := \{f : \mathcal{H} \to \overline{\mathbb{R}} \mid f$ is proper, convex, and lower semicontinuous (lsc) $\}$.

The conjugate function of f is defined by $f^* : \mathcal{H} \to \overline{\mathbb{R}}, f^*(p) = \sup\{\langle p, x \rangle - f(x) : x \in \mathcal{H}\}$ for all $p \in \mathcal{H}$. If $f \in \Gamma_0(\mathcal{H})$, then $f^{**} = f$.

Let $f \in \Gamma_0(\mathcal{H})$, the subdifferential of f is defined by $\partial f : \mathcal{H} \to 2^{\mathcal{H}} : x \mapsto \{u \in \mathcal{H} \mid (\forall y \in \mathcal{H}) f(y) \geq f(x) + \langle u \mid y - x \rangle\}$. If $f \in \Gamma_0(\mathcal{H})$, then ∂f is maximally monotone.

Let two proper functions $f, h : \mathcal{H} \to \overline{\mathbb{R}}$,

$$f \square h : \mathcal{H} \to \overline{\mathbb{R}}, (f \square h)(x) = \inf_{y \in \mathcal{H}} \{f(y) + h(x - y)\}, \tag{12}$$

denotes their infimal convolution.

Let $f \in \Gamma_0(\mathcal{H})$ and $\gamma > 0$, the proximity operator of f is defined by

$$\mathrm{prox}_{\gamma f} : \mathcal{H} \to \mathcal{H} : x \mapsto \underset{y \in \mathcal{H}}{\mathrm{argmin}} \Big\{ f(y) + \frac{1}{2\gamma}\|x - y\|^2 \Big\}. \tag{13}$$

It follows from $f \in \Gamma_0(\mathcal{H})$ that $\mathrm{prox}_{\gamma f}(x) = J_{\gamma \partial f}(x)$.

The following lemma shows the relationship between the proximity operator of f and its convex conjugate f^*.

Lemma 1. *(Moreau's decomposition) Let $\gamma > 0$ and $f \in \Gamma_0(\mathcal{H})$, then*

$$\mathrm{prox}_{\gamma f}(x) + \gamma\, \mathrm{prox}_{\frac{1}{\gamma} f^*}(\frac{1}{\gamma} x) = x, \ \forall x \in \mathcal{H}.$$

Let $C \subseteq \mathcal{H}$ be a nonempty closed convex set, the indicator function of C is denoted by

$$\delta_C : x \mapsto \begin{cases} 0, \text{ if } x \in C \\ +\infty, \text{ otherwise.} \end{cases} \tag{14}$$

The orthogonal projection onto C is defined by P_C, which is $P_C(x) = \underset{y \in C}{\mathrm{argmin}}\|x - y\|$. Let $\gamma > 0$, $\mathrm{prox}_{\gamma \delta_C}(x) = P_C(x)$.

3. Main Results

In this section, we study primal-dual splitting algorithms for solving (3) and (4) and discuss their asymptotic convergence. First, we provide a technique lemma, which shows that the primal-dual monotone inclusions (3) and (4) are equivalent to the sum of three maximally monotone operators. In the following, let $\boldsymbol{\mathcal{H}} = \mathcal{H}_1 \oplus \cdots \mathcal{H}_m$ be the direct sum of real Hilbert spaces $\{\mathcal{H}_i\}_{i=1}^m$. Let $\boldsymbol{v} = (v_1, \cdots, v_m) \in \boldsymbol{\mathcal{H}}$ and $\boldsymbol{q} = (q_1, \cdots, q_m) \in \boldsymbol{\mathcal{H}}$, the inner product and associated norm on $\boldsymbol{\mathcal{H}}$ are defined as

$$\langle \boldsymbol{v}, \boldsymbol{q} \rangle_{\boldsymbol{\mathcal{H}}} = \sum_{i=1}^m \langle v_i, q_i \rangle, \quad \|\boldsymbol{v}\|_{\boldsymbol{\mathcal{H}}} = \sqrt{\sum_{i=1}^m \|v_i\|^2}.$$

Lemma 2. *Let $\mathcal{H}$, A, C, $\mathcal{X}_i$, $\mathcal{Y}_i$, $\mathcal{G}_i$, B_i, D_i, L_i, K_i, M_i, $i = 1, \cdots, m$ be defined as in Problem 2, and let*

$$\begin{aligned}
&\boldsymbol{\mathcal{X}} := \mathcal{X}_1 \oplus \cdots \oplus \mathcal{X}_m, \boldsymbol{\mathcal{Y}} := \mathcal{Y}_1 \oplus \cdots \oplus \mathcal{Y}_m, \boldsymbol{\mathcal{G}} := \mathcal{G}_1 \oplus \cdots \oplus \mathcal{G}_m, \boldsymbol{\mathcal{K}} := \mathcal{H} \oplus \boldsymbol{\mathcal{X}} \oplus \boldsymbol{\mathcal{Y}} \oplus \boldsymbol{\mathcal{G}} \oplus \boldsymbol{\mathcal{G}} \oplus \boldsymbol{\mathcal{G}}, \\
&\boldsymbol{p} = (p_1, \ldots, p_m) \in \boldsymbol{\mathcal{X}}, \boldsymbol{q} = (q_1, \ldots, q_m) \in \boldsymbol{\mathcal{Y}}, \boldsymbol{z} = (z_1, \ldots, z_m) \in \boldsymbol{\mathcal{G}}, \\
&\boldsymbol{y} = (y_1, \ldots, y_m) \in \boldsymbol{\mathcal{G}}, \boldsymbol{v} = (v_1, \ldots, v_m) \in \boldsymbol{\mathcal{G}}, \boldsymbol{r} = (r_1, \ldots, r_m) \in \boldsymbol{\mathcal{G}}, \\
&\boldsymbol{B} : \boldsymbol{\mathcal{X}} \to 2^{\boldsymbol{\mathcal{X}}} : \boldsymbol{p} \mapsto (B_1 p_1, \cdots, B_m p_m), \boldsymbol{D} : \boldsymbol{\mathcal{Y}} \to 2^{\boldsymbol{\mathcal{Y}}}, \boldsymbol{q} \mapsto (D_1 q_1, \cdots, D_m q_m), \\
&\widetilde{M} : \boldsymbol{\mathcal{G}} \to \boldsymbol{\mathcal{Y}}, \boldsymbol{y} \mapsto (M_1 y_1, \cdots, M_m y_m), \widetilde{K} : \boldsymbol{\mathcal{G}} \to \boldsymbol{\mathcal{X}}, \boldsymbol{y} \mapsto (K_1 y_1, \cdots, K_m y_m), \\
&\boldsymbol{M} : \mathcal{K} \to 2^{\mathcal{K}}, (x, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{v}) \mapsto (-z + Ax) \times \boldsymbol{B}^{-1}\boldsymbol{p} \times \boldsymbol{D}^{-1}\boldsymbol{q} \times (-\boldsymbol{v}, -\boldsymbol{v}, \boldsymbol{r} + \boldsymbol{z} + \boldsymbol{y}) \\
&\boldsymbol{S} : \mathcal{K} \to \mathcal{K}, (x, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{v}) \mapsto \Big(\sum_{i=1}^m L_i^* v_i, -\widetilde{K}\boldsymbol{z}, -\widetilde{M}\boldsymbol{y}, \widetilde{K}^*\boldsymbol{p}, \widetilde{M}^*\boldsymbol{q}, -L_1 x, \ldots, -L_m x \Big) \\
&\boldsymbol{Q} : \mathcal{K} \to \mathcal{K}, (x, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{v}) \mapsto (Cx, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0}, \boldsymbol{0})
\end{aligned} \tag{15}$$

Then the following conclusions hold:

(i) *$\boldsymbol{M}$ is maximally monotone.*

(ii) *$\boldsymbol{S}$ is monotone and l-Lipschitzian, where*

$$l = (\max\{\max_{i=1,\cdots,m} \|K_i\|^2, \max_{i=1,\cdots,m} \|M_i\|^2, \sum_{i=1}^m \|L_i\|^2\})^{\frac{1}{2}}. \tag{16}$$

(iii) $\boldsymbol{Q}$ *is* μ^{-1}*-cocoercive.*
(iv) *For any* $\bar{x} \in \mathcal{H}$*,* $\bar{x}$ *is a solution to Problem 2, if and only if* $\bar{x} \in \operatorname{zer}(\boldsymbol{M} + \boldsymbol{S} + \boldsymbol{Q})$*.*

Proof. (i) Since $A, \boldsymbol{B}$, and $\boldsymbol{D}$ are maximally monotone, it follows from [23] Proposition 20.22 and Proposition 20.23 that the set-valued operator $\boldsymbol{M}$ is maximally monotone.

(ii) By taking two arbitrary elements $\boldsymbol{x} = (x, \boldsymbol{p}, \boldsymbol{q}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{v})$ and $\hat{\boldsymbol{x}} = (\hat{x}, \hat{\boldsymbol{p}}, \hat{\boldsymbol{q}}, \hat{\boldsymbol{z}}, \hat{\boldsymbol{y}}, \hat{\boldsymbol{v}})$ in $\boldsymbol{\mathcal{K}}$, we obtain

$$\begin{aligned}
&\langle \boldsymbol{x} - \hat{\boldsymbol{x}}, \boldsymbol{S}\boldsymbol{x} - \boldsymbol{S}\hat{\boldsymbol{x}} \rangle \\
=&\langle (x - \hat{x}, p_1 - \hat{p}_1, \cdots, p_m - \hat{p}_m, q_1 - \hat{q}_1, \cdots, q_m - \hat{q}_m, z_1 - \hat{z}_1, \cdots, z_m - \hat{z}_m, y_1 - \hat{y}_1, \cdots, y_m - \hat{y}_m, \\
& v_1 - \hat{v}_1, \cdots, v_m - \hat{v}_m), (\sum_{i=1}^m L_i^*(v_i - \hat{v}_i), -K_1(z_1 - \hat{z}_1), \cdots, -K_m(z_m - \hat{z}_m), -M_1(y_1 - \hat{y}_1), \cdots, -M_m(y_m - \hat{y}_m), \\
& K_1^*(p_1 - \hat{p}_1), \cdots, K_m^*(p_m - \hat{p}_m), M_1^*(q_1 - \hat{q}_1), \cdots, M_m^*(q_m - \hat{q}_m), -L_1(x - \hat{x}), \cdots, -L_m(x - \hat{x}) \rangle \\
=&\sum_{i=1}^m \langle x - \hat{x}, L_i^*(v_i - \hat{v}_i) \rangle - \sum_{i=1}^m \langle p_i - \hat{p}_i, K_i(z_i - \hat{z}_i) \rangle - \sum_{i=1}^m \langle q_i - \hat{q}_i, M_i(y_i - \hat{y}_i) \rangle + \sum_{i=1}^m \langle z_i - \hat{z}_i, K_i^*(p_i - \hat{p}_i) \rangle \\
& + \sum_{i=1}^m \langle y_i - \hat{y}_i, M_i^*(q_i - \hat{q}_i) \rangle - \sum_{i=1}^m \langle v_i - \hat{v}_i, L_i(x - \hat{x}) \rangle \\
=&0,
\end{aligned} \tag{17}$$

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

$$\begin{aligned}
&\|\boldsymbol{S}\boldsymbol{x} - \boldsymbol{S}\hat{\boldsymbol{x}}\| \\
&= \|(\sum_{i=1}^m L_i^*(v_i - \hat{v}_i), -K_1(z_1 - \hat{z}_1), \cdots, -K_m(z_m - \hat{z}_m), -M_1(y_1 - \hat{y}_1), \cdots, -M_m(y_m - \hat{y}_m), \\
&\quad + K_1^*(p_1 - \hat{p}_1), \cdots, K_m^*(p_m - \hat{p}_m), M_1^*(q_1 - \hat{q}_1), \cdots, M_m^*(q_m - \hat{q}_m), -L_1(x - \hat{x}), \cdots, -L_m(x - \hat{x})\| \\
&= (\| \sum_{i=1}^m L_i^*(v_i - \hat{v}_i)\|^2 + \sum_{i=1}^m \|K_i(z_i - \hat{z}_i)\|^2 + \sum_{i=1}^m \|M_i(y_i - \hat{y}_i)\|^2 + \sum_{i=1}^m \|K_i^*(p_i - \hat{p}_i)\|^2 \\
&\quad + \sum_{i=1}^m \|M_i^*(q_i - \hat{q}_i)\|^2 + \sum_{i=1}^m \|L_i^*(x - \hat{x})\|^2)^{\frac{1}{2}} \\
&\le ((\sum_{i=1}^m \|L_i\|^2) \sum_{i=1}^m \|v_i - \hat{v}_i\|^2 + \sum_{i=1}^m \|K_i\|^2 \|z_i - \hat{z}_i\|^2 + \sum_{i=1}^m \|M_i\|^2 \|y_i - \hat{y}_i\|^2 + \sum_{i=1}^m \|K_i\|^2 \|p_i - \hat{p}_i\|^2 \\
&\quad + \sum_{i=1}^m \|M_i\|^2 \|q_i - \hat{q}_i\|^2 + \sum_{i=1}^m \|L_i\|^2 \|x - \hat{x}\|^2)^{\frac{1}{2}} \\
&\le ((\sum_{i=1}^m \|L_i\|^2) \sum_{i=1}^m \|v_i - \hat{v}_i\|^2 + \max_{i=1,\cdots,m} \|K_i\|^2 \sum_{i=1}^m \|z_i - \hat{z}_i\|^2 + \max_{i=1,\cdots,m} \|M_i\|^2 \sum_{i=1}^m \|y_i - \hat{y}_i\|^2 \\
&\quad + \max_{i=1,\cdots,m} \|K_i\|^2 \sum_{i=1}^m \|p_i - \hat{p}_i\|^2 + \max_{i=1,\cdots,m} \|M_i\|^2 \sum_{i=1}^m \|q_i - \hat{q}_i\|^2 + \sum_{i=1}^m \|L_i\|^2 \|x - \hat{x}\|^2)^{\frac{1}{2}} \\
&\le l(\|x - \hat{x}\|^2 + \sum_{i=1}^m \|p_i - \hat{p}_i\|^2 + \sum_{i=1}^m \|q_i - \hat{q}_i\|^2 + \sum_{i=1}^m \|z_i - \hat{z}_i\|^2 + \sum_{i=1}^m \|y_i - \hat{y}_i\|^2 + \sum_{i=1}^m \|v_i - \hat{v}_i\|^2)^{\frac{1}{2}} \\
&= l\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|.
\end{aligned} \tag{18}$$

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

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

$$\begin{aligned}
&\langle \boldsymbol{x} - \hat{\boldsymbol{x}}, \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\hat{\boldsymbol{x}} \rangle \ge \langle x - \hat{x}, Cx - C\hat{x} \rangle \\
&\ge \mu^{-1}\|Cx - C\hat{x}\|^2 = \mu^{-1}\|\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\hat{\boldsymbol{x}}\|^2.
\end{aligned} \tag{19}$$

Then, the operator $\boldsymbol{Q}$ is μ^{-1}-cocoercive.

(iv) Let $\bar{x} \in \mathcal{H}$, then

$$\bar{x} \text{ solves } (3) \Leftrightarrow \exists(\bar{x},\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{y}}) \in \mathcal{H} \oplus \boldsymbol{\mathcal{X}} \oplus \boldsymbol{\mathcal{Y}} \oplus \boldsymbol{\mathcal{G}}: \begin{cases} z - \sum_{i=1}^{m} L_i^* K_i^* \bar{p}_i \in A\bar{x} + C\bar{x}, \\ K_i(L_i\bar{x} - \bar{y}_i - r_i) \in B_i^{-1}\bar{p}_i, i = 1,\ldots,m, \\ M_i\bar{y}_i \in D_i^{-1}\bar{q}_i, i = 1,\ldots,m, \\ K_i^*\bar{p}_i = M_i^*\bar{q}_i, i = 1,\ldots,m. \end{cases}$$

$$\Leftrightarrow \begin{array}{l} \exists(\bar{x},\overline{\boldsymbol{p}},\overline{\boldsymbol{q}}) \in \mathcal{H} \oplus \boldsymbol{\mathcal{X}} \oplus \boldsymbol{\mathcal{Y}} \\ \exists(\overline{\boldsymbol{z}},\overline{\boldsymbol{y}},\overline{\boldsymbol{v}}) \in \boldsymbol{\mathcal{G}} \oplus \boldsymbol{\mathcal{G}} \oplus \boldsymbol{\mathcal{G}} \end{array} : \begin{cases} 0 \in -z + A\bar{x} + \sum_{i=1}^{m} L_i^*\bar{v}_i + C\bar{x} \\ 0 \in -K_i\bar{z}_i + B_i^{-1}\bar{p}_i, i = 1,\ldots,m, \\ 0 \in -M_i\bar{y}_i + D_i^{-1}\bar{q}_i, i = 1,\ldots,m, \\ 0 = K_i^*\bar{p}_i - \bar{v}_i, i = 1,\ldots,m, \\ 0 = M_i^*\bar{q}_i - \bar{v}_i, i = 1,\ldots,m, \\ 0 = r_i + \bar{z}_i + \bar{y}_i - L_i\bar{x}, i = 1,\ldots,m \end{cases} \tag{20}$$

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

Therefore, if $(\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{z}},\overline{\boldsymbol{y}},\overline{\boldsymbol{v}})$ is a solution of (4), then there exists $\bar{x} \in \mathcal{H}$ such that $(\bar{x},\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{z}},\overline{\boldsymbol{y}},\overline{\boldsymbol{v}})$ is a primal-dual solution of Problem 2. □

3.1. Primal-Dual Forward–Backward Splitting Type Algorithm

In this subsection, we prove the convergence of the primal-dual forward–backward splitting type algorithm (5). By Lemma 2, $\boldsymbol{M} + \boldsymbol{S}$ is maximally monotone, and $\boldsymbol{Q}$ is cocoercive. It is natural to use the forward–backward splitting algorithm. However, the resolvent operator of $\boldsymbol{M} + \boldsymbol{S}$ does not have a closed-form solution. To overcome this difficulty, Boţ and Hendrich [4] introduced a special precondition to the forward–backward splitting algorithm and obtained the primal-dual splitting algorithm (5). In the following, we present an improved convergence analysis of the primal-dual forward–backward splitting type algorithm (5), which sharpens the selection of iterative parameters.

Theorem 1. *Consider Problem 2, suppose that*

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

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

(i) $2\beta > 1$, *where* $\beta = \mu^{-1}\left(\frac{1}{\tau} - \sum_{i=1}^{m} \sigma_i \|L_i\|^2\right)$;

(ii) $(1 - \bar{\alpha}) \min_{i=1,\ldots,m}\left\{\frac{1}{\tau}, \frac{1}{\theta_{1,i}}, \frac{1}{\theta_{2,i}}, \frac{1}{\gamma_{1,i}}, \frac{1}{\gamma_{2,i}}, \frac{1}{\sigma_i}\right\} > 0$, *where $\bar{\alpha}$ is defined by*

$$\bar{\alpha} = \max\left\{ \sqrt{\tau \sum_{i=1}^{m} \sigma_i \|L_i\|^2}, \max_{j=1,\ldots,m}\left\{ \sqrt{\theta_{1,j}\gamma_{1,j}\|K_j\|^2}, \sqrt{\theta_{2,j}\gamma_{2,j}\|M_j\|^2} \right\} \right\}. \tag{21}$$

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

Consider the iterative sequences generated by (5). Then, there exists a primal-dual solution $(\bar{x},\overline{\boldsymbol{p}},\overline{\boldsymbol{q}},\overline{\boldsymbol{z}},\overline{\boldsymbol{y}},\overline{\boldsymbol{v}})$ to Problem 2 such that $x_n \rightharpoonup \bar{x}$, $p_{i,n} \rightharpoonup \bar{p}_i$, $q_{i,n} \rightharpoonup \bar{q}_i$, $z_{i,n} \rightharpoonup \bar{z}_i$, $y_{i,n} \rightharpoonup \bar{y}_i$, and $v_{i,n} \rightharpoonup \bar{v}_i$ for any $i = 1,\ldots,m$ as $n \to +\infty$.

Proof. Let $\mathcal{H}$, A, C, $\mathcal{X}_i$, $\mathcal{Y}_i$, $\mathcal{G}_i$, B_i, D_i, L_i, K_i, M_i, $i = 1,\cdots,m$ be defined as in Problem 2. Let the real Hilbert space $\boldsymbol{\mathcal{K}} = \mathcal{H} \oplus \boldsymbol{\mathcal{X}} \oplus \boldsymbol{\mathcal{Y}} \oplus \boldsymbol{\mathcal{G}} \oplus \boldsymbol{\mathcal{G}} \oplus \boldsymbol{\mathcal{G}}$ and

$$\begin{cases} \boldsymbol{p} = (p_1,\ldots,p_m) \\ \boldsymbol{q} = (q_1,\ldots,q_m) \\ \boldsymbol{y} = (y_1,\ldots,y_m) \end{cases} \text{and} \begin{cases} \boldsymbol{z} = (z_1,\ldots,z_m) \\ \boldsymbol{v} = (v_1,\ldots,v_m) \\ \boldsymbol{r} = (r_1,\ldots,r_m). \end{cases}$$

Define

$$\boldsymbol{V}:\mathcal{K}\to\mathcal{K},(x,\boldsymbol{p},\boldsymbol{q},\boldsymbol{z},\boldsymbol{y},\boldsymbol{v})\mapsto\left(\frac{x}{\tau},\frac{\boldsymbol{p}}{\theta_1},\frac{\boldsymbol{q}}{\theta_2},\frac{\boldsymbol{z}}{\gamma_1},\frac{\boldsymbol{y}}{\gamma_2},\frac{\boldsymbol{v}}{\sigma}\right)+\left(-\sum_{i=1}^m L_i^*v_i,\widetilde{K}\boldsymbol{z},\widetilde{M}\boldsymbol{y},\widetilde{K}^*\boldsymbol{p},\widetilde{M}^*\boldsymbol{q},-L_1x,\ldots,-L_mx\right).$$

Further, for positive real values $\tau,\theta_{1,i},\theta_{2,i},\gamma_{1,i},\gamma_{2,i},\sigma_i\in\mathbb{R}_{++},i=1,\ldots,m$, define the notations

$$\begin{cases}\frac{\boldsymbol{p}}{\theta_1}=\left(\frac{p_1}{\theta_{1,1}},\ldots,\frac{p_m}{\theta_{1,m}}\right)\\ \frac{\boldsymbol{q}}{\theta_2}=\left(\frac{q_1}{\theta_{2,1}},\ldots,\frac{q_m}{\theta_{2,m}}\right)\end{cases},\quad\begin{cases}\frac{\boldsymbol{z}}{\gamma_1}=\left(\frac{z_1}{\gamma_{1,1}},\ldots,\frac{z_m}{\gamma_{1,m}}\right)\\ \frac{\boldsymbol{y}}{\gamma_2}=\left(\frac{y_1}{\gamma_{2,1}},\ldots,\frac{y_m}{\gamma_{2,m}}\right)\end{cases},\quad\left\{\frac{\boldsymbol{v}}{\sigma}=\left(\frac{v_1}{\sigma_1},\ldots,\frac{v_m}{\sigma_m}\right).\right.$$

Then, (5) can be rewritten in the form of

$$(\forall n\geq 0)\left\lfloor\begin{array}{l}\frac{x_n-\widetilde{x}_n}{\tau}-\sum_{i=1}^m L_i^*(v_{i,n}-\widetilde{v}_{i,n})-Cx_n\in -z+A\widetilde{x}_n+\sum_{i=1}^m L_i^*\widetilde{v}_{i,n}\\ \text{For } i=1,\ldots,m\\ \left\lfloor\begin{array}{l}\frac{p_{i,n}-\widetilde{p}_{i,n}}{\theta_{1,i}}+K_i(z_{i,n}-\widetilde{z}_{i,n})\in B_i^{-1}\widetilde{p}_{i,n}-K_i\widetilde{z}_{i,n}\\ \frac{q_{i,n}-\widetilde{q}_{i,n}}{\theta_{2,i}}+M_i(y_{i,n}-\widetilde{y}_{i,n})\in D_i^{-1}\widetilde{q}_{i,n}-M_i\widetilde{y}_{i,n}\\ \frac{z_{i,n}-\widetilde{z}_{i,n}}{\gamma_{1,i}}+K_i^*(p_{i,n}-\widetilde{p}_{i,n})=-\widetilde{v}_{i,n}+K_i^*\widetilde{p}_{i,n}\\ \frac{y_{i,n}-\widetilde{y}_{i,n}}{\gamma_{2,i}}+M_i^*(q_{i,n}-\widetilde{q}_{i,n})=-\widetilde{v}_{i,n}+M_i^*\widetilde{q}_{i,n}\\ \frac{v_{i,n}-\widetilde{v}_{i,n}}{\sigma_i}-L_i(x_n-\widetilde{x}_n)=r_i+\widetilde{z}_{i,n}+\widetilde{y}_{i,n}-L_i\widetilde{x}_n\end{array}\right.\\ x_{n+1}=x_n+\lambda_n(\widetilde{x}_n-x_n).\end{array}\right.\tag{22}$$

Let

$$\begin{cases}\boldsymbol{p}_n=(p_{1,n},\ldots p_{m,n})\in\boldsymbol{\mathcal{X}}\\ \boldsymbol{q}_n=(q_{1,n},\ldots,q_{m,n})\in\boldsymbol{\mathcal{Y}}\\ \boldsymbol{z}_n=(z_{1,n},\ldots,z_{m,n})\in\boldsymbol{\mathcal{G}}\\ \boldsymbol{y}_n=(y_{1,n},\ldots,y_{m,n})\in\boldsymbol{\mathcal{G}}\\ \boldsymbol{v}_n=(v_{1,n},\ldots,v_{m.n})\in\boldsymbol{\mathcal{G}}\end{cases}\qquad\begin{cases}\widetilde{\boldsymbol{p}}_{\boldsymbol{n}}=(\widetilde{p}_{1,n},\ldots\widetilde{p}_{m,n})\in\boldsymbol{\mathcal{X}}\\ \widetilde{\boldsymbol{q}}_{\boldsymbol{n}}=(\widetilde{q}_{1,n},\ldots,\widetilde{q}_{m,n})\in\boldsymbol{\mathcal{Y}}\\ \widetilde{\boldsymbol{z}}_n=(\widetilde{z}_{1,n},\ldots,\widetilde{z}_{m,n})\in\boldsymbol{\mathcal{G}}\\ \widetilde{\boldsymbol{y}}_n=(\widetilde{y}_{1,n},\ldots,\widetilde{y}_{m,n})\in\boldsymbol{\mathcal{G}}\\ \widetilde{\boldsymbol{v}}_n=(\widetilde{v}_{1.n},\ldots,\widetilde{v}_{m,n})\in\boldsymbol{\mathcal{G}}\end{cases}$$

and

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

Therefore, the iteration scheme in (22) is equivalent to

$$(\forall n\geq 0)\left\lfloor\begin{array}{l}\boldsymbol{V}(\boldsymbol{x}_n-\widetilde{\boldsymbol{x}}_n)-\boldsymbol{Q}\boldsymbol{x}_n\in(\boldsymbol{M}+\boldsymbol{S})\widetilde{\boldsymbol{x}}_{\boldsymbol{n}}\\ \boldsymbol{x}_{n+1}=\boldsymbol{x}_n+\lambda_n(\widetilde{\boldsymbol{x}}_n-\boldsymbol{x}_n).\end{array}\right.\tag{23}$$

We introduce the notations

$$\boldsymbol{A}_{\mathcal{K}}:=\boldsymbol{V}^{-1}(\boldsymbol{M}+\boldsymbol{S})\text{ and }\boldsymbol{B}_{\mathcal{K}}:=\boldsymbol{V}^{-1}\boldsymbol{Q}.\tag{24}$$

Then, for any $n \geq 0$, we have

$$\begin{aligned} & \boldsymbol{V}(\boldsymbol{x}_n - \widetilde{\boldsymbol{x}}_n) - \boldsymbol{Q}\boldsymbol{x}_n \in (\boldsymbol{M} + \boldsymbol{S})\widetilde{\boldsymbol{x}}_{\boldsymbol{n}} \\ \Leftrightarrow & \boldsymbol{V}\boldsymbol{x}_n - \boldsymbol{Q}\boldsymbol{x}_n \in (\boldsymbol{V} + \boldsymbol{M} + \boldsymbol{S})\widetilde{\boldsymbol{x}}_{\boldsymbol{n}} \\ \Leftrightarrow & \boldsymbol{x}_n - \boldsymbol{V}^{-1}\boldsymbol{Q}\boldsymbol{x}_n \in \Big(\mathrm{Id} + \boldsymbol{V}^{-1}(\boldsymbol{M} + \boldsymbol{S})\Big)\widetilde{\boldsymbol{x}}_n \\ \Leftrightarrow & \widetilde{\boldsymbol{x}}_n = \Big(\mathrm{Id} + \boldsymbol{V}^{-1}(\boldsymbol{M} + \boldsymbol{S})\Big)^{-1}\Big(\boldsymbol{x}_n - \boldsymbol{V}^{-1}\boldsymbol{Q}\boldsymbol{x}_n\Big) \\ \Leftrightarrow & \widetilde{\boldsymbol{x}}_n = (\mathrm{Id} + \boldsymbol{A}_{\mathcal{K}})^{-1}(\boldsymbol{x}_n - \boldsymbol{B}_{\mathcal{K}}\boldsymbol{x}_n), \end{aligned} \tag{25}$$

which can be written as

$$\widetilde{\boldsymbol{x}}_{\boldsymbol{n}} = J_{\boldsymbol{A}_{\mathcal{K}}}(\boldsymbol{x}_n - \boldsymbol{B}_{\mathcal{K}}\boldsymbol{x}_n). \tag{26}$$

Thus, the iterative scheme in (23) becomes

$$(\forall n \geq 0) \left\lfloor \begin{array}{l} \widetilde{\boldsymbol{x}}_n = J_{\boldsymbol{A}_{\mathcal{K}}}(\boldsymbol{x}_n - \boldsymbol{B}_{\mathcal{K}}\boldsymbol{x}_n) \\ \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \lambda_n(\widetilde{\boldsymbol{x}}_{\boldsymbol{n}} - \boldsymbol{x}_n). \end{array} \right. \tag{27}$$

We then introduce the Hilbert space $\mathcal{K}_V$ with inner product and norm, respectively, defined, for $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}$, via

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\mathcal{K}_V} = \langle \boldsymbol{x}, \boldsymbol{V}\boldsymbol{y} \rangle_{\mathcal{K}} \text{ and } \|\boldsymbol{x}\|_{\mathcal{K}_V} = \sqrt{\langle \boldsymbol{x}, \boldsymbol{V}\boldsymbol{x} \rangle_{\mathcal{K}}} \tag{28}$$

Since $\boldsymbol{M} + \boldsymbol{S}$ and $\boldsymbol{Q}$ are maximally monotone on $\mathcal{K}$, the operators $\boldsymbol{A}_{\mathcal{K}}$ and $\boldsymbol{B}_{\mathcal{K}}$ are maximally monotone on $\mathcal{K}_V$. Moreover, since $\boldsymbol{V}$ is self-adjoint and ρ-strongly positive, one can easily see that weak and strong convergence in $\mathcal{K}_V$ are equivalent with weak and strong convergence in $\mathcal{K}$, respectively. In the following, we prove that $\boldsymbol{B}_{\mathcal{K}}$ is β-cocoercive on $\mathcal{K}_V$. In fact, let $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{K}_V$, we have

$$\begin{aligned} \|\boldsymbol{B}_{\mathcal{K}}\boldsymbol{x} - \boldsymbol{B}_{\mathcal{K}}\boldsymbol{y}\|^2_{\mathcal{K}_V} &= \langle \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{y}, \boldsymbol{V}^{-1}\boldsymbol{Q}\boldsymbol{x} - \boldsymbol{V}^{-1}\boldsymbol{Q}\boldsymbol{y} \rangle_{\mathcal{K}} \\ &= \langle Cx - Cy, (\frac{1}{\tau}Id - \sum_{i=1}^{m}\sigma_i L_i^* L_i)^{-1}(Cx - Cy) \rangle \\ &\leq (\frac{1}{\tau} - \sum_{i=1}^{m}\sigma_i\|L_i\|^2)^{-1}\langle Cx - Cy, Cx - Cy \rangle \\ &= (\frac{1}{\tau} - \sum_{i=1}^{m}\sigma_i\|L_i\|^2)^{-1}\|Cx - Cy\|^2. \end{aligned} \tag{29}$$

It follows from the above inequality that we obtain

$$\begin{aligned} \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{B}_{\mathcal{K}}\boldsymbol{x} - \boldsymbol{B}_{\mathcal{K}}\boldsymbol{y} \rangle_{\mathcal{K}_V} &= \langle \boldsymbol{x} - \boldsymbol{y}, \boldsymbol{Q}\boldsymbol{x} - \boldsymbol{Q}\boldsymbol{y} \rangle_{\mathcal{K}} \\ &= \langle x - y, Cx - Cy \rangle \\ &\geq \mu^{-1}\|Cx - Cy\| \\ &\geq \mu^{-1}(\frac{1}{\tau} - \sum_{i=1}^{m}\sigma_i\|L_i\|^2)\|\boldsymbol{B}_{\mathcal{K}}\boldsymbol{x} - \boldsymbol{B}_{\mathcal{K}}\boldsymbol{y}\|^2_{\mathcal{K}_V} \\ &= \beta\|\boldsymbol{B}_{\mathcal{K}}\boldsymbol{x} - \boldsymbol{B}_{\mathcal{K}}\boldsymbol{y}\|^2_{\mathcal{K}_V}, \end{aligned} \tag{30}$$

where $\beta = \mu^{-1}(\frac{1}{\tau} - \sum_{i=1}^{m}\sigma_i\|L_i\|^2)$.

Since $2\beta > 1$, so the iteration scheme (27) could be viewed as a special case of the forward–backward splitting algorithm. By Corollary 28.9 of [23], the iterative sequences $\{\boldsymbol{x}_n\}$ converge weakly to a point $\bar{\boldsymbol{x}} = (\bar{x}, \overline{\boldsymbol{p}}, \overline{\boldsymbol{q}}, \bar{z}, \overline{\boldsymbol{y}}, \overline{v})$ in $\mathrm{zer}(\boldsymbol{A}_{\mathcal{K}} + \boldsymbol{B}_{\mathcal{K}})$. It is observed that $\mathrm{zer}(\boldsymbol{A}_{\mathcal{K}} + \boldsymbol{B}_{\mathcal{K}}) = \mathrm{zer}\big(\boldsymbol{V}^{-1}(\boldsymbol{M} + \boldsymbol{S} + \boldsymbol{Q})\big) = \mathrm{zer}(\boldsymbol{M} + \boldsymbol{S} + \boldsymbol{Q})$. Then, we obtain that

$x_n \rightharpoonup \bar{x}, p_{i,n} \rightharpoonup \bar{p}_i, q_{i,n} \rightharpoonup \bar{q}_i, z_{i,n} \rightharpoonup \bar{z}_i, y_{i,n} \rightharpoonup \bar{y}_i$, and $v_{i,n} \rightharpoonup \bar{v}_i$ for any $i = 1, \ldots, m$ as $n \to +\infty$. This completes the proof. □

Remark 1. *Theorem 1 improves the FB_BH algorithm (5) in the following aspects:*

(i) The parameter conditions of $\tau, \theta_{1,i}, \theta_{2,i}, \gamma_{1,i}, \gamma_{2,i}$ and σ_i, for any $i = 1, \cdots, m$, are relaxed.

(ii) The range of relaxing parameters $\{\lambda_n\}$ has been expanded. We have improved the relaxation parameter form $\{\lambda_n\} \subseteq (0, 1]$ to $\{\lambda_n\} \subseteq [0, 2 - \frac{1}{2\beta}]$. Since $2\beta > 1$, then $2 - \frac{1}{2\beta} > 1$.

3.2. Primal-Dual Forward–Backward-Half–Forward Splitting Type Algorithm

By Lemma 2, the primal-dual pair of monotone inclusions (3) and (4) are equivalent to the monotone inclusions of the sum of $\boldsymbol{M} + \boldsymbol{S} + \boldsymbol{Q}$, where $\boldsymbol{M}$ is maximally monotone, $\boldsymbol{S}$ is monotone, Lipschitz, and $\boldsymbol{Q}$ is cocoercive. It is well-known that a μ^{-1}-cocoercive operator is μ-lipschitz continuous. The forward–backward–forward splitting type algorithm (8) does not make use of the cocoercive property of $\boldsymbol{Q}$. In the following, we propose a forward–backward–half-forward splitting type algorithm for solving (3) and (4) and prove its convergence.

Theorem 2. *For Problem 3, suppose that*

$$z \in \operatorname{ran}\left(A + \sum_{i=1}^{m} L_i^*((K_i^* \circ B_i \circ K_i)\Box(M_i^* \circ D_i \circ M_i))(L_i \cdot -r_i) + C\right). \tag{31}$$

Let $\{\gamma_n\} \subseteq [\eta, \chi - \eta]$, where $\eta \in (0, \chi/2]$, l is defined by (16), and $\chi := \frac{4\mu^{-1}}{1+\sqrt{1+16(\mu^{-1})^2 l^2}}$. Let $x_0 \in \mathcal{H}$, and for any $i = 1, \ldots, m$, let $p_{i,0} \in \mathcal{X}_i, q_{i,0} \in \mathcal{Y}_i, z_{i,0} \in \mathcal{G}_i, y_{i,0} \in \mathcal{G}_i$ and $v_{i,0} \in \mathcal{G}_i$. Set

$$(\forall n \geq 0)\left\lfloor \begin{array}{l} \widetilde{x}_n = J_{\gamma_n A}(x_n - \gamma_n(Cx_n + \sum_{i=1}^{m} L_i^* v_{i,n} - z)) \\ \text{For } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} \widetilde{p}_{i,n} = J_{\gamma_n B_i^{-1}}(p_{i,n} + \gamma_n K_i z_{i,n}) \\ \widetilde{q}_{i,n} = J_{\gamma_n D_i^{-1}}(q_{i,n} + \gamma_n M_i y_{i,n}) \\ u_{1,i,n} = z_{i,n} - \gamma_n(K_i^* p_{i,n} - v_{i,n} - \gamma_n(L_i x_n - r_i)) \\ u_{2,i,n} = y_{i,n} - \gamma_n(M_i^* q_{i,n} - v_{i,n} - \gamma_n(L_i x_n - r_i)) \\ \widetilde{z}_{i,n} = \dfrac{1+\gamma_n^2}{1+2\gamma_n^2}\left(u_{1,i,n} - \dfrac{\gamma_n^2}{1+\gamma_n^2} u_{2,i,n}\right) \\ \widetilde{y}_{i,n} = \dfrac{1+\gamma_n^2}{1+2\gamma_n^2}\left(u_{2,i,n} - \dfrac{\gamma_n^2}{1+\gamma_n^2} u_{1,i,n}\right) \\ \widetilde{v}_{i,n} = v_{i,n} + \gamma_n(L_i x_n - r_i - \widetilde{z}_{i,n} - \widetilde{y}_{i,n}) \end{array} \right. \\ x_{n+1} = \widetilde{x}_n + \gamma_n \sum_{i=1}^{m} L_i^*(v_{i,n} - \widetilde{v}_{i,n}) \\ \text{For } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} p_{i,n+1} = \widetilde{p}_{i,n} - \gamma_n K_i(z_{i,n} - \widetilde{z}_{i,n}) \\ q_{i,n+1} = \widetilde{q}_{i,n} - \gamma_n M_i(y_{i,n} - \widetilde{y}_{i,n}) \\ z_{i,n+1} = \widetilde{z}_{i,n} + \gamma_n K_i^*(p_{i,n} - \widetilde{p}_{i,n}) \\ y_{i,n+1} = \widetilde{y}_{i,n} + \gamma_n M_i^*(q_{i,n} - \widetilde{q}_{i,n}) \\ v_{i,n+1} = \widetilde{v}_{i,n} - \gamma_n L_i(x_n - \widetilde{x}_n), \end{array} \right. \end{array} \right. \tag{32}$$

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

Proof. Notice that (32) is equivalent to

$$
(\forall n \geq 0) \left\lfloor \begin{array}{l}
(x_n - \gamma_n(Cx_n + \sum_{i=1}^{m} L_i^* v_{i,n})) \in (\mathrm{Id} + \gamma_n(A \cdot - z))\widetilde{x}_n \\
\text{for } i = 1, \ldots, m \\
\left\lfloor \begin{array}{l}
p_{i,n} + \gamma_n K_i z_{i,n} \in (\mathrm{Id} + \gamma_n B_i^{-1})\widetilde{p}_{i,n} \\
q_{i,n} + \gamma_n M_i y_{i,n} \in (\mathrm{Id} + \gamma_n D_i^{-1})\widetilde{q}_{i,n} \\
z_{i,n} - \gamma_n K_i^* p_{i,n} = \widetilde{z}_{i,n} - \gamma_n \widetilde{v}_{i,n} \\
y_{i,n} - \gamma_n M_i^* q_{i,n} = \widetilde{y}_{i,n} - \gamma_n \widetilde{v}_{i,n} \\
v_{i,n} + \gamma_n L_i x_n = \widetilde{v}_{i,n} + \gamma_n(r_i + \widetilde{z}_{i,n} + \widetilde{y}_{i,n})
\end{array} \right. \\
x_{n+1} = \widetilde{x}_n + \gamma_n \sum_{i=1}^{m} L_i^*(v_{i,n} - \widetilde{v}_{i,n}) \\
\text{for } i = 1, \ldots, m \\
\left\lfloor \begin{array}{l}
p_{i,n+1} = \widetilde{p}_{i,n} - \gamma_n K_i(z_{i,n} - \widetilde{z}_{i,n}) \\
q_{i,n+1} = \widetilde{q}_{i,n} - \gamma_n M_i(y_{i,n} - \widetilde{y}_{i,n}) \\
z_{i,n+1} = \widetilde{z}_{i,n} + \gamma_n K_i^*(p_{i,n} - \widetilde{p}_{i,n}) \\
y_{i,n+1} = \widetilde{y}_{i,n} + \gamma_n M_i^*(q_{i,n} - \widetilde{q}_{i,n}) \\
v_{i,n+1} = \widetilde{v}_{i,n} - \gamma_n L_i(x_n - \widetilde{x}_n).
\end{array} \right.
\end{array} \right. \tag{33}
$$

Using the notations in Theorem 1, the iteration scheme (33) could be equivalently written as

$$
(\forall n \geq 0) \left\lfloor \begin{array}{l}
\boldsymbol{x}_n - \gamma_n(\boldsymbol{S} + \boldsymbol{Q})\boldsymbol{x}_n \in (\mathrm{Id} + \gamma_n \boldsymbol{M})\widetilde{\boldsymbol{x}}_n \\
\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \gamma_n(\boldsymbol{S}\boldsymbol{x}_n - \boldsymbol{S}\widetilde{\boldsymbol{x}}_n),
\end{array} \right. \tag{34}
$$

which is equivalent to

$$
(\forall n \geq 0) \left\lfloor \begin{array}{l}
\widetilde{\boldsymbol{x}}_n = J_{\gamma_n \boldsymbol{M}}(\boldsymbol{x}_n - \gamma_n(\boldsymbol{S} + \boldsymbol{Q})\boldsymbol{x}_n) \\
\boldsymbol{x}_{n+1} = \widetilde{\boldsymbol{x}}_n + \gamma_n(\boldsymbol{S}\boldsymbol{x}_n - \boldsymbol{S}\widetilde{\boldsymbol{x}}_n).
\end{array} \right. \tag{35}
$$

Therefore, (35) is an instance of the forward–backward–half-forward splitting algorithm in $\boldsymbol{\mathcal{K}}$, whose convergence has been investigated in [22].

Let $(\bar{x}, \overline{\boldsymbol{p}}, \overline{\boldsymbol{q}}, \overline{z}, \overline{\boldsymbol{y}}, \overline{v}) \in \mathrm{zer}(\boldsymbol{M} + \boldsymbol{S} + \boldsymbol{Q})$, then, $(\bar{x}, \overline{\boldsymbol{p}}, \overline{\boldsymbol{q}}, \overline{z}, \overline{\boldsymbol{y}}, \overline{v}) \in \mathrm{zer}(\boldsymbol{M} + \boldsymbol{S} + \boldsymbol{Q}) \in \mathcal{H} \oplus \mathcal{X} \oplus \mathcal{Y} \oplus \mathcal{G} \oplus \mathcal{G} \oplus \mathcal{G}$ is a primal-dual solution to Problem 1.1. By Theorem 2.3 of [22], we have $x_n \rightharpoonup \bar{x}$, and for $i = 1, \ldots, m$, $p_{i,n} \rightharpoonup \bar{p}_i$, $q_{i,n} \rightharpoonup \bar{q}_i$, $z_{i,n} \rightharpoonup \bar{z}_i$, $y_{i,n} \rightharpoonup \bar{y}_i$, and $v_{i,n} \rightharpoonup \bar{v}_i$ as $n \to +\infty$. This completes the proof. □

Remark 2. *In contrast to the FBF_BH algorithm (8), the proposed algorithm (32) has two advantages:*

(i) The calculation of the cocoercive operator in (8) requires twice, while it only requires once in the proposed algorithm (32).

(ii) The range of the iterative parameter of the proposed algorithm (32) is larger than algorithm (8).

3.3. Applications to Convex Minimization Problems

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

Problem 3. *Let $\mathcal{H}$ be a real Hilbert space, let $z \in \mathcal{H}$ and $h : \mathcal{H} \to R$ is differentiable with μ-Lipschitzian gradient for some $\mu > 0$. Let $f \in \Gamma_0(\mathcal{H})$. For every $i = 1, \cdots, m$, let $\mathcal{G}_i, \mathcal{X}_i, \mathcal{Y}_i$ be real Hilbert spaces, $r_i \in \mathcal{G}_i$, let $g_i \in \Gamma_0(\mathcal{X}_i)$ and $l_i \in \Gamma_0(\mathcal{Y}_i)$ and consider the nonzero linear* *bounded operators* $L_i : \mathcal{H} \to \mathcal{G}_i, K_i : \mathcal{G}_i \to \mathcal{X}_i$ *and* $M_i : \mathcal{G}_i \to \mathcal{Y}_i$*. The primal optimization problem is*

$$\min_{x \in \mathcal{H}} \left\{ f(x) + \sum_{i=1}^{m} ((g_i \circ K_i) \Box (l_i \circ M_i))(L_i x - r_i) + h(x) - \langle x, z \rangle \right\}, \tag{36}$$

together with its conjugate dual problem

$$\max_{(\boldsymbol{p},\boldsymbol{q}) \in \boldsymbol{\mathcal{X}} \oplus \boldsymbol{\mathcal{Y}}, K_i^* p_i = M_i^* q_i, i=1,\ldots,m} \left\{ -(f^* \Box h^*) \left(z - \sum_{i=1}^{m} L_i^* K_i^* p_i \right) - \sum_{i=1}^{m} [g_i^*(p_i) + l_i^*(q_i) + \langle p_i, K_i r_i \rangle] \right\}. \tag{37}$$

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

$$\begin{aligned} & z - \sum_{i=1}^{m} L_i^* K_i^* \bar{p}_i \in \partial f(\bar{x}) + \nabla h(\bar{x}) \\ & \text{and } K_i(L_i \bar{x} - \bar{y}_i - r_i) \in \partial g_i^*(\bar{p}_i), M_i \bar{y}_i \in \partial l_i^*(\bar{q}_i), K_i^* \bar{p}_i = M_i^* \bar{q}_i, i = 1, \ldots, m, \end{aligned} \tag{38}$$

which means that $\bar{x}$ is an optimal solution to (36) and $(\overline{\boldsymbol{p}}, \overline{\boldsymbol{q}})$ is an optimal solution to (37).

For the primal-dual system (38), the iterative sequence proposed in (5) and (32) and the corresponding convergence statements are introduced as follows.

Algorithm 1: Primal-dual forward–backward splitting type algorithm for solving (36)

Let $x_0 \in \mathcal{H}$, and for any $i = 1, \ldots, m$, let $p_{i,0} \in \mathcal{X}_i, q_{i,0} \in \mathcal{Y}_i$ and $z_{i,0}, y_{i,0}, v_{i,0} \in \mathcal{G}_i$
Define

$$(\forall n \geq 0) \quad \left\lfloor \begin{array}{l} \widetilde{x}_n = \operatorname{prox}_{\tau f}\left(x_n - \tau \left(\nabla h(x_n) + \sum_{i=1}^{m} L_i^* v_{i,n} - z \right) \right) \\ \text{For } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} \widetilde{p}_{i,n} = \operatorname{prox}_{\theta_{1,i} g_i^*}(p_{i,n} + \theta_{1,i} K_i z_{i,n}) \\ \widetilde{q}_{i,n} = \operatorname{prox}_{\theta_{2,i} l_i^*}(q_{i,n} + \theta_{2,i} M_i y_{i,n}) \\ u_{1,i,n} = z_{i,n} + \gamma_{1,i}(K_i^*(p_{i,n} - 2\widetilde{p}_{i,n}) + v_{i,n} + \sigma_i(L_i(2\widetilde{x}_n - x_n) - r_i)) \\ u_{2,i,n} = y_{i,n} + \gamma_{2,i}(M_i^*(q_{i,n} - 2\widetilde{q}_{i,n}) + v_{i,n} + \sigma_i(L_i(2\widetilde{x}_n - x_n) - r_i)) \\ \tilde{z}_{i,n} = \dfrac{1 + \sigma_i \gamma_{2,i}}{1 + \sigma_i(\gamma_{1,i} + \gamma_{2,i})} \left(u_{1,i,n} - \dfrac{\sigma_i \gamma_{1,i}}{1 + \sigma_i \gamma_{2,i}} u_{2,i,n} \right) \\ \tilde{y}_{i,n} = \dfrac{1}{1 + \sigma_i \gamma_{2,i}}(u_{2,i,n} - \sigma_i \gamma_{2,i} \tilde{z}_{i,n}) \\ \widetilde{v}_{i,n} = v_{i,n} + \sigma_i(L_i(2\widetilde{x}_n - x_n) - r_i - \tilde{z}_{i,n} - \widetilde{y}_{i,n}) \end{array} \right. \\ x_{n+1} = x_n + \lambda_n(\widetilde{x}_n - x_n) \\ \text{For } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} p_{i,n+1} = p_{i,n} + \lambda_n(\widetilde{p}_{i,n} - p_{i,n}) \\ q_{i,n+1} = q_{i,n} + \lambda_n(\widetilde{q}_{i,n} - q_{i,n}) \\ z_{i,n+1} = z_{i,n} + \lambda_n(\tilde{z}_{i,n} - z_{i,n}) \\ y_{i,n+1} = y_{i,n} + \lambda_n(\widetilde{y}_{i,n} - y_{i,n}) \\ v_{i,n+1} = v_{i,n} + \lambda_n(\widetilde{v}_{i,n} - v_{i,n}) \end{array} \right. \end{array} \right. \tag{39}$$

The convergence of Algorithm 1 is presented in the following theorem.

Theorem 3. *For the convex optimization problem (36), suppose that*

$$z \in \operatorname{ran}\left(\partial f + \sum_{i=1}^{m} L_i^*((K_i^* \circ \partial g_i \circ K_i)\square(M_i^* \circ \partial l_i \circ M_i))(L_i \cdot - r_i) + \nabla h\right) \tag{40}$$

and consider the sequences generated by Algorithm 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 1. Then, there exists an optimal solution $\bar{x}$ to (36) and optimal solution $(\overline{\boldsymbol{p}}, \overline{\boldsymbol{q}})$ to (37) such that $x_n \rightharpoonup \bar{x}$ and for $i = 1, \ldots, m$, $p_{i,n} \rightharpoonup \bar{p}_i$, and $q_{i,n} \rightharpoonup \bar{q}_i$ as $n \to +\infty$.

Proof. In Theorem 1, let

$$A = \partial f, C = \nabla h, \text{ and } B_i = \partial g_i, D_i = \partial l_i, i = 1, \ldots, m. \tag{41}$$

According to Theorem 20.25 of [23], the operators in (41) 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 the Baillon-Haddad theorem, $C = \nabla h$ is μ^{-1}-cocoercive. By Theorem 1, we have $x_n \rightharpoonup \bar{x}$ and for $i = 1, \ldots, m$, $p_{i,n} \rightharpoonup \bar{p}_i$, and $q_{i,n} \rightharpoonup \bar{q}_i$. □

The second algorithm is obtained from (32).

Algorithm 2: Primal-dual forward–backward-half–forward splitting type algorithm for solving (36)

Let $\{\gamma_n\} \subseteq [\eta, \chi - \eta]$, where $\eta \in (0, \chi/2]$, l is defined by (16), and $\chi := \frac{4\mu^{-1}}{1+\sqrt{1+16(\mu^{-1})^2 l^2}}$. Let $x_0 \in \mathcal{H}$, and for any $i = 1, \ldots, m$, let $p_{i,0} \in \mathcal{X}_i, q_{i,0} \in \mathcal{Y}_i, z_{i,0} \in \mathcal{G}_i, y_{i,0} \in \mathcal{G}_i$ and $v_{i,0} \in \mathcal{G}_i$. Set

$$(\forall n \geq 0) \left\lfloor \begin{array}{l} \widetilde{x}_n = \operatorname{prox}_{\gamma_n f}(x_n - \gamma_n(\nabla h(x_n) + \sum_{i=1}^{m} L_i^* v_{i,n} - z)) \\ \text{For } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} \widetilde{p}_{i,n} = \operatorname{prox}_{\gamma_n g_i^*}(p_{i,n} + \gamma_n K_i z_{i,n}) \\ \widetilde{q}_{i,n} = \operatorname{prox}_{\gamma_n l_i^*}(q_{i,n} + \gamma_n M_i y_{i,n}) \\ u_{1,i,n} = z_{i,n} - \gamma_n(K_i^* p_{i,n} - v_{i,n} - \gamma_n(L_i x_n - r_i)) \\ u_{2,i,n} = y_{i,n} - \gamma_n(M_i^* q_{i,n} - v_{i,n} - \gamma_n(L_i x_n - r_i)) \\ \widetilde{z}_{i,n} = \dfrac{1+\gamma_n^2}{1+2\gamma_n^2}\left(u_{1,i,n} - \dfrac{\gamma_n^2}{1+\gamma_n^2} u_{2,i,n}\right) \\ \widetilde{y}_{i,n} = \dfrac{1+\gamma_n^2}{1+2\gamma_n^2}\left(u_{2,i,n} - \dfrac{\gamma_n^2}{1+\gamma_n^2} u_{1,i,n}\right) \\ \widetilde{v}_{i,n} = v_{i,n} + \gamma_n(L_i x_n - r_i - \widetilde{z}_{i,n} - \widetilde{y}_{i,n}) \end{array} \right. \\ x_{n+1} = \widetilde{x}_n + \gamma_n \sum_{i=1}^{m} L_i^*(v_{i,n} - \widetilde{v}_{i,n}) \\ \text{For } i = 1, \ldots, m \\ \left\lfloor \begin{array}{l} p_{i,n+1} = \widetilde{p}_{i,n} - \gamma_n K_i(z_{i,n} - \widetilde{z}_{i,n}) \\ q_{i,n+1} = \widetilde{q}_{i,n} - \gamma_n M_i(y_{i,n} - \widetilde{y}_{i,n}) \\ z_{i,n+1} = \widetilde{z}_{i,n} + \gamma_n K_i^*(p_{i,n} - \widetilde{p}_{i,n}) \\ y_{i,n+1} = \widetilde{y}_{i,n} + \gamma_n M_i(q_{i,n} - \widetilde{q}_{i,n}) \\ v_{i,n+1} = \widetilde{v}_{i,n} - \gamma_n L_i(x_n - \widetilde{x}_n). \end{array} \right. \end{array} \right. \tag{42}$$

As a direct result of Theorem 2, we have the following convergence theorem for (42). Since the proof is the same as Theorem 3, we omit it here.

Theorem 4. *For the convex optimization problem (36), suppose that*

$$z \in \operatorname{ran}\left(\partial f + \sum_{i=1}^{m} L_i^*((K_i^* \circ \partial g_i \circ K_i)\square(M_i^* \circ \partial l_i \circ M_i))(L_i \cdot - r_i) + \nabla h\right), \tag{43}$$

and consider the sequences generated by Algorithm 2. Then, there exists an optimal solution $\bar{x}$ to (36) and optimal solution $(\overline{\boldsymbol{p}}, \overline{\boldsymbol{q}})$ to (37) such that $x_n \rightharpoonup \bar{x}$ and for $i = 1, \dots, m$, $p_{i,n} \rightharpoonup \bar{p}_i$, and $q_{i,n} \rightharpoonup \bar{q}_i$ as $n \to +\infty$.

4. Numerical Experiments

In this section, we present some experimental results on image denoising problems under Gaussian noise. We compare the proposed algorithms with the FB_BH algorithm (5) and the FBF_BH algorithm (8). We call Algorithm 1 the FB algorithm. On the other hand, we refer to the proposed Algorithm 2 as the FBHF algorithm. All numerical experiments were implemented on Matlab R2016b on a Lenovo laptop with Intel i7-6700 CPU 3.40 GHz and 4 GB memory.

4.1. Image Denoising Problems

In this subsection, we show how the proposed algorithms could be applied to solve image denoising problems.

Let $b \in \mathbb{R}^n$ be the observed and vectorized noisy image of size $M \times N$ (with $n = MN$ for greyscale and $n = 3MN$ for colored images). Let $k \geq 1$, and define

$$D_k := \begin{bmatrix} -1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & & \vdots \\ 0 & \cdots & 0 & -1 & 1 & 0 \\ 0 & \cdots & 0 & 0 & -1 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 0 \end{bmatrix} \in \mathbb{R}^{k \times k}, \tag{44}$$

which models the discrete first-order derivative. We denote by $A \otimes B$ the Kronecker product of the matrices A and B and define

$$D_x = I_N \otimes D_M, D_y = D_N \otimes I_M \text{ and } \mathcal{D}_1 = \begin{bmatrix} D_x \\ D_y \end{bmatrix} \tag{45}$$

where D_x and D_y represent the vertical and horizontal difference operators, respectively, and I_N and I_M are the identity matrices of sizes N and M, respectively. Further, we define the discrete second-order derivatives matrices

$$D_{xx} = I_N \otimes \left(-D_M^T D_M\right), D_{yy} = \left(-D_N^T D_N\right) \otimes I_M, \mathcal{D}_2 = \begin{bmatrix} D_{xx} \\ D_{yy} \end{bmatrix}, \tag{46}$$

and

$$L_1 = \begin{bmatrix} -D_x^T & 0 \\ 0 & -D_y^T \end{bmatrix}. \tag{47}$$

We mainly consider the following two constrained image denoising models:

$$(\ell_2\text{-IC}) \quad \min_{x \in C} \left\{ \frac{1}{2}\|x - b\|^2 + ((\alpha_1 \|\cdot\|_1 \circ \mathcal{D}_1)\square(\alpha_2 \|\cdot\|_1 \circ \mathcal{D}_2))(x) \right\}, \tag{48}$$

and

$$(\ell_2\text{-MIC}) \quad \min_{x \in C} \left\{ \frac{1}{2}\|x - b\|^2 + ((\alpha_1\| \cdot \|_1)\Box(\alpha_2\| \cdot \|_1 \circ L_1))(\mathcal{D}_1 x) \right\}, \tag{49}$$

where $\alpha_1 > 0, \alpha_2 > 0$ are the regularization parameters, and C is a nonempty closed convex set. By using the indicator function, both of the constrained ℓ_2-IC and $\ell_2^2 -$ MIC can be reformulated as the following unconstrained optimization problem,

$$(\ell_2\text{-IC}) \quad \min_{x \in R^n} \left\{ \frac{1}{2}\|x - b\|^2 + ((\alpha_1\| \cdot \|_1 \circ \mathcal{D}_1)\Box(\alpha_2\| \cdot \|_1 \circ \mathcal{D}_2))(x) + \delta_C(x) \right\}, \tag{50}$$

and

$$(\ell_2\text{-MIC}) \quad \min_{x \in R^n} \left\{ \frac{1}{2}\|x - b\|^2 + ((\alpha_1\| \cdot \|_1)\Box(\alpha_2\| \cdot \|_1 \circ L_1))(\mathcal{D}_1 x) + \delta_C(x) \right\}. \tag{51}$$

It is easy to see that (50) and (51) are special cases of the general convex minimization problem (36), respectively. In fact, let $m = 1$, $z = 0$, and $r_1 = 0$. For the ℓ_2-IC, let $f(x) = \delta_C(x)$, $g_1(x) = \alpha_1\|x\|_1$, $l_1(x) = \alpha_2\|x\|_1$, $K_1 = \mathcal{D}_1$, $M_1 = \mathcal{D}_2$, $L_1 = I$, and $h(x) = \frac{1}{2}\|x - b\|^2$; for the ℓ_2-MIC, let $f(x) = \delta_C(x)$, $g_1(x) = \alpha_1\|x\|_1$, $l_1(x) = \alpha_2\|x\|_1$, $K_1 = I$, $M_1 = L_1$, $L_1 = \mathcal{D}_1$, and $h(x) = \frac{1}{2}\|x - b\|^2$.

4.2. Numerical Settings

The test images are shown in Figure 1. In our experiments, the test image is added by Gaussian noise with zero mean and standard deviation σ_g. In the following experiment, we set $C = \{x \in R^n | 0 \leq x_i \leq 255\}$.

(**a**)

(**b**)

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

We use the peak-signal-to-noise (PSNR) and the structural similarity index (SSIM) [24] to evaluate the quality of the restored images, which are defined by

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

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 $x \in R^n$ is the original image, $\tilde{x} \in R^n$ is the restored image, $c_1 > 0$ and $c_2 > 0$ are small constants, μ_1 and μ_2 are the mean values of x and $\tilde{x}$, respectively; σ_1 and σ_2 are the variances of x and $\tilde{x}$, respectively; and σ_{12} is the covariance of x and $\tilde{x}$.

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 small constant.

We tune the regularization parameters α_1 and α_2 so as to maximize the PSNR values of the restored images. The choices of α_1 and α_2 are presented in Table 1.

Table 1. The regularization parameters selection of the ℓ_2-IC and ℓ_2-MIC.

Image	**Model**	$\sigma = 15$		$\sigma = 25$		$\sigma = 50$	
		α_1	α_2	α_1	α_2	α_1	α_2
Castle	ℓ_2-IC	7.7	21.2	14.7	29.7	35.5	123.9
	ℓ_2-MIC	7.6	21.1	14.8	50.8	35.7	115.9
Building	ℓ_2-IC	6.1	25.3	12.4	33	31.3	87.8
	ℓ_2-MIC	6.1	27.8	12.5	49.4	31.8	140

4.3. Numerical Results and Discussion

In the first experiment, we discuss the influence of the selection of iterative parameters on the convergence speed of the compared algorithms. According to the convergence theorems, the parameter selection of these algorithms is shown in Table 2. 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 $\mu = 1$.

Table 2. The parameter selection of the compared algorithms.

Model	**Method**	**Parameter**
ℓ_2-IC	FB_BH	$\lambda_n \in (0,1], 2(1-\bar{\alpha})\min\left\{\frac{1}{\tau}, \frac{1}{\theta_1}, \frac{1}{\theta_2}, \frac{1}{\gamma_1}, \frac{1}{\gamma_2}, \frac{1}{\sigma}\right\} > 1,$ $\bar{\alpha} = \max\left\{\sqrt{\tau\sigma}, 2.8072\sqrt{\theta_1\gamma_1}, 5.6133\sqrt{\theta_2\gamma_2}\right\}$
	FBF_BH	$\gamma_n \in (0, 0.1512)$
	FB	$\lambda_n \in (0, 2-\frac{1}{2\beta}), 2\beta > 1, (1-\bar{\alpha})\min\left\{\frac{1}{\tau}, \frac{1}{\theta_1}, \frac{1}{\theta_2}, \frac{1}{\gamma_1}, \frac{1}{\gamma_2}, \frac{1}{\sigma}\right\} > 0,$ $\bar{\alpha} = \max\left\{\sqrt{\tau\sigma}, 2.8072\sqrt{\theta_1\gamma_1}, 5.6133\sqrt{\theta_2\gamma_2}\right\}, \beta = \frac{1}{\tau} - \sigma$
	FBHF	$\gamma_n \in (0, 0.1704)$
ℓ_2-MIC	FB_BH	$\lambda_n \in (0,1], 2(1-\bar{\alpha})\min\left\{\frac{1}{\tau}, \frac{1}{\theta_1}, \frac{1}{\theta_2}, \frac{1}{\gamma_1}, \frac{1}{\gamma_2}, \frac{1}{\sigma}\right\} > 1,$ $\bar{\alpha} = \max\left\{2.8072\sqrt{\tau\sigma}, \sqrt{\theta_1\gamma_1}, 1.9926\sqrt{\theta_2\gamma_2}\right\}$
	FBF_BH	$\gamma_n \in (0, 0.2627)$
	FB	$\lambda_n \in (0, 2-\frac{1}{2\beta}), 2\beta > 1, (1-\bar{\alpha})\min\left\{\frac{1}{\tau}, \frac{1}{\theta_1}, \frac{1}{\theta_2}, \frac{1}{\gamma_1}, \frac{1}{\gamma_2}, \frac{1}{\sigma}\right\} > 0,$ $\bar{\alpha} = \max\left\{2.8072\sqrt{\tau\sigma}, \sqrt{\theta_1\gamma_1}, 1.9926\sqrt{\theta_2\gamma_2}\right\}, \beta = \frac{1}{\tau} - 7.8798\sigma$
	FBHF	$\gamma_n \in (0, 0.3259)$

We chose Castle in Figure 1 as the test image, and the Gaussian noise level $\sigma_g = 15$. The numerical results of the FBF_BH algorithm and the FBHF algorithm with a different selections of the parameters $\{\gamma_n\}$ are reported in Table 3. It can be seen from Table 3 that both the FBF_BH algorithm and the FBHF algorithm gradually reduced the number of iterations as the step size parameter increased.

For the FB_BH algorithm and the FB algorithm, we selected several combinations of the parameters in Table 4.

According to the iteration parameters in Table 4, the obtained numerical results are shown in Table 5.

Table 3. Numerical results of the FBF_BH algorithm and the FBHF algorithm with different selections of parameters in terms of the PSNR, SSIM, and number of iterations (Iter).

Method	Model	γ_n	$\varepsilon = 10^{-5}$			$\varepsilon = 10^{-6}$		
			PSNR	SSIM	Iter	PSNR	SSIM	Iter
FBF_BH	ℓ_2-IC	0.03	30.5138	0.8410	1791	30.5346	0.8409	6572
		0.05	30.5225	0.8410	1411	30.5356	0.8409	5203
		0.07	30.5287	0.8410	1227	30.5361	0.8409	4445
		0.09	30.5309	0.8410	1103	30.5370	0.8409	3994
		0.11	30.5317	0.8410	1003	30.5379	0.8409	3730
		0.13	30.5323	0.8410	946	30.5386	0.8409	3560
		0.15	30.5330	0.8410	888	30.5391	0.8409	3372
	ℓ_2-MIC	0.03	30.5330	0.8384	2287	30.5434	0.8391	4984
		0.07	30.5358	0.8386	1126	30.5451	0.8392	2824
		0.11	30.5378	0.8387	796	30.5460	0.8393	2207
		0.15	30.5390	0.8388	634	30.5365	0.8393	1896
		0.19	30.5400	0.8388	538	30.5467	0.8393	1686
		0.23	30.5409	0.8389	474	30.5468	0.8393	1518
		0.26	30.5414	0.8389	439	30.5470	0.8393	1456
FBHF	ℓ_2-IC	0.03	30.5138	0.8410	1790	30.5346	0.8409	6474
		0.05	30.5225	0.8410	1411	30.5356	0.8409	5203
		0.07	30.5287	0.8410	1227	30.5361	0.8409	4449
		0.09	30.5309	0.8410	1103	30.5370	0.8409	3994
		0.11	30.5317	0.8410	1003	30.5379	0.8409	3729
		0.13	30.5323	0.8410	946	30.5386	0.8409	3561
		0.15	30.5330	0.8410	887	30.5391	0.8409	3371
		0.17	30.5336	0.8410	837	30.5393	0.8409	3169
	ℓ_2-MIC	0.03	30.5330	0.8384	2286	30.5434	0.8391	4983
		0.07	30.5358	0.8386	1126	30.5452	0.8392	2824
		0.11	30.5378	0.8387	796	30.5460	0.8393	2207
		0.15	30.5390	0.8388	633	30.5465	0.8393	1896
		0.19	30.5401	0.8388	538	30.5467	0.8393	1686
		0.23	30.5410	0.8389	474	30.5468	0.8393	1519
		0.27	30.5416	0.8389	428	30.5470	0.8393	1432
		0.31	30.5421	0.8390	394	30.5471	0.8394	1346
		0.32	30.5422	0.8390	387	30.5471	0.8394	1317

According to the results of Tables 3 and 5, we chose the following parameters of the compared algorithms for the following experiments.

(1) For the FBF_BH algorithm, the best parameter of ℓ_2-IC was $\gamma_n = 0.15$, and the best parameter of ℓ_2-MIC was $\gamma_n = 0.26$.

(2) For the FBHF algorithm, the best parameter of ℓ_2-IC was $\gamma_n = 0.17$, and the best parameter of ℓ_2-MIC was $\gamma_n = 0.32$.

(3) For the FB_BH algorithm, the best parameters of ℓ_2-IC were $\theta_1 = 0.3$, $\gamma_1 = 0.3, \theta_2 = 0.15, \gamma_2 = 0.15, \lambda_n = 1, \tau = 0.3$, and $\sigma = 0.3$, and the best parameters of ℓ_2-MIC were $\theta_1 = 0.4, \gamma_1 = 0.3, \theta_2 = 0.2, \gamma_2 = 0.2, \lambda_n = 1, \tau = 0.2$, and $\sigma = 0.3$.

(4) For the FB algorithm, the best parameters of ℓ_2-IC were $\theta_1 = 0.3, \gamma_1 = 0.3$, $\theta_2 = 0.2, \gamma_2 = 0.1, \lambda_n = 1.8, \tau = 0.2$, and $\sigma = 0.2$, and the best parameters of ℓ_2-MIC were $\theta_1 = 0.3, \gamma_1 = 0.3, \theta_2 = 0.2, \gamma_2 = 0.2, \lambda_n = 1.8, \tau = 0.2$, and $\sigma = 0.2$.

In the second experiment, we tested the performance of the compared algorithms for solving ℓ_2-IC and ℓ_2-MIC. We present the numerical results by each algorithm in Table 6.

Table 4. Parameter selection of the FB_BH algorithm and the FB algorithm.

Method	Model	Case	θ_1	θ_2	γ_1	γ_2	τ	σ	λ_n
FB_BH	ℓ_2-IC	1	0.3	0.15	0.3	0.15	0.3	0.3	1
		2	0.2	0.15	0.2	0.15	0.3	0.3	1
		3	0.2	0.1	0.2	0.1	0.2	0.2	1
		4	0.2	0.2	0.2	0.1	0.1	0.3	1
	ℓ_2-MIC	1	0.3	0.15	0.3	0.15	0.3	0.3	1
		2	0.4	0.2	0.3	0.1	0.2	0.3	1
		3	0.1	0.2	0.1	0.2	0.3	0.2	1
		4	0.1	0.1	0.1	0.1	0.2	0.4	1
FB	ℓ_2-IC	1	0.3	0.15	0.3	0.15	0.3	0.3	1.4
		2	0.2	0.1	0.3	0.2	0.3	0.4	1.5
		3	0.3	0.2	0.3	0.1	0.2	0.2	1.8
		4	0.2	0.2	0.2	0.1	0.1	0.3	1.3
	ℓ_2-MIC	1	0.5	0.4	0.5	0.4	0.3	0.3	1.4
		2	0.4	0.3	0.4	0.4	0.25	0.25	1.5
		3	0.3	0.2	0.3	0.2	0.2	0.2	1.8
		4	0.2	0.3	0.2	0.3	0.3	0.3	1.4

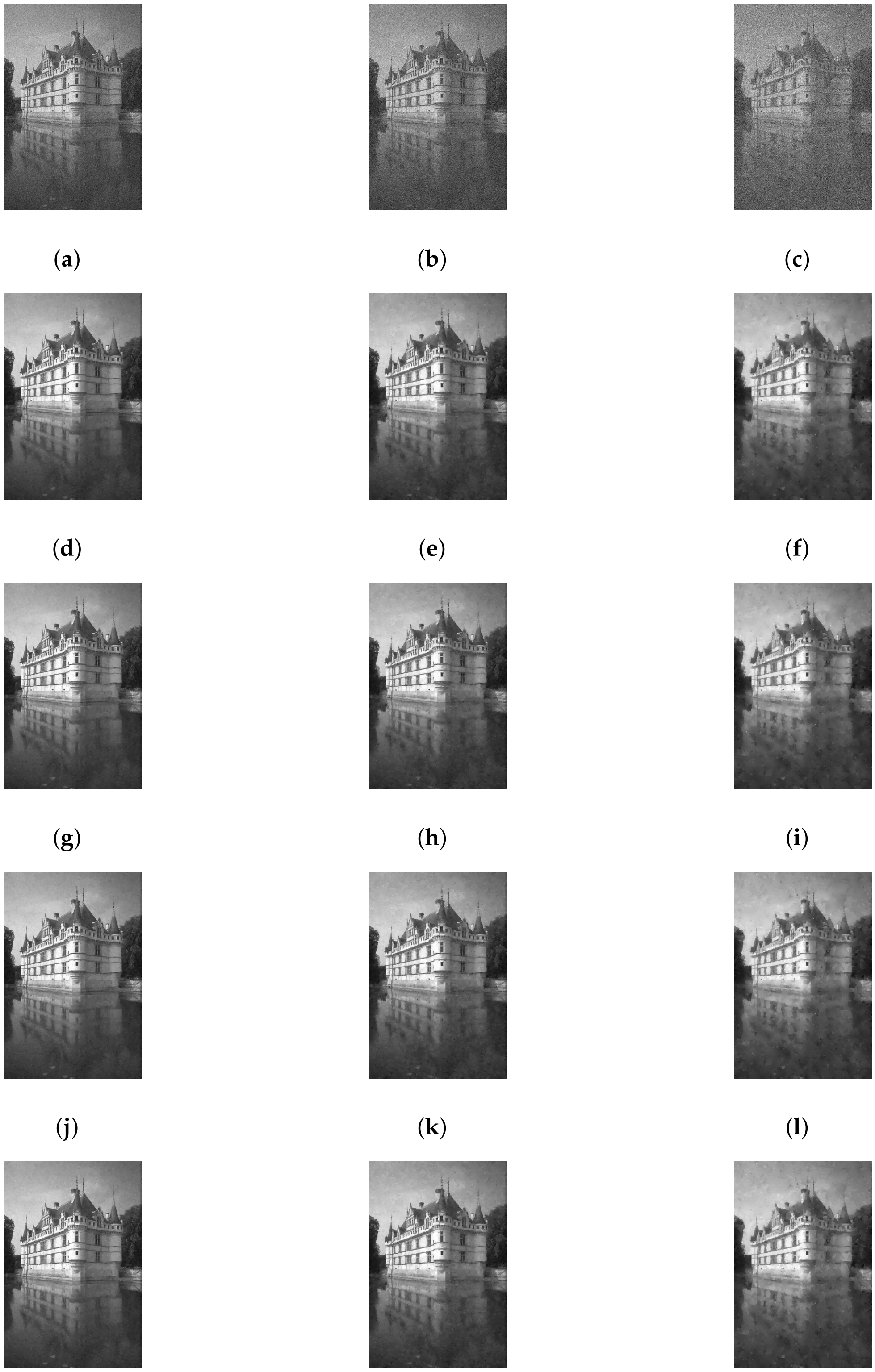
Table 5. Numerical results of the FB_BH algorithm and the FB algorithm with different parameters in terms of the PSNR, SSIM, and number of iterations (Iter).

Method	Model	Case	$\varepsilon = 10^{-5}$			$\varepsilon = 10^{-6}$		
			PSNR	SSIM	Iter	PSNR	SSIM	Iter
FB_BH	ℓ_2-IC	1	30.5386	0.8411	753	30.5405	0.8409	2846
		2	30.5361	0.8411	824	30.5399	0.8409	3117
		3	30.5379	0.8411	885	30.5397	0.8409	3277
		4	30.5369	0.8409	818	30.5395	0.8408	3087
	ℓ_2-MIC	1	30.5386	0.8411	657	30.5411	0.8409	2481
		2	30.5412	0.8389	534	30.5468	0.8393	1632
		3	30.5253	0.8408	930	30.5364	0.8409	3603
		4	30.5313	0.8410	1034	30.5375	0.8309	3875
FB	ℓ_2-IC	1	30.5387	0.8411	701	30.5408	0.8409	2640
		2	30.5382	0.8412	754	30.5419	0.8410	2514
		3	30.5389	0.8409	601	30.5407	0.8409	2242
		4	30.5377	0.8409	735	30.5398	0.8408	2725
	ℓ_2-MIC	1	30.5442	0.8391	292	30.5475	0.8394	1047
		2	30.5435	0.8391	321	30.5474	0.8394	1140
		3	30.5447	0.8392	285	30.5476	0.8394	980
		4	30.5435	0.8391	372	30.5474	0.8394	1121

Table 6. Numerical results of the compared algorithms in terms of the PSNR, SSIM, and number of iterations (Iter).

Image	Model	σ_g	FBF_BH			FBHF		
			PSNR	**SSIM**	**Iter**	**PSNR**	**SSIM**	**Iter**
Castle	ℓ_2-IC	15	30.5330	0.8410	888	30.5336	0.8410	837
		25	27.9370	0.7799	786	27.9376	0.7798	736
		50	24.9407	0.7027	1364	24.9427	0.7026	1315
	ℓ_2-MIC	15	30.5414	0.8389	439	30.5422	0.8390	387
		25	27.9352	0.7796	665	27.9379	0.7798	615
		50	24.9332	0.7010	1169	24.9380	0.7014	1093
Building	ℓ_2-IC	15	28.3617	0.8404	1365	28.3612	0.8404	1339
		25	25.5939	0.7333	1020	25.5943	0.7333	971
		50	22.6506	0.5558	1100	22.6510	0.5558	1045
	ℓ_2-MIC	15	28.3663	0.8405	492	28.3665	0.8405	431
		25	25.5997	0.7332	623	25.6001	0.7332	570
		50	22.6716	0.5565	1146	22.6724	0.5565	1070
Image	**Model**	σ_g	**FB_BH**			**FB**		
			PSNR	**SSIM**	**Iter**	**PSNR**	**SSIM**	**Iter**
Castle	ℓ_2-IC	15	30.5386	0.8411	753	30.5389	0.8409	601
		25	27.9398	0.7799	691	27.9452	0.7797	548
		50	24.9415	0.7027	1321	24.9485	0.7013	1039
	ℓ_2-MIC	15	30.5412	0.8389	534	30.5426	0.8391	398
		25	27.9330	0.7795	734	27.9400	0.7799	601
		50	24.9278	0.7707	1243	24.9418	0.7017	1085
Building	ℓ_2-IC	15	28.3635	0.8405	1047	28.3634	0.8404	906
		25	25.5952	0.7333	832	25.5956	0.7334	719
		50	22.6051	0.5563	1018	22.6521	0.5564	849
	ℓ_2-MIC	15	28.3664	0.8405	623	28.3668	0.8405	420
		25	25.5822	0.7329	584	25.6005	0.7322	554
		50	22.6629	0.5577	1043	22.6653	0.5578	856

From the experimental results of Table 6, we can see that the proposed FBHF algorithm converged faster than the FBF_BH algorithm in terms of the number of iterations while ensuring higher PSNR and SSIM values. Meanwhile, the proposed FB algorithm also converged faster than the FB_BH algorithm. The obtained results verify that the proposed algorithms are better than those in [4]. Some of the recovered images are shown in Figure 2 and Figure 3, respectively.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**) (**i**) (**j**) (**k**) (**l**) (**m**) (**n**) (**o**)

Figure 2. Noisy and restored "Castle" images. (**a**) $\sigma_g = 15$. (**b**) $\sigma_g = 25$. (**c**) $\sigma_g = 50$. (**d**) ℓ_2-IC/FBF_BH. (**e**) ℓ_2-IC/FBF_BH. (**f**) ℓ_2-IC/FBF_BH. (**g**) ℓ_2-IC/FB_BH. (**h**) ℓ_2-IC/FB_BH. (**i**) ℓ_2-IC/FB_BH. (**j**) ℓ_2-IC/FBHF. (**k**) ℓ_2-IC/FBHF. (**l**) ℓ_2-IC/FBHF. (**m**) ℓ_2-IC/FB. (**n**) ℓ_2-IC/FB. (**o**) ℓ_2-IC/FB.

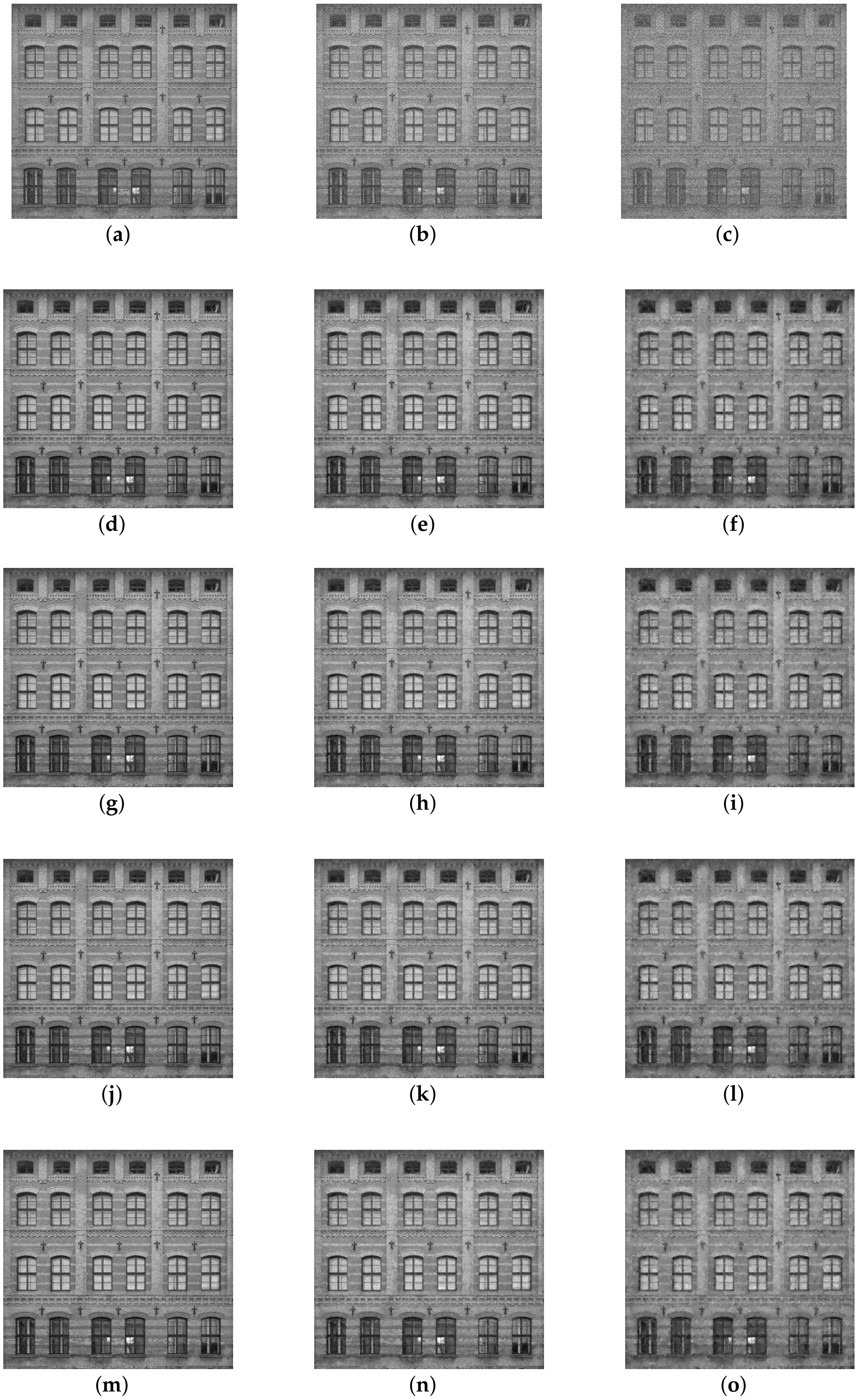


Figure 3. Noisy and restored "Building" images. (**a**) $\sigma_g = 15$. (**b**) $\sigma_g = 25$. (**c**) $\sigma_g = 50$. (**d**) ℓ_2-IC/FBF_BH. (**e**) ℓ_2-IC/FBF_BH. (**f**) ℓ_2-IC/FBF_BH. (**g**) ℓ_2-IC/FB_BH. (**h**) ℓ_2-IC/FB_BH. (**i**) ℓ_2-IC/FB_BH. (**j**) ℓ_2-IC/FBHF. (**k**) ℓ_2-IC/FBHF. (**l**) ℓ_2-IC/FBHF. (**m**) ℓ_2-IC/FB. (**n**) ℓ_2-IC/FB. (**o**) ℓ_2-IC/FB.

5. Conclusions

In this paper, we studied the convergence of two different primal-dual splitting algorithms for solving monotone inclusions (3) and (4). Firstly, we proved the convergence of the forward–backward type algorithm (5). Our parameter conditions improved the results of Boţ and Hendrich [4]. Secondly, we proposed a new forward–backward–half-forward type algorithm (32). In contrast to the forward–backward–forward type algorithm (8), the iterative sequences in the proposed forward–backward–half-forward type algorithm (32) used the cocoercive operator only once via the forward step. Finally, we applied the proposed algorithms to solve image denoising problems (48) and (49). The numerical results demonstrated the advantages of the proposed algorithms.

Author Contributions: Formal analysis, Q.D.; Investigation, Q.D.; Methodology, Y.T.; Supervision, Y.T.; Writing—original draft, X.L.; Writing—review & editing, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Natural Science Foundations of China (12061045, 11661056).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Acknowledgments: The authors would like to thank the referees and the editor for their helpful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vũ, B.C. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.* **2013**, *38*, 667–681. [CrossRef]
2. Pesquet, J.-C.; Repetti, A. A class of randomized primal-dual algorithms for distributed optimization. *J. Nonlinear Convex Anal.* **2015**, *16*, 2453–2490.
3. Vũ, B.C. A spliting algorithm for coupled system of primal-dual monotone inclusions. *J. Optim. Theory Appl.* **2015**, *164*, 993–1025. [CrossRef]
4. Boţ, R.I.; Hendrich, C. Solving monotone inclusions involving parallel sums of linearly composed maximally monotone operators. *Inverse Probl. Imaging* **2016**, *10*, 617–640. [CrossRef]
5. Boţ, R.I.; Hendrich, C. A douglas-rachford type primal-dual method for solving inclusions with mixtures of composite and parallel-sum type monotone operators. *SIAM J. Optim.* **2013**, *4*, 2541–2565. [CrossRef]
6. Boţ, R.I.; Csetnek, E.R.; Heinrich, A. A primal-dual splitting for finding zeros of sums of maximal monotone operators. *SIAM J. Optim.* **2013**, *23*, 2011–2036. [CrossRef]
7. Yang, Y.X.; Tang, Y.C.; Wen, M.; Zeng, T.Y. Preconditioned douglas-rachford type primal-dual method for solving composite monotone inclusion problems with applications. *Inverse Probl. Imaging* **2021**, *15*, 787–825. [CrossRef]
8. Briceño-Arias, L.M.; Combettes, P.L. A monotone+skew splitting splitting model for composite monotone inclusions in duality. *SIAM J. Control Optim.* **2011**, *21*, 1230–1250. [CrossRef]
9. Combettes, P.L.; Pesquet, J.-C. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, lipschitzian, and paralle-sum type monotone operators. *Set-Valued Var. Anal.* **2012**, *20*, 307–330. [CrossRef]
10. Combettes, P.L. Systems of structured monotone inclusions: Duality, algorithms, and applications. *SIAM J. Optim.* **2013**, *23*, 2420–2447. [CrossRef]
11. Becker, S.R.; Combettes, P.L. An algorithm for splitting parallel sums of linearly composed monotone operatos with applications to signal recovery. *J. Nonlinear Convex Anal.* **2014**, *15*, 137–159.
12. Boţ, R.I.; Hendrich, C. Convergence analysis for a primal-dual monotone+skew splitting algorithm with applications to total variation minimization. *J. Math. Imaging Vis.* **2014**, *49*, 551–568.
13. Alotaibi, A.; Combettes, P.L.; Shahzad, N. Solving coupled composite monotone inclusions by successive fejér approximations of their kukn-tucker set. *SIAM J. Optim.* **2014**, *24*, 2076–2095. [CrossRef]
14. Tran-Dinh, Q.; Vu, B.C. A new splitting method for solving composite monotone inclusions involving parallel-sum operators. *arXiv* **2015**, arXiv:1505.07946.
15. Combettes, P.L.; Eckstein, J. Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions. *Math. Program.* **2018**, *168*, 645–672. [CrossRef]
16. Bardaro, C.; Bevignani, G.; Mantellini, I.; Seracini, M. Bivariate generalized exponential sampling series and applications to seismic waves. *Constr. Math. Anal.* **2019**, *2*, 153–167. [CrossRef]

17. Johnstone, P.R.; Eckstein, J. Projective splitting with forward steps. *Math. Program.* **2020**, 1–40. [CrossRef]
18. Chambolle, A.; Lions, P.L. Image recovery via total variation minimizaing and related problems. *Numer. Math.* **1997**, *76*, 167–188. [CrossRef]
19. Setzer, S.; Steidl, G.; Teuber, T. Infimal convolution regularization with discrete l1-type functionals. *Commun. Math. Sci.* **2011**, *9*, 797–827. [CrossRef]
20. Combettes, P.L.; Vũ, B.C. Variable metric forward-backward splitting with applications to monotone inclusions in duality. *Optimization* **2014**, *63*, 1289–1318. [CrossRef]
21. Tseng, P. A modified forward-backward splitting method for maximal monotone mappings. *SIAM J. Control Optim.* **2000**, *38*, 431–446. [CrossRef]
22. Briceño-Arias, L.M.; Davis, D. Forward-backward-half forward algorithm for solving monotone inclusions. *SIAM J. Optim.* **2018**, *28*, 2839–2871. [CrossRef]
23. Bauschke, H.H.; Combettes, P.L. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed.; Springer: London, UK, 2017.
24. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]