

Forward–Partial Inverse–Forward Splitting for Solving Monotone Inclusions

Luis M. Briceño-Arias

Communicated by Viorel Barbu

Received: date / Accepted: date

Abstract In this paper, we provide a splitting method for finding a zero of the sum of a maximally monotone operator, a Lipschitzian monotone operator, and a normal cone to a closed vector subspace of a real Hilbert space. The problem is characterised by a simpler monotone inclusion involving only two operators: the partial inverse of the maximally monotone operator with respect to the vector subspace, and a suitable Lipschitzian monotone operator. By applying the Tseng’s method in this context, we obtain a fully split algorithm, that exploits the whole structure of the original problem and generalises partial inverse and Tseng’s methods. Connections with other methods available in the literature are provided, and the flexibility of our setting is illustrated via applications to some inclusions involving m maximally monotone operators, to primal-dual composite monotone inclusions, and to zero-sum games.

Keywords composite operator · partial inverse · monotone operator theory · splitting algorithms · Tseng’s method

Mathematics Subject Classification (2000) MSC 47H05 · MSC 47J25 · MSC 65K05 · MSC 90C25

Luis M. Briceño-Arias
Universidad Técnica Federico Santa María
Departamento de Matemática
Av. Vicuña Mackenna 3939
San Joaquín, Santiago, Chile
Tel.: +56 2 2432 6662
E-mail: luis.briceno@usm.cl

1 Introduction

This paper is concerned with the numerical resolution of the problem of finding a zero of the sum of a set-valued, maximally monotone operator, a Lipschitzian monotone operator, and a normal cone to a closed vector subspace of a real Hilbert space. This problem arises in a wide range of areas such as optimisation [1,2], variational inequalities [3–5], monotone operator theory [6–9], partial differential equations [3,10,11], economics [12,13], signal and image processing [14–16], evolution inclusions [17,18], traffic theory [19,20], and game theory [21], among others.

When the single-valued operator is zero, the problem is solved in [9] via the method of partial inverses. On the other hand, when the vector subspace is the whole Hilbert space, the normal cone is zero and our problem is reduced to finding a zero of the sum of two monotone operators. In this case, the problem is solved in [22] via the forward-backward-forward splitting or Tseng’s method (see also [23] and the references therein). In addition, in the case when the single-valued operator is cocoercive, the problem is solved in [24].

In the general case, several algorithms are available in the literature for solving our problem, but any of them exploits its intrinsic structure. The Tseng’s method [22] can be applied to the general case, but it needs to compute the resolvent of the sum of the set-valued operator and the normal cone, which is not always easy to implement. It is preferable to activate both operators separately. Some ergodic approaches for solving this problem can be found in [25]. A disadvantage of these methods is the presence of vanishing parameters, which usually lead to numerical instabilities. The algorithms proposed in [9,23,26] permit us to find a zero of the sum of finitely many maximally monotone operators by activating them independently, without considering vanishing parameters. However, these methods need the computation of the resolvent of the single-valued operator, which is not easy to compute in general. An algorithm proposed in [27] overcomes this difficulty by explicitly activating the single-valued operator. However, this method does not take advantage of the vector subspace involved. Indeed, by using product space techniques, the method in [27] needs to store additional auxiliary variables at each iteration, which can be difficult in high-dimensional problems.

In this paper, we propose a fully split method for finding a zero of the sum of the three monotone operators detailed before, by exploiting each of their intrinsic properties. The algorithm computes, at

each iteration, explicit steps on the single-valued operator and the resolvent of the partial inverse of the set-valued operator with respect to the closed vector subspace [9]. This resolvent has an explicit expression in several cases and it reduces to a Douglas-Rachford step [7, 28] in a particular instance. In this case, our method can be perceived as a forward-Douglas-Rachford-forward splitting, which generalises partial inverse and Tseng’s methods when the single-valued operator is zero and the vector subspace is the whole Hilbert space, respectively. We also provide connections with other methods in the literature and we illustrate the flexibility of our framework via some applications to inclusions involving m maximally monotone operators, to primal-dual composite monotone inclusions, and to zero-sum games. In the application to primal-dual inclusions, we introduce a new operation between set-valued operators, called *partial sum* with respect to a closed vector subspace, which preserves monotonicity and takes a central role in the problem and algorithm. On the other hand, in continuous zero-sum games, we provide an interesting splitting algorithm for calculating a Nash equilibrium, that avoids the computation of the projection onto mixed strategy spaces in infinite dimensions by performing simpler projections alternately. These applications enlighten the flexibility and usefulness of the vector space setting, which appears naturally in a different form in each instance.

The paper is organised as follows. In Section 2, we provide the notation and some preliminaries. We also obtain a relaxed version of Tseng’s method [22], which is interesting in its own right. In Section 3, a characterisation of our problem in terms of two appropriate monotone operators is given and a method for solving this problem is derived from the relaxed version of Tseng’s algorithm. Moreover, we provide connections with other methods in the literature. Finally, in Section 4, we apply our method to the problem of finding a zero of a sum of m maximally monotone operators and a Lipschitzian monotone operator, to a primal-dual composite monotone inclusion, and to continuous zero-sum games. The methods derived in each instance generalise and improve available algorithms in the literature.

2 Notation and Preliminaries

Throughout this paper, \mathcal{H} is a real Hilbert space with scalar product denoted by $\langle \cdot | \cdot \rangle$ and associated norm $\| \cdot \|$. The symbols \rightharpoonup and \rightarrow denote, respectively, weak and strong convergence and Id denotes the identity operator. The indicator function of a subset C of \mathcal{H} is ι_C , which takes the value 0 in C

and $+\infty$ in $\mathcal{H} \setminus C$. If C is non-empty, closed, and convex, then the projection of x onto C , denoted by $P_C x$, is the unique point in $\operatorname{Argmin}_{y \in C} \|x - y\|$, and the normal cone to C is the maximally monotone operator

$$N_C: \mathcal{H} \rightrightarrows \mathcal{H}: x \mapsto \begin{cases} \{u \in \mathcal{H} : (\forall y \in C) \langle y - x, u \rangle \leq 0\}, & \text{if } x \in C; \\ \emptyset, & \text{otherwise.} \end{cases} \quad (1)$$

An operator $T: \mathcal{H} \rightarrow \mathcal{H}$ is β -cocoercive for some $\beta \in]0, +\infty[$ iff, for every $x \in \mathcal{H}$ and $y \in \mathcal{H}$, $\langle x - y, Tx - Ty \rangle \geq \beta \|Tx - Ty\|^2$, it is χ -Lipschitzian iff, for every $x, y \in \mathcal{H}$, $\|Tx - Ty\| \leq \chi \|x - y\|$, it is non-expansive iff it is 1-Lipschitzian, and the set of fixed points of T is given by $\operatorname{Fix} T$.

We denote by $\operatorname{gra} A = \{(x, u) \in \mathcal{H} \times \mathcal{H} : u \in Ax\}$ the graph of $A: \mathcal{H} \rightrightarrows \mathcal{H}$, by $J_A = (\operatorname{Id} + A)^{-1}$ its resolvent, by $\operatorname{dom} A = \{x \in \mathcal{H} : Ax \neq \emptyset\}$ its domain, and by $\operatorname{zer} A = \{x \in \mathcal{H} : 0 \in Ax\}$ its set of zeros. If A is monotone, i.e., for every (x, u) and (y, v) in $\operatorname{gra} A$, $\langle x - y, u - v \rangle \geq 0$, then J_A is a single-valued, non-expansive operator. In addition, A is maximally monotone iff $\operatorname{dom} J_A = \mathcal{H}$. Let $A: \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone. The reflection operator of A is $R_A = 2J_A - \operatorname{Id}$, which is non-expansive. The partial inverse of A with respect to a vector subspace V of \mathcal{H} , denoted by A_V , is defined by

$$(\forall (x, y) \in \mathcal{H}^2) \quad y \in A_V x \quad \Leftrightarrow \quad (P_V y + P_{V^\perp} x) \in A(P_V x + P_{V^\perp} y). \quad (2)$$

Note that $A_{\mathcal{H}} = A$ and $A_{\{0\}} = A^{-1}$. The following properties of the partial inverse will be useful throughout this paper.

Proposition 2.1 *Let $A: \mathcal{H} \rightrightarrows \mathcal{H}$ be a set-valued operator and let V be a closed vector subspace of \mathcal{H} . Then, $(A_V)^{-1} = (A^{-1})_V = A_{V^\perp}$ and $P_V(A + N_V)^{-1}P_V = P_V(A_{V^\perp} + N_V)P_V$.*

Proof. Let $(x, u) \in \mathcal{H}^2$. We have from (2) that

$$\begin{aligned} u \in (A_V)^{-1}x &\Leftrightarrow x \in A_V u \\ &\Leftrightarrow P_V x + P_{V^\perp} u \in A(P_V u + P_{V^\perp} x) \end{aligned} \quad (3)$$

$$\begin{aligned} &\Leftrightarrow P_V u + P_{V^\perp} x \in A^{-1}(P_V x + P_{V^\perp} u) \\ &\Leftrightarrow u \in (A^{-1})_V x. \end{aligned} \quad (4)$$

On the other hand, it follows from (3) and (2) that $u \in (A_V)^{-1}x$ is equivalent to $u \in A_{V^\perp}x$. For the second identity, we deduce from (2) that

$$\begin{aligned}
u \in P_V(A + N_V)^{-1}(P_V x) &\Leftrightarrow (u \in V) \quad u \in (A + N_V)^{-1}(P_V x) \\
&\Leftrightarrow (u \in V) \quad P_V x \in Au + N_V u \\
&\Leftrightarrow (u \in V)(\exists y \in V^\perp) \quad P_V x - y \in Au \\
&\Leftrightarrow (u \in V)(\exists y \in V^\perp) \quad u - y \in A_{V^\perp}(P_V x) \\
&\Leftrightarrow (u \in V) \quad u \in (A_{V^\perp} + N_V)(P_V x) \\
&\Leftrightarrow u \in P_V(A_{V^\perp} + N_V)(P_V x),
\end{aligned}$$

which yields the result. \square

The following result is a relaxed version of the methods proposed in [22, 23, 29].

Proposition 2.2 *Let $\eta \in]0, +\infty[$, let $\mathcal{A}: \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone, and let $\mathcal{B}: \mathcal{H} \rightarrow \mathcal{H}$ be monotone and η -Lipschitzian such that $\text{zer}(\mathcal{A} + \mathcal{B}) \neq \emptyset$. Moreover, let $z_0 \in \mathcal{H}$, let $\varepsilon \in]0, \max\{1, 1/2\eta\}[$, let $(\delta_n)_{n \in \mathbb{N}}$ be a sequence in $[\varepsilon, (1/\eta) - \varepsilon]$, let $(\lambda_n)_{n \in \mathbb{N}}$ be a sequence in $[\varepsilon, 1]$, and iterate*

$$(\forall n \in \mathbb{N}) \quad \begin{cases} r_n := z_n - \delta_n \mathcal{B} z_n \\ s_n := J_{\delta_n \mathcal{A}} r_n \\ t_n := s_n - \delta_n \mathcal{B} s_n \\ z_{n+1} := z_n + \lambda_n (t_n - r_n). \end{cases} \quad (5)$$

Then, $z_n \rightarrow \bar{z}$ for some $\bar{z} \in \text{zer}(\mathcal{A} + \mathcal{B})$ and $z_{n+1} - z_n \rightarrow 0$.

Proof. First note that (5) yields

$$(\forall n \in \mathbb{N}) \quad \delta_n^{-1}(r_n - s_n) \in \mathcal{A} s_n. \quad (6)$$

Let $z \in \text{zer}(\mathcal{A} + \mathcal{B})$ and fix $n \in \mathbb{N}$. We have

$$\begin{aligned}
\|z_{n+1} - z\|^2 &= \|(1 - \lambda_n)(z_n - z) + \lambda_n(z_n - z + t_n - r_n)\|^2 \\
&= (1 - \lambda_n)\|z_n - z\|^2 + \lambda_n\|z_n - z + t_n - r_n\|^2 - \lambda_n(1 - \lambda_n)\|t_n - r_n\|^2 \\
&= (1 - \lambda_n)\|z_n - z\|^2 + \lambda_n\|s_n - \delta_n(\mathcal{B} s_n - \mathcal{B} z_n) - z\|^2 - \lambda_n(1 - \lambda_n)\|t_n - r_n\|^2 \\
&\leq (1 - \lambda_n)\|z_n - z\|^2 + \lambda_n(\|z_n - z\|^2 + \delta_n^2\|\mathcal{B} s_n - \mathcal{B} z_n\|^2 - \|s_n - z_n\|^2) - \lambda_n(1 - \lambda_n)\|t_n - r_n\|^2 \\
&\leq \|z_n - z\|^2 - \varepsilon(1 - (\delta_n \eta)^2)\|s_n - z_n\|^2 - \lambda_n(1 - \lambda_n)\|t_n - r_n\|^2,
\end{aligned} \quad (7)$$

where the first and third equality follow from (5), the second equality is a consequence of [30, Corollary 2.14], the inequality in the fourth line is obtained from [22, Lemma 3.1], and the last inequality follows from the Lipschitz property on \mathcal{B} , $\sup_{n \in \mathbb{N}} \delta_n < 1/\eta$, and $\inf_{n \in \mathbb{N}} \lambda_n \geq \varepsilon$. Hence, since $\delta_n < 1/\eta$ and $0 < \lambda_n \leq 1$, we obtain $\|z_{n+1} - z\|^2 \leq \|z_n - z\|^2$, which yields the boundedness of the sequence $(z_k)_{k \in \mathbb{N}}$. Moreover, we deduce from (7) and [23, Lemma 2.1] that $(\|s_k - z_k\|^2)_{k \in \mathbb{N}}$ and $(\|t_k - r_k\|^2)_{k \in \mathbb{N}}$ are summable and, in particular,

$$s_k - z_k \rightarrow 0 \quad \text{and} \quad t_k - r_k \rightarrow 0, \quad (8)$$

which yields $z_{k+1} - z_k = \lambda_k(t_k - r_k) \rightarrow 0$. By setting, for every $k \in \mathbb{N}$, $u_k := \delta_k^{-1}(r_k - t_k)$, it follows from (5), (6), and (8) that, $u_k = \delta_k^{-1}(r_k - s_k) + \mathcal{B}s_k \in (\mathcal{A} + \mathcal{B})s_k$ and $u_k \rightarrow 0$. Hence, for any weak cluster point of $(z_k)_{k \in \mathbb{N}}$, say $z_{k_\ell} \rightharpoonup w$, (8) yields $s_{k_\ell} \rightharpoonup w$, $u_{k_\ell} \rightarrow 0$, and $(s_{k_\ell}, u_{k_\ell}) \in \text{gra}(\mathcal{A} + \mathcal{B})$. Since \mathcal{B} is monotone and continuous, it is maximally monotone [30, Corollary 20.25]. Moreover, since $\text{dom } \mathcal{B} = \mathcal{H}$, we deduce from [30, Corollary 24.4(i)] that $\mathcal{A} + \mathcal{B}$ is maximally monotone and, hence, its graph is sequentially closed in $\mathcal{H}^{\text{weak}} \times \mathcal{H}^{\text{strong}}$ [30, Proposition 20.33(ii)], which yields $w \in \text{zer}(\mathcal{A} + \mathcal{B})$. Finally, from [23, Lemma 2.2], we deduce that there exists $\bar{z} \in \text{zer}(\mathcal{A} + \mathcal{B})$ such that $z_n \rightharpoonup \bar{z}$. \square

Remark 2.1 As in [23, Theorem 2.5], absolutely summable errors can be incorporated in each step of the algorithm in (5). However, for ease of presentation, we only provide the error-free version.

3 Forward–Partial Inverse–Forward Splitting

We aim at solving the following problem.

Problem 3.1 Let \mathcal{H} be a real Hilbert space and let V be a closed vector subspace of \mathcal{H} . Let $A: \mathcal{H} \rightrightarrows \mathcal{H}$ be a maximally monotone operator and let $B: \mathcal{H} \rightarrow \mathcal{H}$ be monotone and χ -Lipschitzian for some $\chi \in]0, +\infty[$. The problem is to

$$\text{find } x \in \mathcal{H} \quad \text{such that} \quad 0 \in Ax + Bx + N_V x, \quad (9)$$

under the assumption $Z := \text{zer}(A + B + N_V) \neq \emptyset$.

In Problem 3.1, the operator N_V is separated from A in order to exploit the intrinsic structure of each operator. Indeed, the proposed method and its variants activate separately each constituent

of the inclusion, as we will see in Section 3.2. In Section 4, we justify the importance of the vector subspace framework via some applications, in which this setting appears naturally. In this section, we study a characterisation of the solutions to Problem 3.1, we provide our algorithm, and we prove its convergence to a solution to Problem 3.1.

3.1 Characterisation

The following result provides a characterisation of the solutions to Problem 3.1, in terms of two suitable monotone operators.

Proposition 3.1 *In the context of Problem 3.1, let $\gamma \in]0, +\infty[$ and define*

$$\mathcal{A}_\gamma := (\gamma A)_V : \mathcal{H} \rightrightarrows \mathcal{H} \quad \text{and} \quad \mathcal{B}_\gamma := \gamma P_V \circ B \circ P_V : \mathcal{H} \rightarrow V. \quad (10)$$

Then, the following hold:

- (i) \mathcal{A}_γ is maximally monotone and, for every $\delta \in]0, +\infty[$ and $x \in \mathcal{H}$, there exist p and q in \mathcal{H} such that $x = p + \gamma q$, $J_{\delta \mathcal{A}_\gamma} x = P_V p + \gamma P_{V^\perp} q$, and

$$\frac{P_V q}{\delta} + P_{V^\perp} q \in A \left(P_V p + \frac{P_{V^\perp} p}{\delta} \right). \quad (11)$$

In particular, $J_{\mathcal{A}_\gamma} = 2P_V J_{\gamma A} - J_{\gamma A} + \text{Id} - P_V = (\text{Id} + R_{N_V} R_{\gamma A})/2$.

- (ii) \mathcal{B}_γ is monotone and $\gamma\chi$ -Lipschitzian.
(iii) Let $x \in \mathcal{H}$. Then, x is a solution to Problem 3.1 if and only if $x \in V$ and

$$(\exists y \in V^\perp \cap (Ax + Bx)) \text{ such that } x + \gamma(y - P_{V^\perp} Bx) \in \text{zer}(\mathcal{A}_\gamma + \mathcal{B}_\gamma). \quad (12)$$

- (iv) $Z = P_V(\text{zer}(\mathcal{A}_\gamma + \mathcal{B}_\gamma))$.

Proof. (i): Since γA is maximally monotone, \mathcal{A}_γ inherits this property [9, Proposition 2.1]. In addition, for every $(r, x) \in \mathcal{H}^2$ and $\delta \in]0, +\infty[$, it follows from (2) that

$$\begin{aligned} r = J_{\delta \mathcal{A}_\gamma} x &\Leftrightarrow \frac{x - r}{\delta} \in \mathcal{A}_\gamma r \\ &\Leftrightarrow \frac{P_V(x - r)}{\delta} + P_{V^\perp} r \in \gamma A \left(P_V r + \frac{P_{V^\perp}(x - r)}{\delta} \right) \\ &\Leftrightarrow \frac{P_V(x - r)}{\gamma \delta} + \frac{P_{V^\perp} r}{\gamma} \in A \left(P_V r + \frac{P_{V^\perp}(x - r)}{\delta} \right). \end{aligned} \quad (13)$$

Hence, by taking $p := P_{V^\perp}(x-r) + P_V r$ and $q := (P_V(x-r) + P_{V^\perp} r)/\gamma$, we have $p + \gamma q = (x-r) + r = x$, $P_V p + \gamma P_{V^\perp} q = r = J_{\delta \mathcal{A}, \gamma} x$, and (11). Now, in the particular case when $\delta = 1$, (11) reduces to $p = J_{\gamma A}(p + \gamma q) = J_{\gamma A} x$ and, hence,

$$\begin{aligned}
J_{\mathcal{A}, \gamma} x &= P_V(J_{\gamma A} x) + P_{V^\perp}(x - J_{\gamma A} x) \\
&= 2P_V J_{\gamma A} x - J_{\gamma A} x + x - P_V x \\
&= \frac{1}{2}(x + 2P_V(2J_{\gamma A} x - x) - 2J_{\gamma A} x + x) \\
&= \frac{1}{2}(x + R_{N_V} R_{\gamma A} x).
\end{aligned} \tag{14}$$

(ii): Let $(x, y) \in \mathcal{H}^2$. We have from (10), the monotonicity of B , the linearity of P_V , and $P_V^* = P_V$ that $\langle x - y \mid \mathcal{B}_\gamma x - \mathcal{B}_\gamma y \rangle = \gamma \langle P_V x - P_V y \mid B(P_V x) - B(P_V y) \rangle \geq 0$ and, from the Lipschitzian property on B and (10), we obtain $\|\mathcal{B}_\gamma x - \mathcal{B}_\gamma y\| \leq \gamma \|B(P_V x) - B(P_V y)\| \leq \gamma \chi \|P_V x - P_V y\| \leq \gamma \chi \|x - y\|$.

(iii): Let $x \in \mathcal{H}$ be a solution to Problem 3.1. We have $x \in V$, and there exists $y \in V^\perp = N_V x$ such that $y \in Ax + Bx$. Since B is single valued and P_V is linear, it follows from (2) that

$$\begin{aligned}
y \in Ax + Bx &\Leftrightarrow \gamma y - \gamma Bx \in \gamma Ax \\
&\Leftrightarrow -\gamma P_V(Bx) \in (\gamma A)_V(x + \gamma(y - P_{V^\perp} Bx)) \\
&\Leftrightarrow 0 \in (\gamma A)_V(x + \gamma(y - P_{V^\perp} Bx)) + \gamma P_V(B(P_V(x + \gamma(y - P_{V^\perp} Bx)))) \\
&\Leftrightarrow x + \gamma(y - P_{V^\perp} Bx) \in \text{zer}(\mathcal{A}_\gamma + \mathcal{B}_\gamma),
\end{aligned} \tag{15}$$

which yields the result. (iv): Direct from (iii). \square

3.2 Algorithm and Convergence

In the following result, we propose our algorithm and we prove its convergence to a solution to Problem 3.1. Since Proposition 3.1 asserts that Problem 3.1 can be written as a monotone inclusion involving a maximally monotone operator and a single-valued Lipschitzian monotone operator, our method is a consequence of Proposition 2.2 applied to this context.

Algorithm 3.1 In the context of Problem 3.1, let $\gamma \in]0, +\infty[$, let $\varepsilon \in]0, \max\{1, 1/(2\gamma\chi)\}[$, let $(\delta_n)_{n \in \mathbb{N}}$ be a sequence in $[\varepsilon, 1/(\gamma\chi) - \varepsilon]$, let $(\lambda_n)_{n \in \mathbb{N}}$ be a sequence in $[\varepsilon, 1]$, let $x_0 \in V$, let $y_0 \in V^\perp$, and, for every $n \in \mathbb{N}$,

$$\begin{aligned} \text{Step 1. find } (p_n, q_n) \in \mathcal{H}^2 \text{ such that } x_n - \delta_n \gamma P_V B x_n + \gamma y_n &= p_n + \gamma q_n \\ \text{and } \frac{P_V q_n}{\delta_n} + P_{V^\perp} q_n &\in A\left(P_V p_n + \frac{P_{V^\perp} p_n}{\delta_n}\right). \end{aligned} \quad (16)$$

$$\begin{aligned} \text{Step 2. set } x_{n+1} &:= x_n + \lambda_n(P_V p_n + \delta_n \gamma P_V(Bx_n - BP_V p_n) - x_n) \\ \text{and } y_{n+1} &:= y_n + \lambda_n(P_{V^\perp} q_n - y_n). \text{ Go to Step 1.} \end{aligned}$$

Theorem 3.1 *Let $(x_n)_{n \in \mathbb{N}}$ and $(y_n)_{n \in \mathbb{N}}$ be the sequences generated by Algorithm 3.1. Then $(x_n)_{n \in \mathbb{N}}$ and $(y_n)_{n \in \mathbb{N}}$ are in V and V^\perp , respectively, $x_n \rightarrow \bar{x}$ and $y_n \rightarrow \bar{y}$ for some solution $\bar{x} \in \text{zer}(A+B+N_V)$ and $\bar{y} \in V^\perp \cap (A\bar{x} + P_V B\bar{x})$, $x_{n+1} - x_n \rightarrow 0$, and $y_{n+1} - y_n \rightarrow 0$.*

Proof. Since $x_0 \in V$ and $y_0 \in V^\perp$, (16) yields $(x_n)_{n \in \mathbb{N}} \subset V$ and $(y_n)_{n \in \mathbb{N}} \subset V^\perp$. Thus, for every $n \in \mathbb{N}$, it follows from (16) and Proposition 3.1(i) that

$$P_V p_n + \gamma P_{V^\perp} q_n = J_{\delta_n(\gamma A)_V}(x_n + \gamma y_n - \delta_n \gamma P_V B x_n). \quad (17)$$

For every $n \in \mathbb{N}$, denote $z_n := x_n + \gamma y_n$ and

$$s_n := J_{\delta_n(\gamma A)_V}(x_n + \gamma y_n - \delta_n \gamma P_V B x_n) = J_{\delta_n \mathcal{A}_\gamma}(z_n - \delta_n \gamma P_V B P_V z_n) = J_{\delta_n \mathcal{A}_\gamma}(z_n - \delta_n \mathcal{B}_\gamma z_n). \quad (18)$$

Hence, it follows from (17) that $P_V p_n = P_V s_n$, $\gamma P_{V^\perp} q_n = P_{V^\perp} s_n$, and, from (16), we obtain

$$x_{n+1} = x_n + \lambda_n(P_V s_n + \delta_n \gamma P_V(Bx_n - BP_V s_n) - x_n) \quad \text{and} \quad \gamma y_{n+1} = \gamma y_n + \lambda_n(P_{V^\perp} s_n - \gamma y_n). \quad (19)$$

By adding the latter equations, we deduce that the algorithm described in (16) can be written as

$$(\forall n \in \mathbb{N}) \quad \begin{cases} r_n := z_n - \delta_n \mathcal{B}_\gamma z_n \\ s_n = J_{\delta_n \mathcal{A}_\gamma} r_n \\ t_n := s_n - \delta_n \mathcal{B}_\gamma s_n \\ z_{n+1} = z_n + \lambda_n(t_n - r_n), \end{cases} \quad (20)$$

which is a particular instance of (5) when $\mathcal{B} = \mathcal{B}_\gamma$ and $\mathcal{A} = \mathcal{A}_\gamma$. Therefore, it follows from Proposition 3.1(i)&(ii) and Proposition 2.2 that $z_n \rightarrow \bar{z} \in \text{zer}(\mathcal{A}_\gamma + \mathcal{B}_\gamma)$ and $z_{n+1} - z_n \rightarrow 0$. By defining $\bar{x} := P_V \bar{z}$ and $\bar{y} := P_{V^\perp} \bar{z} / \gamma$, the results follow from Proposition 3.1 and Proposition 2.2. \square

Remark 3.1

- (i) The Tseng's method allows for errors in the computations of the operators involved [23, 29]. In our algorithm, these inexactitudes have not been considered for simplicity.

- (ii) In the particular case when $\lambda_n \equiv 1$ and $B \equiv 0$ ($\chi = 0$), Algorithm 3.1 reduces to the classical partial inverse method [9] for finding $x \in V$ such that there exists $y \in V^\perp$ satisfying $y \in Ax$.
- (iii) Under further assumptions on the operators \mathcal{A}_γ and/or \mathcal{B}_γ , e.g., as demi-regularity (see [31, Definition 2.3&Proposition 2.4]), strong convergence can be achieved. In particular, under the assumptions of Proposition 4.2 below, the strong monotonicity of \mathcal{A}_γ can be guaranteed, and by following the proof in [23, Theorem 2.5(iii)], we obtain strong convergence of the iterates.

The sequence $(\delta_n)_{n \in \mathbb{N}}$ in Algorithm 3.1 can be manipulated in order to accelerate the convergence. However, as in [9], *Step 1* in Algorithm 3.1 is not always easy to compute. The following results show us a particular case of our method, in which *Step 1* can be obtained explicitly when the resolvent of A is computable. The method can be seen as a forward-Douglas-Rachford-forward splitting.

Corollary 3.1 *In the setting of Problem 3.1, let $\gamma \in]0, 1/\chi[$, let $\varepsilon \in]0, 1[$, let $(\lambda_n)_{n \in \mathbb{N}}$ be a sequence in $[\varepsilon, 1]$, let $z_0 \in \mathcal{H}$, and iterate, for every $n \in \mathbb{N}$,*

$$\begin{cases} r_n := z_n - \gamma P_V B P_V z_n \\ p_n := J_{\gamma A} r_n \\ s_n := 2P_V p_n - p_n + r_n - P_V r_n \\ t_n := s_n - \gamma P_V B P_V s_n \\ z_{n+1} := z_n + \lambda_n (t_n - r_n). \end{cases} \quad (21)$$

Then, by setting, for every $n \in \mathbb{N}$, $x_n := P_V z_n$ and $y_n := P_{V^\perp} z_n / \gamma$, we have $x_n \rightarrow \bar{x}$ and $y_n \rightarrow \bar{y}$ for some $\bar{x} \in \text{zer}(A + B + N_V)$ and $\bar{y} \in V^\perp \cap (A\bar{x} + P_V B\bar{x})$, $x_{n+1} - x_n \rightarrow 0$, and $y_{n+1} - y_n \rightarrow 0$.

Proof. Indeed, it follows from the proof of Theorem 3.1 that Algorithm 3.1 is equivalent to (20), where, for every $n \in \mathbb{N}$, $z_n = x_n + \gamma y_n$. In the particular case when $\delta_n \equiv 1 \in]0, 1/(\gamma\chi)[$, it follows from Proposition 3.1(i) that (20) reduces to (21). Hence, the results follow from Theorem 3.1. \square

Remark 3.2

- (i) Note that, when $V = \mathcal{H}$ and $\lambda_n \equiv 1$, we have $V^\perp = \{0\}$, $P_V = \text{Id}$, $(\text{Id} + R_{N_V} R_{\gamma A})/2 = J_{\gamma A}$, and, therefore, (21) reduces to

$$(\forall n \in \mathbb{N}) \quad \begin{cases} r_n = x_n - \gamma B x_n \\ s_n = J_{\gamma A} r_n \\ t_n = s_n - \gamma B s_n \\ x_{n+1} = x_n + t_n - r_n, \end{cases} \quad (22)$$

which is a version with constant step size of the Tseng's method [22] for finding a zero of $A + B$.

(ii) On the other hand, when $B \equiv 0$, (21) reduces to

$$(\forall n \in \mathbb{N}) \quad \begin{cases} s_n = (z_n + R_{N_V} R_{\gamma A} z_n)/2 \\ z_{n+1} = z_n + \lambda_n (s_n - z_n), \end{cases} \quad (23)$$

which is the Douglas-Rachford splitting method [7, 28] for finding $x \in \mathcal{H}$ such that $0 \in N_V x + Ax$. It coincides with Spingarn's partial inverse method with constant step size [6]. On the other hand, in the particular case when A is a normal cone to a closed and convex set, a detailed study of this method and some extensions and modifications may be found in [32].

(iii) Let \mathbf{H} and \mathbf{G} be real Hilbert spaces, let $\mathbf{L}: \mathbf{H} \rightarrow \mathbf{G}$ be linear and bounded, let \mathbf{A} and \mathbf{B} be maximally monotone operators, and define $T: (x, y) \in \mathcal{H} \mapsto y - \mathbf{L}x$. When $\mathcal{H} = \mathbf{H} \times \mathbf{G}$, $V = \ker T$, $B = 0$, and $A: (x, y) \mapsto \mathbf{A}x \times \mathbf{B}y$, Problem 3.1 reduces to the primal-dual inclusions (see [33])

$$\begin{aligned} \text{find } x \in \mathbf{H} \quad \text{such that} \quad & 0 \in \mathbf{A}x + \mathbf{L}^* \mathbf{B}(\mathbf{L}x) \\ \text{find } u \in \mathbf{G} \quad \text{such that} \quad & 0 \in -\mathbf{L}\mathbf{A}^{-1}(-\mathbf{L}^* u) + \mathbf{B}^{-1}u. \end{aligned} \quad (24)$$

In this context, the algorithm proposed in Corollary 3.1 is the method proposed recently in [33] in the absence of errors. This algorithm needs the computation of P_V , which involves the inverse of a suitable linear operator. Since our framework allows for a non-zero Lipschitzian monotone operator, it can address more complicated structures than (24).

4 Applications

In this section, we study three applications of our method. In each instance, a different closed vector subspace arises naturally, which illustrates the flexibility of our setting. Connections with other algorithms in each framework are provided.

4.1 Inclusion Involving the Sum of m Monotone Operators

Problem 4.1 Let $(\mathbf{H}, |\cdot|)$ be a real Hilbert space, for every $i \in \{1, \dots, m\}$, let $\mathbf{A}_i: \mathbf{H} \rightrightarrows \mathbf{H}$ be maximally monotone, and let $\mathbf{B}: \mathbf{H} \rightarrow \mathbf{H}$ be monotone and χ -Lipschitzian. The problem is to

$$\text{find } x \in \mathbf{H} \quad \text{such that} \quad 0 \in \sum_{i=1}^m \mathbf{A}_i x + \mathbf{B}x, \quad (25)$$

under the assumption that solutions exist.

Problem 4.1 has several applications in image processing, principally in the variational setting (see, e.g., [1,34] and the references therein), variational inequalities [4,5], partial differential equations [10,11], and economics [12,13], among others. In [34,35], Problem 4.1 is solved by a fully split algorithm in the particular case when \mathbf{B} is cocoercive. Nevertheless, this approach does not seem to work in the general case. In [27], a method for solving a more general problem than Problem 4.1 is proposed. However, this approach stores and updates at each iteration m dual variables, which may be unfavourable in large scale systems. Our method exploits the whole structure of the problem and it is obtained from Theorem 3.1, when the underlying closed vector subspace is the diagonal space in \mathbf{H}^m .

Let us first provide a connection between Problem 4.1 and Problem 3.1 via product space techniques. Let $(\omega_i)_{1 \leq i \leq m}$ be real numbers in $]0, 1[$ such that $\sum_{i=1}^m \omega_i = 1$, let \mathcal{H} be the real Hilbert space obtained by endowing the Cartesian product \mathbf{H}^m with the scalar product and associated norm, respectively defined by $\langle x | y \rangle := \sum_{i=1}^m \omega_i \langle x_i | y_i \rangle$ and $\|x\| := \sqrt{\sum_{i=1}^m \omega_i |x_i|^2}$, where $x = (x_i)_{1 \leq i \leq m}$ and $y = (y_i)_{1 \leq i \leq m}$ are generic elements of \mathcal{H} .

Proposition 4.1 *In the context of Problem 4.1, define*

$$\begin{aligned} V &:= \{x = (x_i)_{1 \leq i \leq m} \in \mathcal{H} : x_1 = \dots = x_m\}, & j: \mathbf{H} &\rightarrow V \subset \mathcal{H}: x \mapsto (x, \dots, x), \\ A: \mathcal{H} &\rightrightarrows \mathcal{H}: x \mapsto \frac{1}{\omega_1} \mathbf{A}_1 x_1 \times \dots \times \frac{1}{\omega_m} \mathbf{A}_m x_m, & B: \mathcal{H} &\rightarrow \mathcal{H}: x \mapsto (\mathbf{B}x_1, \dots, \mathbf{B}x_m). \end{aligned} \quad (26)$$

Then, the following hold:

- (i) V is a closed vector subspace of \mathcal{H} , for every $x = (x_i)_{1 \leq i \leq m} \in \mathcal{H}$, $P_V x = j(\sum_{i=1}^m \omega_i x_i)$, and, if $x \in V$, then $N_V x = V^\perp = \{x = (x_i)_{1 \leq i \leq m} \in \mathcal{H} : \sum_{i=1}^m \omega_i x_i = 0\}$; otherwise, $N_V x = \emptyset$.
- (ii) $j: \mathbf{H} \rightarrow V$ is a bijective isometry and $j^{-1}: (x, \dots, x) \mapsto x$.
- (iii) A is a maximally monotone operator and, for every $\gamma \in]0, +\infty[$, $J_{\gamma A}: (x_i)_{1 \leq i \leq m} \mapsto (J_{\gamma \mathbf{A}_i / \omega_i} x_i)$.
- (iv) B is monotone and χ -Lipschitzian, $B(j(x)) = j(\mathbf{B}x)$, and $B(V) \subset V$.
- (v) For every $x \in \mathbf{H}$, x is a solution to Problem 4.1 if and only if $j(x) \in \text{zer}(A + B + N_V)$.

Proof. (i)&(ii): They follow from (1) and easy computations. (iii): [30, Proposition 23.16]. (iv): They follow from simple computations by using (26) and the properties on \mathbf{B} . (v): Let $x \in \mathbf{H}$. We have

$$\begin{aligned}
0 \in \sum_{i=1}^m A_i x + Bx &\Leftrightarrow \left(\exists (y_i)_{1 \leq i \leq m} \in \prod_{i=1}^m A_i x \right) \quad 0 = \sum_{i=1}^m y_i + Bx \\
&\Leftrightarrow \left(\exists (y_i)_{1 \leq i \leq m} \in \prod_{i=1}^m A_i x \right) \quad 0 = \sum_{i=1}^m \omega_i (-y_i / \omega_i - Bx) \\
&\Leftrightarrow \left(\exists (y_i)_{1 \leq i \leq m} \in \prod_{i=1}^m A_i x \right) - (y_1 / \omega_1, \dots, y_m / \omega_m) - j(Bx) \in V^\perp \\
&\Leftrightarrow 0 \in A(j(x)) + B(j(x)) + N_V(j(x)) \\
&\Leftrightarrow j(x) \in \text{zer}(A + B + N_V), \tag{27}
\end{aligned}$$

which yields the result. \square

The following algorithm for solving Problem 4.1 is a direct consequence of Corollary 3.1 applied to the monotone inclusion in Proposition 4.1(v).

Algorithm 4.1 In the context of Problem 4.1, let $\gamma \in]0, 1/\chi[$, let $\varepsilon \in]0, 1[$, let $(\lambda_n)_{n \in \mathbb{N}}$ be a sequence in $[\varepsilon, 1]$, let $(z_{i,0})_{1 \leq i \leq m} \in \mathbf{H}^m$, and iterate, for every $n \in \mathbb{N}$,

$$\left[\begin{array}{l}
x_n := \sum_{j=1}^m \omega_j z_{j,n} \\
\text{For } i = 1, \dots, m \\
\left[\begin{array}{l}
r_{i,n} := z_{i,n} - \gamma Bx_n \\
p_{i,n} := J_{\gamma A_i / \omega_i} r_{i,n}
\end{array} \right. \\
q_n := \sum_{j=1}^m \omega_j p_{j,n} \\
\text{For } i = 1, \dots, m \\
\left[\begin{array}{l}
s_{i,n} := 2q_n - p_{i,n} + z_{i,n} - x_n \\
t_{i,n} := s_{i,n} - \gamma Bq_n \\
z_{i,n+1} := z_{i,n} + \lambda_n (t_{i,n} - r_{i,n}).
\end{array} \right.
\end{array} \right. \tag{28}$$

Theorem 4.1 Let $(x_n)_{n \in \mathbb{N}}$ be the sequence generated by Algorithm 4.1. Then, $x_n \rightarrow \bar{x}$ for some solution \bar{x} to Problem 4.1 and $x_{n+1} - x_n \rightarrow 0$.

Proof. Set, for every $n \in \mathbb{N}$, $x_n := j(\mathbf{x}_n)$, $q_n := j(\mathbf{q}_n)$, $s_n := (\mathbf{s}_{i,n})_{1 \leq i \leq m}$, $z_n := (\mathbf{z}_{i,n})_{1 \leq i \leq m}$, and $p_n := (\mathbf{p}_{i,n})_{1 \leq i \leq m}$. It follows from Proposition 4.1(i) and (28) that, for every $n \in \mathbb{N}$, $x_n = P_V z_n$ and $q_n = P_V p_n = P_V s_n$. Hence, it follows from (26) and Proposition 4.1 that (28) can be written equivalently as (21). Altogether, Corollary 3.1 and Proposition 4.1(v) yield the results. \square

Remark 4.1 In the particular case when $m = 2$, $B = 0$, and $\omega_1 = \omega_2 = 1/2$, Algorithm 4.1 reduces to the method in [24, Remark 6.2(ii)] for finding a zero of $A_1 + A_2$, which computes the resolvents of A_1 and A_2 in parallel. When these resolvents are hard to calculate, this method provides an alternative to the Douglas-Rachford splitting [7].

4.2 Primal-Dual Monotone Inclusions

This section is devoted to the numerical resolution of a very general composite primal-dual monotone inclusion involving vector subspaces. The proposed algorithm addresses monotone operators composed with linear transformations, and solves simultaneously primal and dual inclusions.

Let us introduce a partial sum operation with respect to a closed vector subspace. This notion is a generalisation of the parallel sum (see, e.g., [36] and the references therein).

Definition 4.1 Let \mathcal{H} be a real Hilbert space, let $U \subset \mathcal{H}$ be a closed vector subspace, and let $A: \mathcal{H} \rightrightarrows \mathcal{H}$ and $B: \mathcal{H} \rightrightarrows \mathcal{H}$ be non-linear operators. The *partial sum of A and B with respect to U* is defined by

$$A \square_U B := (A_U + B_U)_U. \quad (29)$$

In particular, we have $A \square_{\mathcal{H}} B = A + B$ and $A \square_{\{0\}} B = A \square B = (A^{-1} + B^{-1})^{-1}$.

The operation $A \mapsto A_U$ preserves monotonicity [9]. Hence, $A \square_U B$ is monotone if A and B are monotone. In this section, we are interested in the following problem.

Problem 4.2 Let $\mathbf{H}, (\mathbf{G}_i)_{1 \leq i \leq m}$ be real Hilbert spaces, for every $i \in \{1, \dots, m\}$, let $\mathbf{U} \subset \mathbf{H}$ and $\mathbf{V}_i \subset \mathbf{G}_i$ be closed vector spaces, let $\mathbf{A}: \mathbf{H} \rightrightarrows \mathbf{H}$ and $\mathbf{B}_i: \mathbf{G}_i \rightrightarrows \mathbf{G}_i$ be maximally monotone, let $\mathbf{L}_i: \mathbf{H} \rightarrow \mathbf{G}_i$ be linear and bounded, let $\mathbf{D}_i: \mathbf{G}_i \rightrightarrows \mathbf{G}_i$ be monotone such that $(\mathbf{D}_i)_{\mathbf{V}_i^\perp}$ is ν_i -Lipschitzian for some $\nu_i \in]0, +\infty[$, let $\mathbf{C}: \mathbf{H} \rightarrow \mathbf{H}$ be monotone and μ -Lipschitzian for some $\mu \in]0, +\infty[$, let $\mathbf{z} \in \mathbf{H}$, and let $\mathbf{b}_i \in \mathbf{G}_i$. The problem is to solve the primal inclusion

$$\text{find } \mathbf{x} \in \mathbf{H} \quad \text{such that} \quad \mathbf{z} \in \mathbf{A}\mathbf{x} + N_{\mathbf{U}}\mathbf{x} + \sum_{i=1}^m \left(\mathbf{L}_i^* P_{\mathbf{V}_i} (\mathbf{B}_i \square_{\mathbf{V}_i^\perp} \mathbf{D}_i + N_{\mathbf{V}_i}) P_{\mathbf{V}_i} (\mathbf{L}_i \mathbf{x} - \mathbf{b}_i) \right) + \mathbf{C}\mathbf{x} \quad (30)$$

together with the dual inclusion: find $\mathbf{u}_1 \in \mathbf{G}_1, \dots, \mathbf{u}_m \in \mathbf{G}_m$ such that

$$(\exists \mathbf{x} \in \mathbf{H}) \quad \begin{cases} \mathbf{z} - \sum_{i=1}^m \mathbf{L}_i^* P_{V_i} \mathbf{u}_i \in \mathbf{A}\mathbf{x} + \mathbf{C}\mathbf{x} + N_{\mathbf{U}}\mathbf{x} \\ (\forall i \in \{1, \dots, m\}) \quad \mathbf{u}_i \in P_{V_i}(\mathbf{B}_i \square_{V_i^\perp} \mathbf{D}_i + N_{V_i}) P_{V_i}(\mathbf{L}_i \mathbf{x} - \mathbf{b}_i). \end{cases} \quad (31)$$

The set of solutions to (30) and (31) are denoted by $\mathcal{P} \neq \emptyset$ and $\mathcal{D} \neq \emptyset$, respectively.

In the particular case when $\mathbf{U} = \mathbf{H}$, for every $i \in \{1, \dots, m\}$, $\mathbf{V}_i = \mathbf{G}_i$, $\mathbf{D}_i 0 = \mathbf{G}_i$, for every $\mathbf{y} \neq 0$, $\mathbf{D}_i \mathbf{y} = \emptyset$, and $\mathbf{C} = 0$, Problem 4.2 is solved in [23, 33] via fully split primal-dual algorithms. In particular, in [33], a proximal point algorithm applied to the partial inverse of a maximally monotone operator with respect to the kernel of a linear operator is proposed for solving Problem 4.2. On the other hand, when $\mathbf{U} = \mathbf{H}$ and, for every $i \in \{1, \dots, m\}$, $\mathbf{V}_i = \mathbf{G}_i$, Problem 4.2 is solved by a splitting method proposed in [27]. To the best of our knowledge, the general case has not been tackled in the literature via splitting methods.

Problem 4.2 requires a Lipschitzian condition on $(\mathbf{D}_i \square_{V_i^\perp})_{1 \leq i \leq m}$. When, for every $i \in \{1, \dots, m\}$, $\mathbf{V}_i = \mathbf{G}_i$, this condition reduces to the Lipschitzian property on \mathbf{D}_i^{-1} , which is trivially satisfied, e.g., when $\mathbf{D}_i 0 = \mathbf{G}_i$ and, for every $\mathbf{y} \neq 0$, $\mathbf{D}_i \mathbf{y} = \emptyset$. The next proposition furnishes other non-trivial instances, in which the partial inverse of a monotone operator is Lipschitzian.

Proposition 4.2 *Let $\mathbf{V} \subset \mathbf{H}$ be a closed vector space and suppose that one of the following holds:*

- (i) $\mathbf{D}: \mathbf{H} \rightarrow \mathbf{H}$ is β -strongly monotone and ν -cocoercive.
- (ii) $\mathbf{D} = \nabla f$, where $f: \mathbf{H} \rightarrow]-\infty, +\infty]$ is differentiable, β -strongly convex, and ∇f is ν^{-1} -Lipschitzian.
- (iii) \mathbf{D} is a linear bounded operator satisfying, for every $\mathbf{x} \in \mathbf{H}$, $\langle \mathbf{x} \mid \mathbf{D}\mathbf{x} \rangle \geq \beta \|\mathbf{x}\|^2$, and $\nu = \beta / \|\mathbf{D}\|^2$.

Then, $\mathbf{D}_{\mathbf{V}}$ is α -cocoercive and α -strongly monotone with $\alpha = \min\{\beta, \nu\}/2$. In particular, $\mathbf{D}_{\mathbf{V}}$ is α^{-1} -Lipschitzian.

Proof. (i): Let (\mathbf{x}, \mathbf{u}) and (\mathbf{y}, \mathbf{v}) in $\text{gra}(\mathbf{D}_{\mathbf{V}})$. It follows from (2) that $(P_{\mathbf{V}}\mathbf{x} + P_{\mathbf{V}^\perp}\mathbf{u}, P_{\mathbf{V}}\mathbf{u} + P_{\mathbf{V}^\perp}\mathbf{x})$ and $(P_{\mathbf{V}}\mathbf{y} + P_{\mathbf{V}^\perp}\mathbf{v}, P_{\mathbf{V}}\mathbf{v} + P_{\mathbf{V}^\perp}\mathbf{y})$ are in $\text{gra}(\mathbf{D})$, and, from the strong monotonicity assumption on \mathbf{D} , we have

$$\begin{aligned} \langle \mathbf{x} - \mathbf{y} \mid \mathbf{u} - \mathbf{v} \rangle &= \langle P_{\mathbf{V}}(\mathbf{x} - \mathbf{y}) \mid P_{\mathbf{V}}(\mathbf{u} - \mathbf{v}) \rangle + \langle P_{\mathbf{V}^\perp}(\mathbf{u} - \mathbf{v}) \mid P_{\mathbf{V}^\perp}(\mathbf{x} - \mathbf{y}) \rangle \\ &= \langle P_{\mathbf{V}}\mathbf{x} + P_{\mathbf{V}^\perp}\mathbf{u} - (P_{\mathbf{V}}\mathbf{y} + P_{\mathbf{V}^\perp}\mathbf{v}) \mid P_{\mathbf{V}}\mathbf{u} + P_{\mathbf{V}^\perp}\mathbf{x} - (P_{\mathbf{V}}\mathbf{v} + P_{\mathbf{V}^\perp}\mathbf{y}) \rangle \\ &\geq \beta \|P_{\mathbf{V}}\mathbf{x} + P_{\mathbf{V}^\perp}\mathbf{u} - (P_{\mathbf{V}}\mathbf{y} + P_{\mathbf{V}^\perp}\mathbf{v})\|^2 \\ &= \beta (\|P_{\mathbf{V}}(\mathbf{x} - \mathbf{y})\|^2 + \|P_{\mathbf{V}^\perp}(\mathbf{u} - \mathbf{v})\|^2). \end{aligned} \quad (32)$$

Analogously, the cocoercivity assumption on D yields $\langle x - y \mid u - v \rangle \geq \nu(\|P_V(u - v)\|^2 + \|P_{V^\perp}(x - y)\|^2)$.

Hence, it follows from (32) that

$$\langle x - y \mid u - v \rangle \geq \frac{\beta}{2}(\|P_V(x - y)\|^2 + \|P_{V^\perp}(u - v)\|^2) + \frac{\nu}{2}(\|P_V(u - v)\|^2 + \|P_{V^\perp}(x - y)\|^2), \quad (33)$$

which yields $\langle x - y \mid u - v \rangle \geq \alpha(\|x - y\|^2 + \|u - v\|^2)$, and the result follows. (ii): From the strong convexity of f , we have that $D = \nabla f$ is β -strongly monotone and, from [37], it is ν -cocoercive. Hence, the result follows from (i). (iii): Since D is linear and bounded, we have $\|x\|^2 \geq \|Dx\|^2 / \|D\|^2$. Then, D is β -strongly monotone and ν -cocoercive and the result follows from (i). \square

The following proposition gives a connection between Problem 4.2 and Problem 3.1.

Proposition 4.3 *Set $\mathcal{H} := H \oplus G_1 \oplus \cdots \oplus G_m$, set $\chi := \max\{\mu, \nu_1, \dots, \nu_m\} + \sqrt{\sum_{i=1}^m \|L_i\|^2}$, and set*

$$\begin{aligned} A: \mathcal{H} \rightrightarrows \mathcal{H}: (x, u_1, \dots, u_m) &\mapsto (-z + Ax) \times (P_{V_1} b_1 + (B_1)_{V_1^\perp} u_1) \times \cdots \times (P_{V_m} b_m + (B_m)_{V_m^\perp} u_m) \\ L: \mathcal{H} \rightarrow \mathcal{H}: (x, u_1, \dots, u_m) &\mapsto \left(\sum_{i=1}^m L_i^* P_{V_i} u_i, -P_{V_1} L_1 x, \dots, -P_{V_m} L_m x \right) \\ C: \mathcal{H} \rightarrow \mathcal{H}: (x, u_1, \dots, u_m) &\mapsto (Cx, (D_1)_{V_1^\perp} u_1, \dots, (D_m)_{V_m^\perp} u_m) \\ B: \mathcal{H} \rightarrow \mathcal{H}: (x, u_1, \dots, u_m) &\mapsto (C + L)(x, u_1, \dots, u_m) \\ W &:= U \times V_1 \times \cdots \times V_m. \end{aligned} \quad (34)$$

Then, the following hold:

(i) A is maximally monotone and, for every $\gamma \in]0, +\infty[$,

$$J_{\gamma A}: (x, u_1, \dots, u_m) \mapsto \left(J_{\gamma A}(x + z), J_{\gamma(B_1)_{V_1^\perp}}(u_1 - P_{V_1} b_1), \dots, J_{\gamma(B_m)_{V_m^\perp}}(u_m - P_{V_m} b_m) \right). \quad (35)$$

(ii) L is linear, bounded, $L^* = -L$, and $\|L\| \leq \sqrt{\sum_{i=1}^m \|L_i\|^2}$.

(iii) B is monotone and χ -Lipschitzian.

(iv) W is a closed vector subspace of \mathcal{H} , $N_W: (x, u_1, \dots, u_m) \mapsto N_U x \times N_{V_1} u_1 \times \cdots \times N_{V_m} u_m$, and

$$P_W: (x, u_1, \dots, u_m) \mapsto (P_U x, P_{V_1} u_1, \dots, P_{V_m} u_m).$$

(v) $\text{zer}(A + B + N_W) \subset \mathcal{P} \times \mathcal{D}$.

(vi) $\mathcal{P} \neq \emptyset \Leftrightarrow \text{zer}(A + B + N_W) \neq \emptyset \Leftrightarrow \mathcal{D} \neq \emptyset$.

Proof. (i): Since, for every $i \in \{1, \dots, m\}$, $(B_i)_{V_i^\perp}$ is maximally monotone, the result follows from [30, Proposition 23.15 and Proposition 23.16]. (ii): Let us define $M: (u_1, \dots, u_m) \mapsto \sum_{i=1}^m L_i^* P_{V_i} u_i$. Since

$(L_i)_{1 \leq i \leq m}$ and $(P_{V_i})_{1 \leq i \leq m}$ are linear bounded operators, M is linear and bounded and, for every $x \in H$, $M^*x = (P_{V_1}L_1x, \dots, P_{V_m}L_mx)$. Since L can be written as $L: (x, u_1, \dots, u_m) \mapsto (M(u_1, \dots, u_m), -M^*x)$, we deduce from [23, Proposition 2.7(ii)] that L is linear and bounded, that $L^* = -L$, and that $\|L\| = \|M\|$. Now, for every $(u_1, \dots, u_m) \in G_1 \oplus \dots \oplus G_m$, we have from triangle and Hölder inequalities that $\|M(u_1, \dots, u_m)\| \leq \sum_{i=1}^m \|L_i\| \|P_{V_i}\| \|u_i\| \leq \sum_{i=1}^m \|L_i\| \|u_i\| \leq \sqrt{\sum_{i=1}^m \|L_i\|^2} \sqrt{\sum_{i=1}^m \|u_i\|^2}$. (iii): Since (ii) implies that L is linear, bounded, and skew, it is monotone and $\|L\|$ -Lipschitzian. Moreover, because C and $(D_i)_{V_i^\perp}$ are monotone and Lipschitzian, C is monotone and Lipschitzian with constant $\max\{\mu, \nu_1, \dots, \nu_m\}$. Altogether, $B = C + L$ is monotone and χ -Lipschitzian. (iv): Clear. (v): Let $(x, u_1, \dots, u_m) \in H \times G_1 \times \dots \times G_m$. From (34) and Proposition 2.1, we obtain

$$(x, u_1, \dots, u_m) \in \text{zer}(A + B + N_W)$$

$$\begin{aligned} & \Leftrightarrow \begin{cases} 0 \in -z + Ax + Cx + \sum_{i=1}^m L_i^* P_{V_i} u_i + N_U x \\ 0 \in P_{V_1} b_1 + (B_1)_{V_1^\perp} u_1 + (D_1)_{V_1^\perp} u_1 - P_{V_1} L_1 x + N_{V_1} u_1 \\ \vdots \\ 0 \in P_{V_m} b_m + (B_m)_{V_m^\perp} u_m + (D_m)_{V_m^\perp} u_m - P_{V_m} L_m x + N_{V_m} u_m \end{cases} \\ & \Leftrightarrow \begin{cases} 0 \in -z + Ax + Cx + \sum_{i=1}^m L_i^* P_{V_i} u_i + N_U x \\ P_{V_1}(L_1 x - b_1) \in ((B_1)_{V_1^\perp} + (D_1)_{V_1^\perp} + N_{V_1}) u_1, u_1 \in V_1 \\ \vdots \\ P_{V_m}(L_m x - b_m) \in ((B_m)_{V_m^\perp} + (D_m)_{V_m^\perp} + N_{V_m}) u_m, u_m \in V_m \end{cases} \\ & \Leftrightarrow \begin{cases} 0 \in -z + Ax + Cx + \sum_{i=1}^m L_i^* P_{V_i} u_i + N_U x \\ u_1 \in P_{V_1} ((B_1)_{V_1^\perp} + (D_1)_{V_1^\perp} + N_{V_1})^{-1} P_{V_1}(L_1 x - b_1) \\ \vdots \\ u_m \in P_{V_m} ((B_m)_{V_m^\perp} + (D_m)_{V_m^\perp} + N_{V_m})^{-1} P_{V_m}(L_m x - b_m) \end{cases} \\ & \Leftrightarrow \begin{cases} z - \sum_{i=1}^m L_i^* P_{V_i} u_i \in Ax + Cx + N_U x \\ u_1 \in P_{V_1} (B_1 \square_{V_1^\perp} D_1 + N_{V_1}) P_{V_1}(L_1 x - b_1) \\ \vdots \\ u_m \in P_{V_m} (B_m \square_{V_m^\perp} D_m + N_{V_m}) P_{V_m}(L_m x - b_m) \end{cases} \tag{36} \\ & \Rightarrow z \in Ax + N_U x + \sum_{i=1}^m L_i^* P_{V_i} (B_i \square_{V_i^\perp} D_i + N_{V_i}) P_{V_i}(L_i x - b_i) + Cx, \tag{37} \end{aligned}$$

which yields $x \in \mathcal{P}$. Moreover, (36) yields $(u_1, \dots, u_m) \in \mathcal{D}$. (vi): If $x \in \mathcal{P}$, then there exist (u_1, \dots, u_m) such that (36) holds and, hence, $(u_1, \dots, u_m) \in \mathcal{D}$. Now, if $(u_1, \dots, u_m) \in \mathcal{D}$, then there exists $x \in \mathbf{H}$ such that (36) holds and, hence, $(x, u_1, \dots, u_m) \in \text{zer}(A + B + N_W)$. The last implication follows from (v). \square

Algorithm 4.2 In the setting of Problem 4.2, let $\gamma \in]0, 1/\chi[$, where χ is defined in Proposition 4.3, let $(\lambda_n)_{n \in \mathbb{N}}$ be a sequence in $[\varepsilon, 1]$, let $x_0 \in \mathbf{H}$, let $(u_{i,0})_{1 \leq i \leq m} \in \mathbf{G}_1 \times \dots \times \mathbf{G}_m$, and iterate, for every $n \in \mathbb{N}$,

$$\begin{cases}
 r_{1,n} := x_n - \gamma P_U(CP_U x_n + \sum_{i=1}^m L_i^* P_{V_i} u_{i,n}) \\
 p_{1,n} := J_{\gamma A}(r_{1,n} + \gamma z) \\
 s_{1,n} := 2P_U p_{1,n} - p_{1,n} + r_{1,n} - P_U r_{1,n} \\
 \text{For } i = 1, \dots, m \\
 \begin{cases}
 r_{2,i,n} := u_{i,n} - \gamma P_{V_i}(D_{iV_i^\perp} P_{V_i} u_{i,n} - L_i P_U x_n) \\
 p_{2,i,n} := J_{\gamma B_{iV_i^\perp}}(r_{2,i,n} - \gamma P_{V_i} b_i) \\
 s_{2,i,n} := 2P_{V_i} p_{2,i,n} - p_{2,i,n} + r_{2,i,n} - P_{V_i} r_{2,i,n} \\
 t_{2,i,n} := s_{2,i,n} - \gamma P_{V_i}(D_{iV_i^\perp} P_{V_i} s_{2,i,n} - L_i P_U s_{1,n}) \\
 u_{i,n+1} := u_{i,n} + \lambda_n(t_{2,i,n} - r_{2,i,n})
 \end{cases} \\
 t_{1,n} := s_{1,n} - \gamma P_U(CP_U s_{1,n} + \sum_{i=1}^m L_i^* P_{V_i} s_{2,i,n}) \\
 x_{n+1} := x_n + \lambda_n(t_{1,n} - r_{1,n}).
 \end{cases} \tag{38}$$

Theorem 4.2 Let $(x_n)_{n \in \mathbb{N}}$ and $(u_{1,n})_{n \in \mathbb{N}}, \dots, (u_{m,n})_{n \in \mathbb{N}}$ be the sequences generated by Algorithm 4.2. Then, $x_n \rightarrow \bar{x} \in \mathbf{H}$, $x_{n+1} - x_n \rightarrow 0$, for every $i \in \{1, \dots, m\}$, $u_{i,n} \rightarrow \bar{u}_i \in \mathbf{G}_i$, $u_{i,n+1} - u_{i,n} \rightarrow 0$, and $(P_U \bar{x}, P_{V_1} \bar{u}_1, \dots, P_{V_m} \bar{u}_m)$ is a solution to Problem 4.2.

Proof. By defining, for every $n \in \mathbb{N}$, $z_n := (x_n, u_{1,n}, \dots, u_{m,n})$, $r_n := (r_{1,n}, r_{2,1,n}, \dots, r_{2,m,n})$, $p_n := (p_{1,n}, p_{2,1,n}, \dots, p_{2,m,n})$, $s_n := (s_{1,n}, s_{2,1,n}, \dots, s_{2,m,n})$, and $t_n = (t_{1,n}, t_{2,1,n}, \dots, t_{2,m,n})$, it follows from Proposition 4.3 that (38) is a particular instance of (21). Hence, the results follow from Corollary 3.1 and Proposition 4.3(v). \square

Remark 4.2

- (i) Even if Problem 4.1 can be seen as a particular case of Problem 4.2, the methods in (38) and (28) have different structures. Indeed, in (38), dual variables are updated at each iteration, which may be numerically costly in large scale problems. Only primal variables are updated in Algorithm 4.1.
- (ii) Algorithm 4.2 activates each operator involved in Problem 4.2 independently. The algorithm is explicit in each step if the resolvents of \mathbf{A} and $(\mathbf{B}_{iV_i^\perp})_{1 \leq i \leq m}$ can be computed. Observe that the resolvent of the partial inverse of a monotone operator can be explicitly found via Proposition 3.1(i).
- (iii) Note that, when $\lambda_n \equiv 1$, $\mathbf{U} = \mathbf{H}$, and, for every $i \in \{1, \dots, m\}$, $\mathbf{V}_i = \mathbf{G}_i$, Algorithm 4.2 reduces to the method in [27, Theorem 3.1] with constant step-size.
- (iv) In the simplest case when $m = 2$, $\mathbf{z} = \mathbf{A} = \mathbf{C} = \mathbf{b}_1 = \mathbf{b}_2 = 0$, $\mathbf{L}_1 = \mathbf{L}_2 = \text{Id}$, $\mathbf{U} = \mathbf{H}$, $\mathbf{V}_1 \equiv \mathbf{G}_1$, $\mathbf{V}_2 \equiv \mathbf{G}_2$, $\mathbf{D}_1 0 = \mathbf{G}_1$, $\mathbf{D}_2 0 = \mathbf{G}_2$, and for every $\mathbf{y} \neq 0$, $\mathbf{D}_1 \mathbf{y} = \mathbf{D}_2 \mathbf{y} = \emptyset$, we have, for every $i \in \{1, 2\}$, $\mathbf{D}_{iV_i^\perp} = \mathbf{D}_{i\{0\}} = \mathbf{D}_i^{-1} : \mathbf{y} \mapsto 0$, Problem 4.2 reduces to find a zero of $\mathbf{B}_1 + \mathbf{B}_2$, and (38) becomes

$$(\forall n \in \mathbb{N}) \quad \left\{ \begin{array}{l} \mathbf{p}_{1,n} = J_{\gamma \mathbf{B}_1^{-1}}(\mathbf{u}_{1,n} + \gamma \mathbf{x}_n) \\ \mathbf{p}_{2,n} = J_{\gamma \mathbf{B}_2^{-1}}(\mathbf{u}_{2,n} + \gamma \mathbf{x}_n) \\ \mathbf{x}_{n+1} = \mathbf{x}_n - \gamma \lambda_n (\mathbf{p}_{1,n} + \mathbf{p}_{2,n}) \\ \mathbf{u}_{1,n+1} = (1 - \lambda_n) \mathbf{u}_{1,n} + \lambda_n (\mathbf{p}_{1,n} - \gamma^2 (\mathbf{u}_{1,n} + \mathbf{u}_{2,n})) \\ \mathbf{u}_{2,n+1} = (1 - \lambda_n) \mathbf{u}_{2,n} + \lambda_n (\mathbf{p}_{2,n} - \gamma^2 (\mathbf{u}_{1,n} + \mathbf{u}_{2,n})). \end{array} \right. \quad (39)$$

This method solves this problem and its dual, simultaneously, and differs from the algorithm derived in Remark 4.1.

4.3 Zero-Sum Games

Our last application is devoted to the problem of finding a Nash equilibrium in continuous zero-sum games. Some comments on finite zero-sum games are also provided. This problem can be formulated in the form of Problem 3.1, and it can be solved via an algorithm derived from Algorithm 3.1.

Problem 4.3 For every $i \in \{1, 2\}$, let \mathbf{H}_i and \mathbf{G}_i be real Hilbert spaces, let $\mathbf{C}_i \subset \mathbf{H}_i$ be closed and convex, let $\mathbf{L}_i : \mathbf{H}_i \rightarrow \mathbf{G}_i$ be a linear bounded operator with closed range, let $\mathbf{e}_i \in \mathbf{H}_i$, set $\mathbf{b}_i := \mathbf{L}_i \mathbf{e}_i$, set $\mathbf{S}_i := \{\mathbf{x} \in \mathbf{C}_i : \mathbf{L}_i \mathbf{x} = \mathbf{b}_i\}$, let $\chi \in]0, +\infty[$, and let $f : \mathbf{H}_1 \times \mathbf{H}_2 \rightarrow \mathbb{R}$ be a differentiable function with a χ -Lipschitzian gradient such that, for every $\mathbf{z}_1 \in \mathbf{H}_1$, $f(\mathbf{z}_1, \cdot)$ is concave and, for every $\mathbf{z}_2 \in \mathbf{H}_2$,

$f(\cdot, z_2)$ is convex. Moreover, suppose that $\text{int}(C_1 - e_1) \cap \ker L_1 \neq \emptyset$ and $\text{int}(C_2 - e_2) \cap \ker L_2 \neq \emptyset$. The problem is to

$$\text{find } x_1 \in S_1 \text{ and } x_2 \in S_2 \text{ such that } \begin{cases} x_1 \in \underset{z_1 \in S_1}{\text{Argmin}} f(z_1, x_2) \\ x_2 \in \underset{z_2 \in S_2}{\text{Argmax}} f(x_1, z_2), \end{cases} \quad (40)$$

under the assumption that solutions exist.

Problem 4.3 is a generic zero-sum game, in which the sets S_1 and S_2 are usually convex bounded sets representing mixed strategy spaces. For example, if, for every $i \in \{1, 2\}$, $H_i = \mathbb{R}^{N_i}$, C_i is the positive orthant, $G_i \equiv \mathbb{R}$, $b_i \equiv 1$, and L_i is the sum of the components in \mathbb{R}^{N_i} , then S_i is the simplex in \mathbb{R}^{N_i} . In that case, for a bilinear function f , Problem 4.3 reduces to a finite zero-sum game. Beyond this particular case, Problem 4.3 covers continuous zero-sum games, in which mixed strategies are distributions and L_1 and L_2 are integral operators.

As far as we know, some alternating methods are proposed in [38, 39] for solving Problem 4.3, when the function f has a special separable structure involving specific coupling schemes. On the other hand, a method proposed in [40] can solve Problem 4.3 when the projections onto S_1 and S_2 are computable. However, in infinite dimension, these projections are not always easy to compute, as we will discuss in Example 4.1. The following result provides a convergent algorithm for solving Problem 4.3, which replaces the projections onto S_1 and S_2 by simpler projections onto C_1 , C_2 , $\ker(L_1)$, and $\ker(L_2)$. It is a consequence of Corollary 3.1 when the underlying subspace is $V = \ker(L_1) \times \ker(L_2)$. Let us first introduce the *generalised Moore-Penrose inverse* of a bounded linear operator $L: H \rightarrow G$ with closed range, defined by $L^\dagger: G \rightarrow H: y \mapsto P_{C_y} 0$, where, for every $y \in G$, $C_y := \{x \in H : L^*Lx = L^*y\}$. The operator L^\dagger is also linear and bounded and, in the particular case when L^*L is invertible, we have $L^\dagger = (L^*L)^{-1}L^*$. For further details and properties, see [30, Section 3].

Algorithm 4.3 In the context of Problem 4.3, let $\varepsilon \in]0, 1[$, let $\gamma \in]0, 1/\chi[$, let $(\lambda_n)_{n \in \mathbb{N}}$ be a sequence in $[\varepsilon, 1]$, let $(z_{1,0}, z_{2,0}) \in H_1 \oplus H_2$, and iterate, for every $n \in \mathbb{N}$,

$$\begin{aligned}
\mathbf{u}_{1,n} &:= (\text{Id} - \mathbf{L}_1^* \mathbf{L}_1^{*\dagger}) \mathbf{z}_{1,n} + \mathbf{e}_1 \\
\mathbf{u}_{2,n} &:= (\text{Id} - \mathbf{L}_2^* \mathbf{L}_2^{*\dagger}) \mathbf{z}_{2,n} + \mathbf{e}_2 \\
\mathbf{g}_{1,n} &:= (\text{Id} - \mathbf{L}_1^* \mathbf{L}_1^{*\dagger}) \nabla(f(\cdot, \mathbf{u}_{2,n}))(\mathbf{u}_{1,n}) \\
\mathbf{g}_{2,n} &:= -(\text{Id} - \mathbf{L}_2^* \mathbf{L}_2^{*\dagger}) \nabla(f(\mathbf{u}_{1,n}, \cdot))(\mathbf{u}_{2,n}) \\
\mathbf{r}_{1,n} &:= \mathbf{z}_{1,n} - \gamma \mathbf{g}_{1,n} \\
\mathbf{r}_{2,n} &:= \mathbf{z}_{2,n} - \gamma \mathbf{g}_{2,n} \\
\mathbf{p}_{1,n} &:= P_{\mathbf{C}_1}(\mathbf{r}_{1,n} + \mathbf{e}_1) - \mathbf{e}_1 \\
\mathbf{p}_{2,n} &:= P_{\mathbf{C}_2}(\mathbf{r}_{2,n} + \mathbf{e}_2) - \mathbf{e}_2 \\
\mathbf{v}_{1,n} &:= (\text{Id} - \mathbf{L}_1^* \mathbf{L}_1^{*\dagger}) \mathbf{p}_{1,n} \\
\mathbf{v}_{2,n} &:= (\text{Id} - \mathbf{L}_2^* \mathbf{L}_2^{*\dagger}) \mathbf{p}_{2,n} \\
\mathbf{s}_{1,n} &:= 2\mathbf{v}_{1,n} - \mathbf{p}_{1,n} + \mathbf{L}_1^* \mathbf{L}_1^{*\dagger} \mathbf{r}_{1,n} \\
\mathbf{s}_{2,n} &:= 2\mathbf{v}_{2,n} - \mathbf{p}_{2,n} + \mathbf{L}_2^* \mathbf{L}_2^{*\dagger} \mathbf{r}_{2,n} \\
\mathbf{h}_{1,n} &:= (\text{Id} - \mathbf{L}_1^* \mathbf{L}_1^{*\dagger}) \nabla(f(\cdot, \mathbf{e}_2 + \mathbf{v}_{2,n}))(\mathbf{e}_1 + \mathbf{v}_{1,n}) \\
\mathbf{h}_{2,n} &:= -(\text{Id} - \mathbf{L}_2^* \mathbf{L}_2^{*\dagger}) \nabla(f(\mathbf{e}_1 + \mathbf{v}_{1,n}, \cdot))(\mathbf{e}_2 + \mathbf{v}_{2,n}) \\
\mathbf{t}_{1,n} &:= \mathbf{s}_{1,n} - \gamma \mathbf{h}_{1,n} \\
\mathbf{t}_{2,n} &:= \mathbf{s}_{2,n} - \gamma \mathbf{h}_{2,n} \\
\mathbf{z}_{1,n+1} &:= \mathbf{z}_{1,n} + \lambda_n (\mathbf{t}_{1,n} - \mathbf{r}_{1,n}) \\
\mathbf{z}_{2,n+1} &:= \mathbf{z}_{2,n} + \lambda_n (\mathbf{t}_{2,n} - \mathbf{r}_{2,n}).
\end{aligned} \tag{41}$$

Theorem 4.3 *Let $(\mathbf{u}_{1,n}, \mathbf{u}_{2,n})_{n \in \mathbb{N}}$ be the sequence generated by Algorithm 4.3. Then, $\mathbf{u}_{1,n} \rightarrow \bar{\mathbf{x}}_1$ and $\mathbf{u}_{2,n} \rightarrow \bar{\mathbf{x}}_2$, where $(\bar{\mathbf{x}}_1, \bar{\mathbf{x}}_2)$ is a solution to Problem 4.3.*

Proof. It follows from [30, Theorem 16.2] that Problem 4.3 can be written equivalently as the problem of finding \mathbf{x}_1 and \mathbf{x}_2 such that $0 \in \partial(\iota_{\mathbf{S}_1} + f(\cdot, \mathbf{x}_2))(\mathbf{x}_1)$ and $0 \in \partial(\iota_{\mathbf{S}_2} - f(\mathbf{x}_1, \cdot))(\mathbf{x}_2)$, or, equivalently, such that $0 \in N_{\mathbf{S}_1}(\mathbf{x}_1) + \nabla(f(\cdot, \mathbf{x}_2))(\mathbf{x}_1)$ and $0 \in N_{\mathbf{S}_2}(\mathbf{x}_2) - \nabla(f(\mathbf{x}_1, \cdot))(\mathbf{x}_2)$ [30, Corollary 16.38]. Now since, for every $i \in \{1, 2\}$, $\mathbf{S}_i = \mathbf{C}_i \cap \mathbf{L}_i^{-1}(\mathbf{b}_i) = \mathbf{C}_i \cap (\mathbf{e}_i + \ker \mathbf{L}_i)$, it follows from qualification conditions that Problem 4.3 is equivalent to

$$\begin{aligned}
0 &\in N_{\mathbf{C}_1}(\mathbf{e}_1 + \mathbf{z}_1) + N_{\ker \mathbf{L}_1}(\mathbf{z}_1) + \nabla(f(\cdot, \mathbf{e}_2 + \mathbf{z}_2))(\mathbf{e}_1 + \mathbf{z}_1) \\
0 &\in N_{\mathbf{C}_2}(\mathbf{e}_2 + \mathbf{z}_2) + N_{\ker \mathbf{L}_2}(\mathbf{z}_2) - \nabla(f(\mathbf{e}_1 + \mathbf{z}_1, \cdot))(\mathbf{e}_2 + \mathbf{z}_2),
\end{aligned} \tag{42}$$

where $\mathbf{z}_1 := \mathbf{x}_1 - \mathbf{e}_1$ and $\mathbf{z}_2 := \mathbf{x}_2 - \mathbf{e}_2$. Hence, by defining $V := \ker(L_1) \times \ker(L_2)$ and, for every $(\mathbf{z}_1, \mathbf{z}_2) \in \mathbf{H}_1 \times \mathbf{H}_2$,

$$A(\mathbf{z}_1, \mathbf{z}_2) := N_{C_1 \times C_2}(\mathbf{e}_1 + \mathbf{z}_1, \mathbf{e}_2 + \mathbf{z}_2) \quad \text{and} \quad B(\mathbf{z}_1, \mathbf{z}_2) := \begin{pmatrix} \nabla(f(\cdot, \mathbf{e}_2 + \mathbf{z}_2))(\mathbf{e}_1 + \mathbf{z}_1) \\ -\nabla(f(\mathbf{e}_1 + \mathbf{z}_1, \cdot))(\mathbf{e}_2 + \mathbf{z}_2) \end{pmatrix}, \quad (43)$$

Problem 4.3 is equivalent to find $\mathbf{z}_1 \in \mathbf{H}_1$ and $\mathbf{z}_2 \in \mathbf{H}_2$ such that $0 \in A(\mathbf{z}_1, \mathbf{z}_2) + B(\mathbf{z}_1, \mathbf{z}_2) + N_V(\mathbf{z}_1, \mathbf{z}_2)$. Note that V is a closed vector subspace of $\mathbf{H}_1 \times \mathbf{H}_2$, A is maximally monotone [30, Proposition 20.22], and B is monotone ([30, Proposition 20.22] and [41]). Moreover, since ∇f is χ -Lipschitzian, B is also χ -Lipschitzian. On the other hand, it follows from [30, Proposition 3.28(iii)] and [30, Proposition 23.15(iii)] that, for every $(\mathbf{z}_1, \mathbf{z}_2) \in \mathbf{H}_1 \times \mathbf{H}_2$, $P_V(\mathbf{z}_1, \mathbf{z}_2) = (\mathbf{z}_1 - L_1^* L_1^{*\dagger} \mathbf{z}_1, \mathbf{z}_2 - L_2^* L_2^{*\dagger} \mathbf{z}_2)$, $J_{\gamma A}(\mathbf{z}_1, \mathbf{z}_2) = (P_{C_1}(\mathbf{z}_1 + \mathbf{e}_1) - \mathbf{e}_1, P_{C_2}(\mathbf{z}_2 + \mathbf{e}_2) - \mathbf{e}_2)$, and we deduce that (41) is a particular case of (21) when A , B , and V are defined as before. Altogether, the result follows from Corollary 3.1. \square

Remark 4.3 The proposed method does not need to compute the projection onto S_1 and S_2 at each iteration, but it converges to solution strategies belonging to these sets. This new feature is very useful when the projection onto S_1 and S_2 are not easy to compute.

Example 4.1 We consider a 2-player, zero-sum game, where $X_1 \subset \mathbb{R}^{N_1}$ is a bounded set of pure strategies for player 1 and $S_1 := \{f \in L^2(X_1) : f \geq 0 \text{ a.e.}, \int_{X_1} f(x) dx = 1\}$ is her set of mixed strategies (X_2 , N_2 , and S_2 are defined likewise). We recall that $L^2(X)$ stands for the set of square-integrable functions $f: X \subset \mathbb{R}^n \rightarrow]-\infty, +\infty]$. Moreover, let $F \in L^2(X_1 \times X_2)$ be a function representing the payoff for player 1 and let $-F$ be the payoff of player 2. The problem is to

$$\text{find } f_1 \in S_1 \text{ and } f_2 \in S_2 \text{ such that } \begin{cases} f_1 \in \underset{g_1 \in S_1}{\text{Argmin}} \int_{X_1} \int_{X_2} F(x_1, x_2) g_1(x_1) f_2(x_2) dx_2 dx_1 \\ f_2 \in \underset{g_2 \in S_2}{\text{Argmax}} \int_{X_1} \int_{X_2} F(x_1, x_2) f_1(x_1) g_2(x_2) dx_2 dx_1. \end{cases} \quad (44)$$

Note that S_1 and S_2 are closed and convex sets in $L^2(X_1)$ and $L^2(X_2)$, respectively. Hence, the projection of any square-integrable function onto S_1 or S_2 is well defined. However, these projections are not easy to compute. A possible way for avoiding the explicit computation of these projections is to split S_1 and S_2 in $S_1 = C_1 \cap (\mathbf{e}_1 + \ker L_1)$ and $S_2 = C_2 \cap (\mathbf{e}_2 + \ker L_2)$, as in the proof of Theorem 4.3, where, for every $i \in \{1, 2\}$, $\mathbf{H}_i := L^2(X_i)$, $C_i := \{f \in \mathbf{H}_i : f \geq 0 \text{ a.e.}\}$, $\mathbf{e}_i := (m_i(X_i))^{-1}$, $L_i: f \mapsto \int_{X_i} f(x) dx$,

and $m_i(X_i)$ is the Lebesgue measure of the set X_i . Moreover, define the bilinear differentiable function $f: (f_1, f_2) \mapsto \int_{X_1} \int_{X_2} F(x_1, x_2) f_1(x_1) f_2(x_2) dx_2 dx_1$. It follows from $F \in L^2(X_1 \times X_2)$ that

$$\nabla f: (f_1, f_2) \mapsto \left(\int_{X_2} F(\cdot, x_2) f_2(x_2) dx_2, \int_{X_1} F(x_1, \cdot) f_1(x_1) dx_1 \right) \in \mathbf{H}_1 \times \mathbf{H}_2, \quad (45)$$

and that ∇f is χ -Lipschitzian with $\chi := \|F\|_{L^2(X_1 \times X_2)}$. Thus, (44) is a particular instance of Problem 4.3. Note that, for every $i \in \{1, 2\}$, $\mathbf{L}_i^*: \mathbb{R} \rightarrow L^2(X_i): \xi \mapsto \delta_\xi$, where, for every $\xi \in \mathbb{R}$, $\delta_\xi: x \mapsto \xi$ is the constant function. Moreover, $\mathbf{L}_i \circ \mathbf{L}_i^*: \xi \rightarrow m_i(X_i)\xi$ is invertible and $(\mathbf{L}_i \circ \mathbf{L}_i^*)^{-1}: \xi \mapsto \xi/m_i(X_i)$, which yields $\text{Id} - \mathbf{L}_i^* \mathbf{L}_i^{\dagger} = \text{Id} - \mathbf{L}_i^* (\mathbf{L}_i \circ \mathbf{L}_i^*)^{-1} \mathbf{L}_i: f \mapsto f - \delta_{\bar{f}}$, where $\bar{f} = \int_{X_i} f(x) dx / m_i(X_i)$ is the mean value of f . In addition, for every $i \in \{1, 2\}$, $P_{C_i}: f \mapsto f_+: t \mapsto \max\{0, f(t)\}$. Altogether, Algorithm 4.3 applied to this instance is a fully split method for solving (44). In the particular case when X_1 and X_2 are finite sets of actions (or pure strategies), S_1 and S_2 are finite dimensional simplexes, $F: (x_1, x_2) \mapsto x_1^\top F x_2$, and F is a payoff matrix, Algorithm 4.3 provides a first order method for finding Nash equilibria in the finite zero-sum game (for complements and background on finite games, see [42])

$$\text{find } x_1 \in S_1 \text{ and } x_2 \in S_2 \text{ such that } \begin{cases} x_1 \in \underset{y_1 \in S_1}{\text{Argmin}} & x_1^\top F x_2 \\ x_2 \in \underset{y_2 \in S_2}{\text{Argmax}} & x_1^\top F x_2. \end{cases} \quad (46)$$

When a large number of pure actions are involved (e.g., Texas Hold'em poker), classical linear programming methods for solving (44) are enormous and unsolvable via standard algorithms as simplex. Other attempts use acceleration schemes for obtaining good convergence rates [21]. However, the proposed method does not guarantee the convergence of the iterates. Algorithm 4.3 is an explicit convergent method that solves (46) overcoming previous difficulties. Numerical simulations and comparisons with other methods are part of further research.

5 Conclusions

We provide a fully split algorithm for finding a zero of $A + B + N_V$. The proposed method exploits the intrinsic properties of each of the operators involved, by explicitly activating the single-valued operator B and by computing the resolvent of A and projections onto V . Weak convergence to a zero of $A + B + N_V$ is guaranteed and the flexibility of our framework is illustrated via applications to

monotone inclusions involving m maximally monotone operators, to primal-dual composite inclusions involving partial sums of monotone operators, and continuous zero-sum games. It is worth mentioning that the three applications studied in this paper use different closed vector subspaces: the diagonal vector subspace in a product Hilbert space, the product of vector subspaces, and the kernel of a linear bounded operator. In addition, it follows from Remark 3.2(iii) that it is possible to tackle very complex monotone inclusions, including compositions with linear operators, by using the kernel of appropriate bounded linear operators as closed vector subspaces. The influence of the linear operators of the original inclusion can be split from the other operators involved via these closed vector subspaces. The resulting algorithm shall need to compute inverses of suitable linear operators, which can be obtained easily without perturbing the efficiency of the method in several cases [33]. Altogether, the flexibility of the vector subspace setting gives a promising future to splitting methods involving this feature. It is part of further research to study the performance of the methods under specific assumptions of each problem. On the other hand, the partial sum of two set-valued operators with respect to a closed vector subspace is introduced. This operation preserves monotonicity, and further study will be done in this direction in future work. Finally, in zero-sum games, a splitting method is provided for computing Nash equilibria. The algorithm replaces the projections onto mixed strategy spaces by simpler projections.

Acknowledgements The author is thankful to the anonymous reviewers for comments and suggestions, which improved the quality of the paper. This work was supported by CONICYT under grants FONDECYT 3120054, FONDECYT 11140360, ECOS-CONICYT C13E03, Anillo ACT 1106, Math-Amsud N 13MATH01, and by “Programa de financiamiento basal” from the Center for Mathematical Modeling, Universidad de Chile.

References

1. Combettes, P.L., Pesquet, J.-C.: A proximal decomposition method for solving convex variational inverse problems. *Inverse Problems* 24, 065014, 27 pp (2008)
2. Spingarn, J.E.: Applications of the method of partial inverses to convex programming: decomposition. *Math. Programming* 32, 199–223 (1985)
3. Lions, J.-L., Stampacchia, G.: Variational inequalities. *Comm. Pure Appl. Math.* 20, 493–519 (1967)
4. Tseng, P.: Further applications of a splitting algorithm to decomposition in variational inequalities and convex programming. *Math. Programming* 48, 249–263 (1990)

5. Tseng, P.: Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Control Optim.* 29, 119–138 (1991)
6. Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming* 55, 293–318 (1992)
7. Lions, P.-L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.* 16, 964–979 (1979)
8. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. *SIAM J. Control Optim.* 14, 877–898 (1976)
9. Spingarn, J.E.: Partial inverse of a monotone operator. *Appl. Math. Optim.* 10, 247–265 (1983)
10. Mercier, B.: *Inéquations Variationnelles de la Mécanique*. Publications Mathématiques d’Orsay, no. 80.01, Université de Paris-XI, Orsay, France (1980)
11. Zeidler, E.: *Nonlinear Functional Analysis and Its Applications II/B– Nonlinear Monotone Operators*. Springer-Verlag, New York (1990)
12. Jofré, A., Rockafellar, R.T., Wets, R.J.-B.: Variational inequalities and economic equilibrium. *Math. Oper. Res.* 32, 32–50 (2007)
13. Pennanen, T.: Introduction to convex optimization in financial markets. *Math. Program.* 134, 157–186 (2012)
14. Aujol, J.-F., Aubert, G., Blanc-Feraud, L., Chambolle, A.: Image decomposition into a bounded variation component and an oscillating component. *J. Math. Imaging Vision* 22, 71–88 (2005)
15. Chambolle, A., Lions, P.L.: Image recovery via total variation minimization and related problems. *Numer. Math.* 76, 167–188 (1997)
16. Daubechies, I., Defrise, M., De Mol, C.: An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.* 57, 1413–1457 (2004)
17. Aubin, J.-P., Frankowska, H.: *Set-Valued Analysis*. Birkhäuser, Boston, MA (1990)
18. Showalter, R.E.: *Monotone Operators in Banach Space and Nonlinear Partial Differential Equations*. Mathematical Surveys and Monographs 49, Amer. Math. Soc., Providence, RI (1997)
19. Bertsekas, D.P., Gafni, E.M.: Projection methods for variational inequalities with application to the traffic assignment problem. *Math. Programming Stud.* 17, 139–159 (1982)
20. Fukushima, M.: The primal Douglas-Rachford splitting algorithm for a class of monotone mappings with applications to the traffic equilibrium problem. *Math. Program.* 72, 1–15 (1996)
21. Gilpin, A., Peña, J., Sandholm, T.: First-order algorithm with $\mathcal{O}(\ln(1/\epsilon))$ convergence for ϵ -equilibrium in two-person zero-sum games. *Math. Program. Ser. A* 133, 279–298 (2012)
22. Tseng, P.: A modified forward-backward splitting method for maximal monotone mappings. *SIAM J. Control Optim.* 38, 431–446 (2000)
23. Briceño-Arias, L.M., Combettes, P.L.: A monotone+skew splitting model for composite monotone inclusions in duality. *SIAM J. Optim.* 21, 1230–1250 (2011)
24. Briceño-Arias, L.M.: Forward-Douglas-Rachford splitting and forward-partial inverse method for solving monotone inclusions. *Optimization*, to appear.

25. Passty, G.B.: Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *J. Math. Anal. Appl.* 72, 383–390 (1979)
26. Combettes, P.L.: Iterative construction of the resolvent of a sum of maximal monotone operators. *J. Convex Anal.* 16, 727–748 (2009)
27. Combettes, P.L., Pesquet, J.-C.: Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators. *Set-Valued Var. Anal.* 20, 307–330 (2012)
28. Svaiter, B.F.: Weak convergence on Douglas-Rachford method. *SIAM J. Control Optim.* 49, 280–287 (2011)
29. Briceño-Arias, L.M.: Outer approximation method for constrained composite fixed point problems involving Lipschitz pseudo contractive operators. *Numer. Funct. Anal. Optim.* 32, 1099–1115 (2011)
30. Bauschke, H.H., Combettes, P.L.: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer-Verlag, New York (2011)
31. Attouch, H., Briceño-Arias, L.M., Combettes, P.L.: A parallel splitting method for coupled monotone inclusions. *SIAM J. Control Optim.* 48, 3246–3270 (2010)
32. Luke, D.R.: Finding best approximation pairs relative to a convex and prox-regular set in a Hilbert space. *SIAM J. Optim.* 19, 714–739 (2008).
33. Combettes, P.L.: A primal-dual method of partial inverses for composite inclusions. *Optim. Lett.*, published on-line (2014).
34. Raguét, H., Fadili, J., Peyré, G.: Generalized forward-backward splitting. *SIAM J. Imaging Sci.* 6, 1199–1226 (2013)
35. Vũ, B.C.: A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Adv. Comput. Math.* 38, 667–681 (2013)
36. Boţ, R.I., László, S.: On the generalized parallel sum of two maximal monotone operators of Gossez type (D). *J. Math. Anal. Appl.* 391, 82–98 (2012)
37. Baillon, J.-B., Haddad, G.: Quelques propriétés des opérateurs angle-bornés et n -cycliquement monotones. *Israel J. Math.* 26, 137–150 (1977)
38. Attouch, H., Redont, P., Soubeyran, A.: A new class of alternating proximal minimization algorithms with costs-to-move. *SIAM J. Optim.* 18, 1061–1081 (2007)
39. Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Alternating proximal algorithms for weakly coupled convex minimization problems. Applications to dynamical games and PDE’s. *J. Convex Anal.* 15, 485–506 (2008)
40. Briceño-Arias, L.M., Combettes, P.L.: Monotone operator methods for Nash equilibria in non-potential games. In: Bailey, D., Bauschke, H.H., Borwein, P., Garvan, F., Théra, M., Vanderwerff, J., Wolkowicz, H. (eds.): *Computational and Analytical Mathematics*, vol. 50, pp. 143–159. Springer, New York (2013)
41. Rockafellar, R.T.: Monotone operators associated with saddle-functions and minimax problems. In: Browder, F.E. (ed.): *Nonlinear Functional Analysis Part 1, Proc. Sympos. Pure Math.*, vol. 18, pp. 241–250. Amer. Math. Soc., Providence, RI (1970)
42. Weibull, J.W.: *Evolutionary Game Theory*. MIT Press, Cambridge, MA (1995)