Iteration complexity of an inexact Douglas-Rachford method and of a Douglas-Rachford-Tseng's F-B four-operator splitting method for solving monotone inclusions

M. Marques Alves [*] Marina Geremia [†]

November 30, 2017

Abstract

In this paper, we propose and study the iteration complexity of an inexact Douglas-Rachford splitting (DRS) method and a Douglas-Rachford-Tseng's forward-backward (F-B) splitting method for solving two-operator and four-operator monotone inclusions, respectively. The former method (although based on a slightly different mechanism of iteration) is motivated by the recent work of J. Eckstein and W. Yao, in which an inexact DRS method is derived from a special instance of the hybrid proximal extragradient (HPE) method of Solodov and Svaiter, while the latter one combines the proposed inexact DRS method (used as an outer iteration) with a Tseng's F-B splitting type method (used as an inner iteration) for solving the corresponding subproblems. We prove iteration complexity bounds for both algorithms in the pointwise (non-ergodic) as well as in the ergodic sense by showing that they admit two different iterations: one that can be embedded into the HPE method, for which the iteration complexity is known since the work of Monteiro and Svaiter, and another one which demands a separate analysis. Finally, we perform simple numerical experiments to show the performance of the proposed methods when compared with other existing algorithms.

2000 Mathematics Subject Classification: 47H05, 49M27, 90C25.

Key words: Inexact Douglas-Rachford method; splitting; monotone operators; HPE method; complexity; Tseng's forward-backward method.

1 Introduction

Let $\mathcal{H}$ be a real Hilbert space. In this paper, we consider the *two-operator monotone inclusion problem* (MIP) of finding z such that

$$0 \in A(z) + B(z) \tag{1}$$

[*]Departamento de Matemática, Universidade Federal de Santa Catarina, Florianópolis, Brazil, 88040-900 (maicon.alves@ufsc.br). The work of this author was partially supported by CNPq grants no. 406250/2013-8, 306317/2014-1 and 405214/2016-2.

[†]Departamento de Matemática, Universidade Federal de Santa Catarina, Florianópolis, Brazil, 88040-900 (marinageremia@yahoo.com.br).

as well as the *four-operator* MIP

$$0 \in A(z) + C(z) + F_1(z) + F_2(z) \tag{2}$$

where A, B and C are (set-valued) maximal monotone operators on $\mathcal{H}$, $F_1 : D(F_1) \to \mathcal{H}$ is (point-to-point) *Lipschitz continuous* and $F_2 : \mathcal{H} \to \mathcal{H}$ is (point-to-point) *cocoercive* (see Section 4 for the precise statement). Problems (1) and (2) appear in different fields of applied mathematics and optimization including convex optimization, signal processing, PDEs, inverse problems, among others [2, 22]. Under mild conditions on the operators C, F_1 and F_2, problem (2) becomes a special instance of (1) with $B := C + F_1 + F_2$. This fact will be important later on in this paper.

In this paper, we propose and study the iteration complexity of an inexact Douglas-Rachford splitting method (Algorithm 3) and of a Douglas-Rachford-Tseng's forward-backward (F-B) four-operator splitting method (Algorithm 5) for solving (1) and (2), respectively. The former method is inspired and motivated (although based on a slightly different mechanism of iteration) by the recent work of J. Eckstein and W. Yao [21], while the latter one, which, in particular, will be shown to be a special instance of the former one, is motivated by some variants of the standard Tseng's F-B splitting method [42] recently proposed in the current literature [1, 8, 31]. For more detailed information about the contributions of this paper in the light of reference [21], we refer the reader to the first remark after Algorithm 3. Moreover, we mention that Algorithm 5 is a purely primal splitting method for solving the *four-operator* MIP (2), and this seems to be new. The main contributions of this paper will be discussed in Subsection 1.5.

1.1 The Douglas-Rachford splitting (DRS) method

One of the most popular algorithms for finding approximate solutions of (1) is the *Douglas-Rachford splitting* (DRS) *method*. It consists of an iterative procedure in which at each iteration the resolvents $J_{\gamma A} = (\gamma A + I)^{-1}$ and $J_{\gamma B} = (\gamma B + I)^{-1}$ of A and B, respectively, are employed separately instead of the resolvent $J_{\gamma(A+B)}$ of the full operator $A + B$, which may be expensive to compute numerically. An iteration of the method can be described by

$$z_k = J_{\gamma A}(2J_{\gamma B}(z_{k-1}) - z_{k-1}) + z_{k-1} - J_{\gamma B}(z_{k-1}) \qquad \forall k \geq 1, \tag{3}$$

where $\gamma > 0$ is a scaling parameter and z_{k-1} is the current iterate. Originally proposed in [18] for solving problems with linear operators, the DRS method was generalized in [26] for general nonlinear maximal monotone operators, where the formulation (3) was first obtained. It was proved in [26] that $\{z_k\}$ converges (weakly, in infinite dimensional Hilbert spaces) to some z^* such that $x^* := J_{\gamma B}(z^*)$ is a solution of (1). Recently, [40] proved the (weak) convergence of the sequence $\{z_k\}$ generated in (3) to a solution of (1).

1.2 The Rockafellar's proximal point (PP) method

The *proximal point* (PP) *method* is an iterative method for seeking approximate solutions of the MIP

$$0 \in T(z) \tag{4}$$

where T is a maximal monotone operator on $\mathcal{H}$ for which the solution set of (4) is nonempty. In its exact formulation, an iteration of the PP method can be described by

$$z_k = (\lambda_k T + I)^{-1} z_{k-1} \qquad \forall k \geq 1, \tag{5}$$

where $\lambda_k > 0$ is a stepsize parameter and z_{k-1} is the current iterate. It is well-known that the practical applicability of numerical schemes based on the exact computation of resolvents of monotone operators strongly depends on strategies that allow for inexact computations. This is the case of the PP method (5). In his pioneering work [35], Rockafellar proved that if, at each iteration $k \geq 1$, z_k is computed satisfying

$$\|z_k - (\lambda_k T + I)^{-1} z_{k-1}\| \leq e_k, \quad \sum_{k=1}^{\infty} e_k < \infty, \tag{6}$$

and $\{\lambda_k\}$ is bounded away from zero, then $\{z_k\}$ converges (weakly, in infinite dimensions) to a solution of (4). This result has found important applications in the design and analysis of many practical algorithms for solving challenging problems in optimization and related fields.

1.3 The DRS method is an instance of the PP method (Eckstein and Bertsekas)

In [19], the DRS method (3) was shown to be a special instance of the PP method (5) with $\lambda_k \equiv 1$. More precisely, it was observed in [19] (among other results) that the sequence $\{z_k\}$ in (3) satisfies

$$z_k = (S_{\gamma, A, B} + I)^{-1} z_{k-1} \qquad \forall k \geq 1, \tag{7}$$

where $S_{\gamma, A, B}$ is the maximal monotone operator on $\mathcal{H}$ whose graph is

$$S_{\gamma, A, B} = \{(y + \gamma b, \gamma a + \gamma b) \in \mathcal{H} \times \mathcal{H} \mid b \in B(x),\, a \in A(y),\, \gamma a + y = x - \gamma b\}\,. \tag{8}$$

It can be easily checked that z^* is a solution of (1) if and only if $z^* = J_{\gamma B}(x^*)$ for some x^* such that $0 \in S_{\gamma, A\, B}(x^*)$. The fact that (3) is equivalent to (7) clarifies the proximal nature of the DRS method and allowed [19] to obtain inexact and relaxed versions of it by alternatively describing (7) according to the following procedure:

$$\text{compute } (x_k, b_k) \text{ such that } b_k \in B(x_k) \text{ and } \gamma b_k + x_k = z_{k-1}; \tag{9}$$

$$\text{compute } (y_k, a_k) \text{ such that } a_k \in A(y_k) \text{ and } \gamma a_k + y_k = x_k - \gamma b_k; \tag{10}$$

$$\text{set } z_k = y_k + \gamma b_k. \tag{11}$$

1.4 The hybrid proximal extragradient (HPE) method of Solodov and Svaiter

Many modern inexact versions of the PP method, as opposed to the summable error criterion (6), use *relative error tolerances* for solving the associated subproblems. The first method of this type was proposed in [37], and subsequently studied, e.g., in [31, 32, 33, 36, 37, 38, 39]. The key idea consists of decoupling (5) in an inclusion-equation system:

$$v \in T(z_+), \quad \lambda v + z_+ - z = 0, \tag{12}$$

where $(z, z_+, \lambda) := (z_{k-1}, z_k, \lambda_k)$, and relaxing (12) within relative error tolerance criteria. Among these new methods, the *hybrid proximal extragradient* (HPE) method of Solodov and Svaiter [36], which we discuss in details in Subsection 2.2, has been shown to be very effective as a framework for the design and analysis of many concrete algorithms (see, e.g., [4, 11, 20, 24, 25, 27, 29, 30, 33, 36, 38, 39]).

1.5 The main contributions of this work

In [21], J. Eckstein and W. Yao proposed and studied the (asymptotic) convergence of an inexact version of the DRS method (3) by applying a special instance of HPE method to the maximal monotone operator given in (8). The resulting algorithm (see [21, Algorihm 3]) allows for inexact computations *in the equation* in (9) and, in particular, resulted in an inexact version of the ADMM which is suited for large-scale problems, in which fast inner solvers can be employed for solving the corresponding subproblems (see [21, Section 6]).

In the present work, motivated by [21], we first propose in Section 3 an inexact version of the DRS method (Algorithm 3) for solving (1) in which inexact computations are allowed *in both the inclusion and the equation* in (9). At each iteration, instead of a point in the graph of B, Algorithm 3 computes a point in the graph of the ε-enlargement B^ε of B (it has the property that $B^\varepsilon(z) \supset B(z)$). Moreover, contrary to the reference [21], we study the *iteration complexity* of the proposed method for solving (1). We show that Algorithm 3 admits two type of iterations, one that can be embedded into the HPE method and, on the other hand, another one which demands a separate analysis. We emphasize again that, although motivated by the latter reference, the Douglas-Rachford type method proposed in this paper is based on a slightly different mechanism of iteration, specially designed for allowing its iteration complexity analysis (see Theorems 3.5 and 3.6).

Secondly, in Section 4, we consider the four-operator MIP (2) and propose and study the iteration complexity of a Douglas-Rachford-Tseng's F-B splitting type method (Algorithm 5) which combines Algorithm 3 (as an outer iteration) and a Tseng's F-B splitting type method (Algorithm 4) (as an inner iteration) for solving the corresponding subproblems. The resulting algorithm, namely Algorithm 5, has a splitting nature and solves (2) without introducing extra variables.

Finally, in Section 5, we perform simple numerical experiments to show the performance of the proposed methods when compared with other existing algorithms.

1.6 Most related works

In [6], the relaxed forward-Douglas-Rachford splitting (rFDRS) method was proposed and studied to solve *three-operator MIPs* consisting of (2) with $C = N_V$, V closed vector subspace, and $F_1 = 0$. Subsequently, among other results, the iteration complexity of the latter method (specialized to variational problems) was analyzed in [16]. Problem (2) with $F_1 = 0$ was also considered in [17], where a three-operator splitting (TOS) method was proposed and its iteration complexity studied. On the other hand, problem (2) with $C = N_V$ and $F_2 = 0$ was studied in [7], where the forward-partial inverse-forward splitting method was proposed and analyzed. In [8], a Tseng's F-B splitting type method was proposed and analyzed to solve the special instance of (2) in which $C = 0$.

The iteration complexity of a relaxed Peaceman-Rachford splitting method for solving (1) was recently studied in [34]. The method of [34] was shown to be a special instance of a non-Euclidean HPE framework, for which the iteration complexity was also analyzed in the latter reference (see also [23]). Moreover, as we mentioned earlier, an inexact version of the DRS method for solving (1) was proposed and studied in [21].

2 Preliminaries and background materials

2.1 General notation and ε-enlargements

We denote by $\mathcal{H}$ a real Hilbert space with inner product $\langle\cdot,\cdot\rangle$ and induced norm $\|\cdot\| := \sqrt{\langle\cdot,\cdot\rangle}$ and by $\mathcal{H}\times\mathcal{H}$ the product Cartesian endowed with usual inner product and norm.

A set-valued map $T:\mathcal{H}\rightrightarrows\mathcal{H}$ is said to be a *monotone operator* on $\mathcal{H}$ if $\langle z-z', v-v'\rangle \geq 0$ for all $v\in T(z)$ and $v'\in T(z')$. Moreover, T is a *maximal monotone operator* if T is monotone and $T=S$ whenever S is monotone on $\mathcal{H}$ and $T\subset S$. Here, we identify any monotone operator T with its graph, i.e., we set $T=\{(z,v)\in\mathcal{H}\times\mathcal{H}\,|\,v\in T(z)\}$. The *sum* $T+S$ of two set-valued maps T,S is defined via the usual Minkowski sum and for $\lambda\geq 0$ the operator λT is defined by $(\lambda T)(z)=\lambda T(z):=\{\lambda v\,|\,v\in T(z)\}$. The *inverse* of $T:\mathcal{H}\rightrightarrows\mathcal{H}$ is $T^{-1}:\mathcal{H}\rightrightarrows\mathcal{H}$ defined by $v\in T^{-1}(z)$ if and only if $z\in T(v)$. In particular, $\operatorname{zer}(T):=T^{-1}(0)=\{z\in\mathcal{H}\,|\,0\in T(z)\}$. The *resolvent* of a maximal monotone operator T is $J_T:=(T+I)^{-1}$, where I denotes the identity map on $\mathcal{H}$, and, in particular, the following holds: $x=J_{\lambda T}(z)$ if and only if $\lambda^{-1}(z-x)\in T(x)$ if and only if $0\in\lambda T(x)+x-z$. We denote by $\partial_\varepsilon f$ the usual ε-subdifferential of a proper closed convex function $f:\mathcal{H}\to(-\infty,+\infty]$ and by $\partial f:=\partial f_0$ the Fenchel-subdifferential of f as well. The *normal cone* of a closed convex set X will be denoted by N_X and by P_X we denote the orthogonal projection onto X.

For $T:\mathcal{H}\rightrightarrows\mathcal{H}$ maximal monotone and $\varepsilon\geq 0$, the ε-enlargement [9, 28] of T is the operator $T^\varepsilon:\mathcal{H}\rightrightarrows\mathcal{H}$ defined by

$$T^\varepsilon(z) := \{v\in\mathcal{H}\mid \langle z-z', v-v'\rangle \geq -\varepsilon\;\;\forall (z',v')\in T\}\quad \forall z\in\mathcal{H}. \tag{13}$$

Note that $T(z)\subset T^\varepsilon(z)$ for all $z\in\mathcal{H}$.

The following summarizes some useful properties of T^ε which will be useful in this paper.

Proposition 2.1. *Let $T,S:\mathcal{H}\rightrightarrows\mathcal{H}$ be set-valued maps. Then,*

(a) *if $\varepsilon\leq\varepsilon'$, then $T^\varepsilon(x)\subseteq T^{\varepsilon'}(x)$ for every $x\in\mathcal{H}$;*

(b) *$T^\varepsilon(x)+S^{\varepsilon'}(x)\subseteq(T+S)^{\varepsilon+\varepsilon'}(x)$ for every $x\in\mathcal{H}$ and $\varepsilon,\varepsilon'\geq 0$;*

(c) *T is monotone if, and only if, $T\subseteq T^0$;*

(d) *T is maximal monotone if, and only if, $T=T^0$;*

Next we present the transportation formula for ε-enlargements.

Theorem 2.2. ([10, Theorem 2.3]) *Suppose $T:\mathcal{H}\rightrightarrows\mathcal{H}$ is maximal monotone and let $z_\ell, v_\ell\in\mathcal{H}$, $\varepsilon_\ell,\alpha_\ell\in\mathbb{R}_+$, for $\ell=1,\dots,j$, be such that*

$$v_\ell\in T^{\varepsilon_\ell}(z_\ell),\quad \ell=1,\dots,j,\quad \sum_{\ell=1}^{j}\alpha_\ell=1,$$

and define

$$\overline{z}_j := \sum_{\ell=1}^{j}\alpha_\ell\, z_\ell\,,\quad \overline{v}_j := \sum_{\ell=1}^{j}\alpha_\ell\, v_\ell\,,\quad \overline{\varepsilon}_j := \sum_{\ell=1}^{j}\alpha_\ell\left[\varepsilon_\ell+\langle z_\ell-\overline{z}_j, v_\ell-\overline{v}_j\rangle\right].$$

Then, the following hold:

(a) $\overline{\varepsilon}_j \geq 0$ *and* $\overline{v}_j \in T^{\overline{\varepsilon}_j}(\overline{z}_j)$.

(b) *If, in addition,* $T = \partial f$ *for some proper, convex and closed function* f *and* $v_\ell \in \partial_{\varepsilon_\ell} f(z_\ell)$ *for* $\ell = 1, \ldots, j$, *then* $\overline{v}_j \in \partial_{\overline{\varepsilon}_j} f(\overline{z}_j)$.

2.2 The hybrid proximal extragradient (HPE) method

Consider the *monotone inclusion problem* (MIP) (4), i.e.,

$$0 \in T(z) \tag{14}$$

where $T : \mathcal{H} \rightrightarrows \mathcal{H}$ is a maximal monotone operator for which the solution set $T^{-1}(0)$ of (14) is nonempty.

As we mentioned earlier, the proximal point (PP) method of Rockafellar [35] is one of the most popular algorithms for finding approximate solutions of (14) and, among the modern inexact versions of the PP method, the *hybrid proximal extragradient* (HPE) method of [36], which we present in what follows, has been shown to be very effective as a framework for the design and analysis of many concrete algorithms (see e.g. [4, 11, 20, 24, 25, 27, 29, 30, 33, 36, 38, 39]).

Algorithm 1. Hybrid proximal extragradient (HPE) method for (14)

(0) Let $z_0 \in \mathcal{H}$ and $\sigma \in [0, 1)$ be given and set $j \leftarrow 1$.

(1) Compute $(\widetilde{z}_j, v_j, \varepsilon_j) \in \mathcal{H} \times \mathcal{H} \times \mathbb{R}_+$ and $\lambda_j > 0$ such that

$$v_j \in T^{\varepsilon_j}(\widetilde{z}_j), \quad \|\lambda_j v_j + \widetilde{z}_j - z_{j-1}\|^2 + 2\lambda_j \varepsilon_j \leq \sigma^2 \|\widetilde{z}_j - z_{j-1}\|^2. \tag{15}$$

(2) Define

$$z_j = z_{j-1} - \lambda_j v_j, \tag{16}$$

set $j \leftarrow j + 1$ and go to step 1.

Remarks.

1. If $\sigma = 0$ in (15), then it follows from Proposition 2.1(d) and (16) that $(z_+, v) := (z_j, v_j)$ and $\lambda := \lambda_j > 0$ satisfy (12), which means that the HPE method generalizes the exact Rockafellar's PP method.

2. Condition (15) clearly relaxes both the inclusion and the equation in (12) within a relative error criterion. Recall that $T^\varepsilon(\cdot)$ denotes the ε-enlargement of T and has the property that $T^\varepsilon(z) \supset T(z)$ (see Subsection 2.1 for details). Moreover, in (16) an extragradient step from the current iterate z_{j-1} gives the next iterate z_j.

3. We emphasize that specific strategies for computing the triple $(\widetilde{z}_j, v_j, \varepsilon_j)$ as well as the step-size $\lambda_j > 0$ satisfying (15) will depend on the particular instance of the problem (14) under consideration. On the other hand, as mentioned before, the HPE method can also be used as

a framework for the design and analysis of concrete algorithms for solving specific instances of (14) (see, e.g., [20, 29, 30, 31, 32, 33]). We also refer the reader to Sections 3 and 4, in this work, for applications of the HPE method in the context of decomposition/splitting algorithms for monotone inclusions.

Since the appearance of the paper [32], we have seen an increasing interest in studding the *iteration complexity* of the HPE method and its special instances (e.g., Tseng's forward-backward splitting method, Korpelevich extragradient method and ADMM [31, 32, 33]). This depends on the following termination criterion [32]: given tolerances $\rho, \epsilon > 0$, find $z, v \in \mathcal{H}$ and $\varepsilon > 0$ such that

$$v \in T^{\varepsilon}(z), \quad \|v\| \le \rho, \quad \varepsilon \le \epsilon. \tag{17}$$

Note that, by Proposition 2.1(d), if $\rho = \epsilon = 0$ in (17) then $0 \in T(z)$, i.e., $z \in T^{-1}(0)$.

We now summarize the main results on *pointwise (non ergodic)* and *ergodic* iteration complexity [32] of the HPE method that will be used in this paper. The *aggregate stepsize sequence* $\{\Lambda_j\}$ and the *ergodic sequences* $\{\overline{\widetilde{z}}_j\}$, $\{\overline{v}_j\}$, $\{\overline{\varepsilon}_j\}$ associated to $\{\lambda_j\}$ and $\{\widetilde{z}_j\}$, $\{v_j\}$, and $\{\varepsilon_j\}$ are, respectively,

$$\Lambda_j := \sum_{\ell=1}^{j} \lambda_\ell\,, \tag{18}$$

$$\overline{\widetilde{z}}_j := \frac{1}{\Lambda_j} \sum_{\ell=1}^{j} \lambda_\ell\, \widetilde{z}_\ell, \quad \overline{v}_j := \frac{1}{\Lambda_j} \sum_{\ell=1}^{j} \lambda_\ell\, v_\ell, \tag{19}$$

$$\overline{\varepsilon}_j := \frac{1}{\Lambda_j} \sum_{\ell=1}^{j} \lambda_\ell \left[\varepsilon_\ell + \langle \widetilde{z}_\ell - \overline{\widetilde{z}}_j, v_\ell - \overline{v}_j\rangle\right] = \frac{1}{\Lambda_j} \sum_{\ell=1}^{j} \lambda_\ell \left[\varepsilon_\ell + \langle \widetilde{z}_\ell - \overline{\widetilde{z}}_j, v_\ell\rangle\right]. \tag{20}$$

Theorem 2.3 ([32, Theorem 4.4(a) and 4.7])**.** *Let* $\{\widetilde{z}_j\}$, $\{v_j\}$, *etc, be generated by the* HPE method (Algorithm 1) *and let* $\{\overline{\widetilde{z}}_j\}$, $\{\overline{v}_j\}$, *etc, be given in* (18)–(20)*. Let also* d_0 *denote the distance from* z_0 *to* $T^{-1}(0) \neq \emptyset$ *and assume that* $\lambda_j \ge \underline{\lambda} > 0$ *for all* $j \ge 1$*. Then, the following hold:*

(a) *For any* $j \ge 1$, *there exists* $i \in \{1, \ldots, j\}$ *such that*

$$v_i \in T^{\varepsilon_i}(\widetilde{z}_i), \quad \|v_i\| \le \frac{d_0}{\underline{\lambda}\sqrt{j}}\sqrt{\frac{1+\sigma}{1-\sigma}}, \quad \varepsilon_i \le \frac{\sigma^2 d_0^2}{2(1-\sigma^2)\underline{\lambda}\, j}\,.$$

(b) *For any* $j \ge 1$,

$$\overline{v}_j \in T^{\overline{\varepsilon}_j}(\overline{\widetilde{z}}_j), \quad \|\overline{v}_j\| \le \frac{2d_0}{\underline{\lambda}\, j}, \quad \overline{\varepsilon}_j \le \frac{2(1+\sigma/\sqrt{1-\sigma^2})d_0^2}{\underline{\lambda}\, j}\,.$$

Remark.

The (*pointwise* and *ergodic*) bounds given in (a) and (b) of Theorem 2.3 guarantee, respectively, that for given tolerances $\rho, \epsilon > 0$, the termination criterion (17) is satisfied in at most

$$\mathcal{O}\left(\max\left\{\frac{d_0^2}{\underline{\lambda}^2\rho^2}, \frac{d_0^2}{\underline{\lambda}\epsilon}\right\}\right) \text{ and } \mathcal{O}\left(\max\left\{\frac{d_0}{\underline{\lambda}\rho}, \frac{d_0^2}{\underline{\lambda}\epsilon}\right\}\right)$$

iterations, respectively. We refer the reader to [32] for a complete study of the iteration complexity of the HPE method and its special instances.

The proposition below will be useful in the next sections.

Proposition 2.4 ([32, Lemma 4.2 and Eq. (34)]). *Let $\{z_j\}$ be generated by the* HPE method (Algorithm 1). *Then, for any $z^* \in T^{-1}(0)$, the sequence $\{\|z^* - z_j\|\}$ is nonincreasing. As a consequence, for every $j \geq 1$, we have*

$$\|z_j - z_0\| \leq 2d_0, \tag{21}$$

where d_0 denotes the distance of z_0 to $T^{-1}(0)$.

2.2.1 A HPE variant for strongly monotone sums

We now consider the MIP

$$0 \in S(z) + B(z) =: T(z) \tag{22}$$

where the following is assumed to hold:

(C1) S and B are maximal monotone operators on $\mathcal{H}$;

(C2) S is (additionally) μ–strongly monotone for some $\mu > 0$, i.e., there exists $\mu > 0$ such that

$$\langle z - z', v - v' \rangle \geq \mu \|z - z'\|^2 \qquad \forall v \in S(z), v' \in S(z'); \tag{23}$$

(C3) the solution set $(S + B)^{-1}(0)$ of (22) is nonempty.

The main motivation to consider the above setting is Subsection 4.1, in which the monotone inclusion (81) is clearly a special instance of (22) with $S(\cdot) := (1/\gamma)(\cdot - \mathring{z})$, which is obviously $(1/\gamma)$-strongly maximal monotone on $\mathcal{H}$.

The algorithm below was proposed and studied (with a different notation) in [1, Algorithm 1].

Algorithm 2. A specialized HPE method for solving strongly monotone inclusions

(0) Let $z_0 \in \mathcal{H}$ and $\sigma \in [0, 1)$ be given and set $j \leftarrow 1$.

(1) Compute $(\tilde{z}_j, v_j, \varepsilon_j) \in \mathcal{H} \times \mathcal{H} \times \mathbb{R}_+$ and $\lambda_j > 0$ such that

$$v_j \in S(\tilde{z}_j) + B^{\varepsilon_j}(\tilde{z}_j), \quad \|\lambda_j v_j + \tilde{z}_j - z_{j-1}\|^2 + 2\lambda_j \varepsilon_j \leq \sigma^2 \|\tilde{z}_j - z_{j-1}\|^2. \tag{24}$$

(2) Define

$$z_j = z_{j-1} - \lambda_j v_j, \tag{25}$$

set $j \leftarrow j + 1$ and go to step 1.

Next proposition will be useful in Subsection 4.1.

Proposition 2.5 ([1, Proposition 2.2]). *Let* $\{\tilde{z}_j\}$, $\{v_j\}$ *and* $\{\varepsilon_j\}$ *be generated by* Algorithm 2*, let* $z^* := (S+B)^{-1}(0)$ *and* $d_0 := \|z_0 - z^*\|$*. Assume that* $\lambda_j \geq \underline{\lambda} > 0$ *for all* $j \geq 1$ *and define*

$$\alpha := \left(\frac{1}{2\underline{\lambda}\mu} + \frac{1}{1-\sigma^2} \right)^{-1} \in (0,1). \tag{26}$$

Then, for all $j \geq 1$*,*

$$\begin{aligned} & v_j \in S(\tilde{z}_j) + B^{\varepsilon_j}(\tilde{z}_j), \\ & \|v_j\| \leq \sqrt{\frac{1+\sigma}{1-\sigma}} \left(\frac{(1-\alpha)^{(j-1)/2}}{\underline{\lambda}} \right) d_0, \\ & \varepsilon_j \leq \frac{\sigma^2}{2(1-\sigma^2)} \left(\frac{(1-\alpha)^{j-1}}{\underline{\lambda}} \right) d_0^2. \end{aligned} \tag{27}$$

Next section presents one of the main contributions of this paper, namely an inexact Douglas-Rachford type method for solving (1) and its iteration complexity analysis.

3 An inexact Douglas-Rachford splitting (DRS) method and its iteration complexity

Consider problem (1), i.e., the problem of finding $z \in \mathcal{H}$ such that

$$0 \in A(z) + B(z) \tag{28}$$

where the following hold:

(D1) A and B are maximal monotone operators on $\mathcal{H}$;

(D2) the solution set $(A+B)^{-1}(0)$ of (28) is nonempty.

In this section, we propose and analyze the iteration complexity of an inexact version of the *Douglas-Rachford splitting* (DRS) *method* [26] for finding approximate solutions of (28) according to the following termination criterion: given tolerances $\rho, \epsilon > 0$, find $a, b, x, y \in \mathcal{H}$ and $\varepsilon_a, \varepsilon_b \geq 0$ such that

$$a \in A^{\varepsilon_a}(y),\ b \in B^{\varepsilon_b}(x), \quad \gamma\|a+b\| = \|x-y\| \leq \rho,\ \varepsilon_a + \varepsilon_b \leq \epsilon, \tag{29}$$

where $\gamma > 0$ is a scaling parameter. Note that if $\rho = \epsilon = 0$ in (29), then $z^* := x = y$ is a solution of (28).

As we mentioned earlier, the algorithm below is motivated by (9)–(11) as well as by the recent work of Eckstein and Yao [21].

Algorithm 3. An inexact Douglas-Rachford splitting method for (28)

(0) Let $z_0 \in \mathcal{H}$, $\gamma > 0$, $\tau_0 > 0$ and $0 < \sigma, \theta < 1$ be given and set $k \leftarrow 1$.

(1) Compute $(x_k, b_k, \varepsilon_{b,\,k}) \in \mathcal{H} \times \mathcal{H} \times \mathbb{R}_+$ such that

$$b_k \in B^{\varepsilon_{b,\,k}}(x_k), \quad \|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma\varepsilon_{b,\,k} \leq \tau_{k-1}. \tag{30}$$

(2) Compute $(y_k, a_k) \in \mathcal{H} \times \mathcal{H}$ such that

$$a_k \in A(y_k), \quad \gamma a_k + y_k = x_k - \gamma b_k. \tag{31}$$

(3) (3.a) If

$$\|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma\varepsilon_{b,k} \leq \sigma^2\|\gamma b_k + y_k - z_{k-1}\|^2, \tag{32}$$

then

$$z_k = z_{k-1} - \gamma(a_k + b_k), \quad \tau_k = \tau_{k-1} \qquad \text{[extragradient step]}. \tag{33}$$

(3.b) Else

$$z_k = z_{k-1}, \quad \tau_k = \theta\,\tau_{k-1} \qquad \text{[null step]}. \tag{34}$$

(4) Set $k \leftarrow k + 1$ and go to step 1.

Remarks.

1. We emphasize that although it has been motivated by [21, Algorithm 3], Algorithm 3 is based on a slightly different mechanism of iteration. Moreover, it also allows for the computation of (x_k, b_k) in (30) in the $\varepsilon_{b,k}$– enlargement of B (it has the property that $B^{\varepsilon_{b,k}}(x) \supset B(x)$ for all $x \in \mathcal{H}$); this will be crucial for the design and iteration complexity analysis of the four-operator splitting method of Section 4. We also mention that, contrary to this work, no iteration complexity analysis is performed in [21].

2. Computation of $(x_k, b_k, \varepsilon_{b,\,k})$ satisfying (30) will depend on the particular instance of the problem (28) under consideration. In Section 4, we will use Algorithm 3 for solving a four-operator splitting monotone inclusion. In this setting, at every iteration $k \geq 1$ of Algorithm 3, called an outer iteration, a Tseng's forward-backward (F-B) splitting type method will be used, as an inner iteration, to solve the (prox) subproblem (30).

3. Whenever the resolvent $J_{\gamma B} = (\gamma B + I)^{-1}$ is computable, then it follows that $(x_k, b_k) := (J_{\gamma B}(z_{k-1}), (z_{k-1} - x_k)/\gamma)$ and $\varepsilon_{b,\,k} := 0$ clearly solve (30). In this case, the left hand side of the inequality in (30) is zero and, as a consequence, the inequality (32) is always satisfied. In particular, (9)–(11) hold, i.e., in this case Algorithm 3 reduces to the (exact) DRS method.

4. In this paper, we assume that the resolvent $J_{\gamma A} = (\gamma A + I)^{-1}$ is computable, which implies

that $(y_k, a_k) := (J_{\gamma A}(x_k - \gamma b_k), (x_k - \gamma b_k - y_k)/\gamma)$ is the demanded pair in (31).

5. Algorithm 3 potentially performs extragradient steps and null steps, depending on the condition (32). It will be shown in Proposition 3.2 that iterations corresponding to extragradient steps reduce to a special instance of the HPE method, in which case pointwise and ergodic iteration complexity results are available in the current literature (see Proposition 3.3). On the other hand, iterations corresponding to the null steps will demand a separate analysis (see Proposition 3.4).

As we mentioned in the latter remark, each iteration of Algorithm 3 is either an extragradient step or a null step (see (33) and (34)). This will be formally specified by considering the sets:

$$\begin{aligned} &\mathcal{A} := \text{indexes } k \geq 1 \text{ for which an extragradient step is executed at the iteration } k. \\ &\mathcal{B} := \text{indexes } k \geq 1 \text{ for which a null step is executed at the iteration } k. \end{aligned} \tag{35}$$

That said, we let

$$\mathcal{A} = \{k_j\}_{j \in J}, \quad J := \{j \geq 1 \mid j \leq \#\mathcal{A}\} \tag{36}$$

where $k_0 := 0$ and $k_0 < k_j < k_{j+1}$ for all $j \in J$, and let $\beta_0 := 0$ and

$$\beta_k := \text{the number of indexes for which a null step is executed until the iteration } k. \tag{37}$$

Note that direct use of the above definition and (34) yield

$$\tau_k = \theta^{\beta_k} \tau_0 \quad \forall k \geq 0. \tag{38}$$

In order to study the *ergodic iteration complexity* of Algorithm 3 we also define the *ergodic sequences* associated to the sequences $\{x_{k_j}\}_{j\in J}$, $\{y_{k_j}\}_{j\in J}$, $\{a_{k_j}\}_{j\in J}$, $\{b_{k_j}\}_{j\in J}$, and $\{\varepsilon_{b,\,k_j}\}_{j\in J}$, for all $j \in J$, as follows:

$$\overline{x}_{k_j} := \frac{1}{j}\sum_{\ell=1}^{j} x_{k_\ell}, \quad \overline{y}_{k_j} := \frac{1}{j}\sum_{\ell=1}^{j} y_{k_\ell}\,, \tag{39}$$

$$\overline{a}_{k_j} := \frac{1}{j}\sum_{\ell=1}^{j} a_{k_\ell}\,, \quad \overline{b}_{k_j} := \frac{1}{j}\sum_{\ell=1}^{j} b_{k_\ell}\,, \tag{40}$$

$$\overline{\varepsilon}_{a,\,k_j} := \frac{1}{j}\sum_{\ell=1}^{j} \langle y_{k_\ell} - \overline{y}_{k_j}, a_{k_\ell} - \overline{a}_{k_j}\rangle = \frac{1}{j}\sum_{\ell=1}^{j} \langle y_{k_\ell} - \overline{y}_{k_j}, a_{k_\ell}\rangle, \tag{41}$$

$$\overline{\varepsilon}_{b,\,k_j} := \frac{1}{j}\sum_{\ell=1}^{j} \left[\varepsilon_{b,\,k_\ell} + \langle x_{k_\ell} - \overline{x}_{k_j}, b_{k_\ell} - \overline{b}_{k_j}\rangle\right] = \frac{1}{j}\sum_{\ell=1}^{j} \left[\varepsilon_{b,\,k_\ell} + \langle x_{k_\ell} - \overline{x}_{k_j}, b_{k_\ell}\rangle\right]. \tag{42}$$

Moreover, the results on iteration complexity of Algorithm 3 (pointwise and ergodic) obtained in this paper will depend on the following quantity:

$$d_{0,\,\gamma} := \operatorname{dist}(z_0, \operatorname{zer}(S_{\gamma,A,B})) = \min\{\|z_0 - z\| \mid z \in \operatorname{zer}(S_{\gamma,A,B})\} \tag{43}$$

which measures the quality of the initial guess z_0 in Algorithm 3 with respect to $\operatorname{zer}(S_{\gamma,A,B})$, where the operator $S_{\gamma,A,B}$ is such that $J_{\gamma B}(\operatorname{zer}(S_{\gamma,\,A,\,B})) = (A+B)^{-1}(0)$ (see (8)).

In the next proposition, we show that the procedure resulting by selecting the extragradient steps in Algorithm 3 can be embedded into HPE method.

First, we need the following lemma.

Lemma 3.1. *Let $\{z_k\}$ be generated by* Algorithm 3 *and let the set J be defined in* (36)*. Then,*

$$z_{k_{j-1}} = z_{k_j - 1} \qquad \forall j \in J. \tag{44}$$

Proof. Using (35) and (36) we have $\{k \geq 1 \,|\, k_{j-1} < k < k_j\} \subset \mathcal{B}$, for all $j \in J$. Consequently, using the definition of $\mathcal{B}$ in (35) and (34) we conclude that $z_k = z_{k_{j-1}}$ whenever $k_{j-1} \leq k < k_j$. As a consequence, we obtain that (44) follows from the fact that $k_{j-1} \leq k_j - 1 < k_j$. □

Proposition 3.2. *Let $\{z_k\}$, $\{(x_k, b_k)\}$, $\{\varepsilon_{b,k}\}$ and $\{(y_k, a_k)\}$ be generated by* Algorithm 3 *and let the operator $S_{\gamma, A, B}$ be defined in* (8)*. Define, for all $j \in J$,*

$$\widetilde{z}_{k_j} := y_{k_j} + \gamma b_{k_j}, \quad v_{k_j} := \gamma(a_{k_j} + b_{k_j}), \quad \varepsilon_{k_j} := \gamma \varepsilon_{b,k_j}. \tag{45}$$

Then, for all $j \in J$,

$$\begin{aligned} &v_{k_j} \in (S_{\gamma, A, B})^{\varepsilon_{k_j}}(\widetilde{z}_{k_j}), \qquad \|v_{k_j} + \widetilde{z}_{k_j} - z_{k_{j-1}}\|^2 + 2\varepsilon_{k_j} \leq \sigma^2 \|\widetilde{z}_{k_j} - z_{k_{j-1}}\|^2, \\ &z_{k_j} = z_{k_{j-1}} - v_{k_j}. \end{aligned} \tag{46}$$

As a consequence, the sequences $\{\widetilde{z}_{k_j}\}_{j \in J}$, $\{v_{k_j}\}_{j \in J}$, $\{\varepsilon_{k_j}\}_{j \in J}$ and $\{z_{k_j}\}_{j \in J}$ are generated by Algorithm 1 *with $\lambda_j \equiv 1$ for solving* (14) *with $T := S_{\gamma, A, B}$.*

Proof. For any $(z', v') := (y + \gamma b, \gamma a + \gamma b) \in S_{\gamma, A, B}$ we have, in particular, $b \in B(x)$ and $a \in A(y)$ (see (8)). Using these inclusions, the inclusions in (30) and (31), the monotonicity of the operator A and (13) with $T = B$ we obtain

$$\langle x_{k_j} - x, b_{k_j} - b \rangle \geq -\varepsilon_{b,k_j}, \qquad \langle y_{k_j} - y, a_{k_j} - a \rangle \geq 0. \tag{47}$$

Moreover, using the identity in (31) and the corresponding one in (8) we find

$$(y_{k_j} - y) + \gamma(b_{k_j} - b) = (x_{k_j} - x) - \gamma(a_{k_j} - a). \tag{48}$$

Using (45), (47) and (48) we have

$$\begin{aligned} \langle \widetilde{z}_{k_j} - z', v_{k_j} - v' \rangle &= \langle (y_{k_j} + \gamma b_{k_j}) - (y + \gamma b), (\gamma a_{k_j} + \gamma b_{k_j}) - (\gamma a + \gamma b) \rangle \\ &= \langle y_{k_j} - y + \gamma(b_{k_j} - b), \gamma(a_{k_j} - a) + \gamma(b_{k_j} - b) \rangle \\ &= \gamma \langle y_{k_j} - y + \gamma(b_{k_j} - b), a_{k_j} - a \rangle + \gamma \langle y_{k_j} - y + \gamma(b_{k_j} - b), b_{k_j} - b \rangle \\ &= \gamma \langle y_{k_j} - y + \gamma(b_{k_j} - b), a_{k_j} - a \rangle + \gamma \langle x_{k_j} - x - \gamma(a_{k_j} - a), b_{k_j} - b \rangle \\ &= \gamma \langle y_{k_j} - y, a_{k_j} - a \rangle + \gamma \langle x_{k_j} - x, b_{k_j} - b \rangle \\ &\geq \gamma \langle x_{k_j} - x, b_{k_j} - b \rangle \\ &\geq -\varepsilon_{k_j}, \end{aligned}$$

which combined with definition (13) gives the inclusion in (46).

From (45), (44), the identity in (31) and (32) we also obtain

$$\begin{aligned}\|v_{k_j}+\widetilde{z}_{k_j}-z_{k_{j-1}}\|^2 &= \|\gamma(a_{k_j}+b_{k_j})+(y_{k_j}+\gamma b_{k_j})-z_{k_j-1}\|^2\\ &= \|(x_{k_j}-y_{k_j})+(y_{k_j}+\gamma b_{k_j})-z_{k_j-1}\|^2\\ &= \|\gamma b_{k_j}+x_{k_j}-z_{k_j-1}\|^2\\ &\le \sigma^2\|\gamma b_{k_j}+y_{k_j}-z_{k_j-1}\|^2-2\gamma\varepsilon_{b,k_j}\\ &= \sigma^2\|\widetilde{z}_{k_j}-z_{k_{j-1}}\|^2-2\varepsilon_{k_j},\end{aligned}$$

which gives the inequality in (46). To finish the proof of (46), note that the desired identity in (46) follows from the first one in (33), the second one in (45) and (44). The last statement of the proposition follows from (45), (46) and Algorithm 1's definition. □

Proposition 3.3. (rate of convergence for extragradient steps) *Let* $\{(x_k, b_k)\}$, $\{(y_k, a_k)\}$ *and* $\{\varepsilon_{b,\,k}\}$ *be generated by* Algorithm 3 *and consider the ergodic sequences defined in* (39)–(42). *Let* $d_{0,\gamma}$ *and the set* J *be defined in* (43) *and* (36), *respectively. Then,*

(a) *For any* $j \in J$, *there exists* $i \in \{1,\dots,j\}$ *such that*

$$a_{k_i}\in A(y_{k_i}),\quad b_{k_i}\in B^{\varepsilon_{b,\,k_i}}(x_{k_i}), \tag{49}$$

$$\gamma\|a_{k_i}+b_{k_i}\|=\|x_{k_i}-y_{k_i}\|\le\frac{d_{0,\gamma}}{\sqrt{j}}\sqrt{\frac{1+\sigma}{1-\sigma}}, \tag{50}$$

$$\varepsilon_{b,\,k_i}\le\frac{\sigma^2 d_{0,\gamma}^2}{2\gamma(1-\sigma^2)j}\,. \tag{51}$$

(b) *For any* $j \in J$,

$$\overline{a}_{k_j}\in A^{\overline{\varepsilon}_{a,k_j}}(\overline{y}_{k_j}),\quad \overline{b}_{k_j}\in B^{\overline{\varepsilon}_{b,\,k_j}}(\overline{x}_{k_j}), \tag{52}$$

$$\gamma\|\overline{a}_{k_j}+\overline{b}_{k_j}\|=\|\overline{x}_{k_j}-\overline{y}_{k_j}\|\le\frac{2d_{0,\gamma}}{j}, \tag{53}$$

$$\overline{\varepsilon}_{a,\,k_j}+\overline{\varepsilon}_{b,\,k_j}\le\frac{2(1+\sigma/\sqrt{1-\sigma^2})d_{0,\gamma}^2}{\gamma j}\,. \tag{54}$$

Proof. Note first that (49) follow from the inclusions in (30) and (31). Using the last statement in Proposition 3.2, Theorem 2.3 (with $\underline{\lambda}=1$) and (43) we obtain that there exists $i \in \{1,\dots,j\}$ such that

$$\|v_{k_i}\|\le\frac{d_{0,\gamma}}{\sqrt{j}}\sqrt{\frac{1+\sigma}{1-\sigma}},\quad \varepsilon_{k_i}\le\frac{\sigma^2 d_{0,\gamma}^2}{2(1-\sigma^2)j}\,, \tag{55}$$

which, in turn, combined with the identity in (31) and the definitions of v_{k_i} and ε_{k_i} in (45) gives the desired inequalities in (50) and (51) (concluding the proof of (a)) and

$$\|\overline{v}_j\|\le\frac{2d_{0,\gamma}}{j},\quad \overline{\varepsilon}_j\le\frac{2(1+\sigma/\sqrt{1-\sigma^2})d_{0,\gamma}^2}{j}\,, \tag{56}$$

where $\overline{v}_j$ and $\overline{\varepsilon}_j$ are defined in (19) and (20), respectively, with $\Lambda_j = j$ and

$$\lambda_\ell := 1, \quad v_\ell := v_{k_\ell}, \quad \varepsilon_\ell := \varepsilon_{k_\ell}, \quad \tilde{z}_\ell := \widetilde{z}_{k_\ell} \qquad \forall \ell = 1, \dots, j. \tag{57}$$

Since the inclusions in (52) are a direct consequence of the ones in (30) and (31), Proposition 2.1(d), (39)–(42) and Theorem 2.2, it follows from (53), (54) and (56) that to finish the proof of (b), it suffices to prove that

$$\overline{v_j} = \gamma(\overline{a}_{k_j} + \overline{b}_{k_j}), \quad \gamma(\overline{a}_{k_j} + \overline{b}_{k_j}) = \overline{x}_{k_j} - \overline{y}_{k_j}, \qquad \overline{\varepsilon}_j = \gamma(\overline{\varepsilon}_{a,\,k_j} + \overline{\varepsilon}_{b,\,k_j}). \tag{58}$$

The first identity in (58) follows from (57), the second identities in (19) and (45), and (40). On the other hand, from (31) we have $\gamma(a_{k_\ell} + b_{k_\ell}) = x_{k_\ell} - y_{k_\ell}$, for all $\ell = 1, \dots, j$, which combined with (39) and (40) gives the second identity in (58). Using the latter identity and the second one in (58) we obtain

$$(y_{k_\ell} - \overline{y}_{k_j}) + \gamma(b_{k_\ell} - \overline{b}_{k_j}) = (x_{k_\ell} - \overline{x}_{k_j}) - \gamma(a_{k_\ell} - \overline{a}_{k_j}) \qquad \forall \ell = 1, \dots, j. \tag{59}$$

Moreover, it follows from (19), (57), the first identity in (45), (39) and (40) that

$$\overline{\tilde{z}}_j = \overline{\tilde{z}}_{k_j} = \frac{1}{j}\sum_{\ell=1}^{j} (y_{k_\ell} + \gamma b_{k_\ell}) = \overline{y}_{k_j} + \gamma \overline{b}_{k_j}. \tag{60}$$

Using (60), (57), (45) and (59) we obtain, for all $\ell = 1, \dots, j$,

$$\begin{aligned}
\langle \tilde{z}_\ell - \overline{\tilde{z}}_j, v_\ell \rangle &= \langle (y_{k_\ell} + \gamma b_{k_\ell}) - (\overline{y}_{k_j} + \gamma \overline{b}_{k_j}), \gamma(a_{k_\ell} + b_{k_\ell}) \rangle \\
&= \gamma \langle (y_{k_\ell} - \overline{y}_{k_j}) + \gamma(b_{k_\ell} - \overline{b}_{k_j}), a_{k_\ell} \rangle + \gamma \langle (y_{k_\ell} - \overline{y}_{k_j}) + \gamma(b_{k_\ell} - \overline{b}_{k_j}), b_{k_\ell} \rangle \\
&= \gamma \langle (y_{k_\ell} - \overline{y}_{k_j}) + \gamma(b_{k_\ell} - \overline{b}_{k_j}), a_{k_\ell} \rangle + \gamma \langle (x_{k_\ell} - \overline{x}_{k_j}) - \gamma(a_{k_\ell} - \overline{a}_{k_j}), b_{k_\ell} \rangle \\
&= \gamma \langle y_{k_\ell} - \overline{y}_{k_j}, a_{k_\ell} \rangle + \gamma^2 \langle b_{k_\ell} - \overline{b}_{k_j}, a_{k_\ell} \rangle + \gamma \langle x_{k_\ell} - \overline{x}_{k_j}, b_{k_\ell} \rangle - \gamma^2 \langle a_{k_\ell} - \overline{a}_{k_j}, b_{k_\ell} \rangle,
\end{aligned}$$

which combined with (20), (57), (41) and (42) yields

$$\begin{aligned}
\overline{\varepsilon}_j = \frac{1}{j}\sum_{\ell=1}^{j} \left[\varepsilon_\ell + \langle \tilde{z}_\ell - \overline{\tilde{z}}_j, v_\ell \rangle\right] &= \frac{1}{j}\sum_{\ell=1}^{j} \gamma \left[\varepsilon_{b,\,k_\ell} + \langle x_{k_\ell} - \overline{x}_{k_j}, b_{k_\ell} \rangle + \langle y_{k_\ell} - \overline{y}_{k_j}, a_{k_\ell} \rangle\right] \\
&= \gamma(\overline{\varepsilon}_{a,\,k_j} + \overline{\varepsilon}_{b,\,k_j}),
\end{aligned}$$

which is exactly the last identity in (58). This finishes the proof. □

Proposition 3.4. (rate of convergence for null steps) *Let $\{(x_k, b_k)\}$, $\{(y_k, a_k)\}$ and $\{\varepsilon_{b,k}\}$ be generated by* Algorithm 3. *Let $\{\beta_k\}$ and the set $\mathcal{B}$ be defined in* (37) *and* (35)*, respectively. Then, for $k \in \mathcal{B}$,*

$$a_k \in A(y_k), \quad b_k \in B^{\varepsilon_{b,\,k}}(x_k), \tag{61}$$

$$\gamma \|a_k + b_k\| = \|x_k - y_k\| \le \frac{2\sqrt{\tau_0}}{\sigma}\, \theta^{\frac{\beta_{k-1}}{2}}, \tag{62}$$

$$\gamma \varepsilon_{b,\,k} \le \frac{\tau_0}{2}\, \theta^{\beta_{k-1}}. \tag{63}$$

Proof. Note first that (61) follows from (30) and (31). Using (35), (30) and Step 3.b's definition (see Algorithm 3) we obtain

$$\tau_{k-1} \geq \|\underbrace{\gamma b_k + x_k - z_{k-1}}_{p_k}\|^2 + 2\gamma\varepsilon_{b,k} > \sigma^2\|\underbrace{\gamma b_k + y_k - z_{k-1}}_{q_k}\|^2,$$

which, in particular, gives

$$\gamma\varepsilon_{b,\,k} \leq \frac{\tau_{k-1}}{2}, \tag{64}$$

and combined with the identity in (31) yields,

$$\begin{aligned} \gamma\|a_k + b_k\| = \|x_k - y_k\| &= \|p_k - q_k\| \\ &\leq \|p_k\| + \|q_k\| \\ &\leq \left(1 + \frac{1}{\sigma}\right)\sqrt{\tau_{k-1}}. \end{aligned} \tag{65}$$

To finish the proof, use (64), (65) and (38). □

Next we present the main results regarding the pointwise and ergodic iteration complexity of Algorithm 3 for finding approximate solutions of (28) satisfying the termination criterion (29). While Theorem 3.5 is a consequence of Proposition 3.3(a) and Proposition 3.4, the ergodic iteration complexity of Algorithm 3, namely Theorem 3.6, follows by combining the latter proposition and Proposition 3.3(b). Since the proof of Theorem 3.6 follows the same outline of Theorem 3.5's proof, it will be omitted.

Theorem 3.5. (pointwise iteration complexity of Algorithm 3) *Assume that* $\max\{(1-\sigma)^{-1}, \sigma^{-1}\} = \mathcal{O}(1)$ *and let* $d_{0,\gamma}$ *be as in* (43). *Then, for given tolerances* $\rho, \epsilon > 0$, Algorithm 3 *finds* $a, b, x, y \in \mathcal{H}$ *and* $\varepsilon_b \geq 0$ *such that*

$$a \in A(y),\ b \in B^{\varepsilon_b}(x), \quad \gamma\|a + b\| = \|x - y\| \leq \rho, \quad \varepsilon_b \leq \epsilon \tag{66}$$

after performing at most

$$\mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}^2}{\rho^2}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\}\right) \tag{67}$$

extragradient steps *and*

$$\mathcal{O}\left(1 + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{68}$$

null steps. *As a consequence, under the above assumptions,* Algorithm 3 *terminates with* $a, b, x, y \in \mathcal{H}$ *and* $\varepsilon_b \geq 0$ *satisfying* (66) *in at most*

$$\mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}^2}{\rho^2}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\} + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{69}$$

iterations.

Proof. Let $\mathcal{A}$ be as in (35) and consider the cases:

$$\#\mathcal{A} \geq M_{\text{ext}} := \left\lceil \max\left\{ \frac{2\, d_{0,\gamma}^2}{(1-\sigma)\rho^2}, \frac{\sigma^2 d_{0,\gamma}^2}{2\gamma(1-\sigma^2)\epsilon} \right\} \right\rceil \quad \text{and} \quad \#\mathcal{A} < M_{\text{ext}}. \tag{70}$$

In the first case, the desired bound (67) on the number of extragradient steps to find $a, b, x, y \in \mathcal{H}$ and $\varepsilon_b \geq 0$ satisfying (66) follows from the definition of J in (36) and Proposition 3.3(a).

On the other hand, in the second case, i.e., $\#\mathcal{A} < M_{\text{ext}}$, the desired bound (68) is a direct consequence of Proposition 3.4. The last statement of the theorem follows from (67) and (68). □

Next is the main result on the ergodic iteration complexity of Algorithm 3. As mentioned before, its proof follows the same outline of Theorem 3.5's proof, now applying Proposition 3.3(b) instead of the item (a) of the latter proposition.

Theorem 3.6. (ergodic iteration complexity of Algorithm 3) *For given tolerances $\rho, \epsilon > 0$, under the same assumptions of* Theorem 3.5*,* Algorithm 3 *provides $a, b, x, y \in \mathcal{H}$ and $\varepsilon_a, \varepsilon_b \geq 0$ such that*

$$a \in A^{\varepsilon_a}(y),\; b \in B^{\varepsilon_b}(x), \quad \gamma\|a + b\| = \|x - y\| \leq \rho, \quad \varepsilon_a + \varepsilon_b \leq \epsilon. \tag{71}$$

after performing at most

$$\mathcal{O}\left(1 + \max\left\{ \frac{d_{0,\gamma}}{\rho}, \frac{d_{0,\gamma}^2}{\gamma\epsilon} \right\}\right) \tag{72}$$

extragradient steps *and*

$$\mathcal{O}\left(1 + \max\left\{ \log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right) \right\}\right) \tag{73}$$

null steps. *As a consequence, under the above assumptions,* Algorithm 3 *terminates with $a, b, x, y \in \mathcal{H}$ and $\varepsilon_a, \varepsilon_b \geq 0$ satisfying* (71) *in at most*

$$\mathcal{O}\left(1 + \max\left\{ \frac{d_{0,\gamma}}{\rho}, \frac{d_{0,\gamma}^2}{\gamma\epsilon} \right\} + \max\left\{ \log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right) \right\}\right) \tag{74}$$

iterations.

Proof. The proof follows the same outline of Theorem 3.5's proof, now applying Proposition 3.3(b) instead of Proposition 3.3(a). □

Remarks.

1. Theorem 3.6 ensures that for given tolerances $\rho, \epsilon > 0$, up to an additive logarithmic factor, Algorithm 3 requires no more than

$$\mathcal{O}\left(1 + \max\left\{ \frac{d_{0,\gamma}}{\rho}, \frac{d_{0,\gamma}^2}{\gamma\epsilon} \right\}\right)$$

   iterations to find an approximate solution of the monotone inclusion problem (28) according to the termination criterion (29).

2. While the (ergodic) upper bound on the number of iterations provided in (74) is better than the corresponding one in (69) (in terms of the dependence on the tolerance $\rho > 0$) by a factor of $\mathcal{O}(1/\rho)$, the inclusion in (71) is potentially weaker than the corresponding one in (66), since one may have $\varepsilon_a > 0$ in (71), and the set $A^{\varepsilon_a}(y)$ is in general larger than $A(y)$.

3. Iteration complexity results similar to the ones in Proposition 3.3 were recently obtained for a relaxed Peaceman-Rachford method in [34] . We emphasize that, in contrast to this work, the latter reference considers only the case where the resolvents $J_{\gamma A}$ and $J_{\gamma B}$ of A and B, respectively, are both computable.

The proposition below will be important in the next section.

Proposition 3.7. *Let* $\{z_k\}$ *be generated by* Algorithm 3 *and* $d_{0,\gamma}$ *be as in* (43)*. Then,*

$$\|z_k - z_0\| \leq 2d_{0,\gamma} \quad \forall k \geq 1. \tag{75}$$

Proof. Note that (i) if $k = k_j \in \mathcal{A}$, for some $j \in J$, see (36), then (75) follows from the last statement in Proposition 3.2 and Proposition 2.4; (ii) if $k \in \mathcal{B}$, from the first identity in (34), see (35), we find that either $z_k = z_0$, in which case (75) holds trivially, or $z_k = z_{k_j}$ for some $j \in J$, in which case the results follows from (i). □

4 A Douglas-Rachford-Tseng's forward-backward (F-B) four-operator splitting method

In this section, we consider problem (2), i.e., the problem of finding $z \in \mathcal{H}$ such that

$$0 \in A(z) + C(z) + F_1(z) + F_2(z) \tag{76}$$

where the following hold:

(E1) A and C are (set-valued) maximal monotone operators on $\mathcal{H}$.

(E2) $F_1 : D(F_1) \subset \mathcal{H} \to \mathcal{H}$ is monotone and L-Lipschitz continuous on a (nonempty) closed convex set Ω such that $D(C) \subset \Omega \subset D(F_1)$, i.e., F_1 is monotone on Ω and there exists $L \geq 0$ such that

$$\|F_1(z) - F_1(z')\| \leq L\|z - z'\| \qquad \forall z, z' \in \Omega. \tag{77}$$

(E3) $F_2 : \mathcal{H} \to \mathcal{H}$ is $\eta-$cocoercivo, i.e., there exists $\eta > 0$ such that

$$\langle F_2(z) - F_2(z'), z - z' \rangle \geq \eta \|F_2(z) - F_2(z')\|^2 \qquad \forall z, z' \in \mathcal{H}. \tag{78}$$

(E4) $B^{-1}(0)$ is nonempty, where

$$B := C + F_1 + F_2. \tag{79}$$

(E5) The solution set of (76) is nonempty.

Aiming at solving the monotone inclusion (76), we present and study the iteration complexity of a (four-operator) splitting method which combines Algorithm 3 (used as an outer iteration) and a Tseng's forward-backward (F-B) splitting type method (used as an inner iteration for solving, for each outer iteration, the prox subproblems in (30)). We prove results on pointwise and ergodic iteration complexity of the proposed four-operator splitting algorithm by analyzing it in the framework of Algorithm 3 for solving (28) with B as in (79) and under assumptions (E1)–(E5). The (outer) iteration complexities will follow from results on pointwise and ergodic iteration complexities of Algorithm 3, obtained in Section 3, while the computation of an upper bound on the overall number of inner iterations required to achieve prescribed tolerances will require a separate analysis. Still regarding the results on iteration complexity, we mention that we consider the following notion of approximate solution for (76): given tolerances $\rho, \epsilon > 0$, find $a, b, x, y \in \mathcal{H}$ and $\varepsilon_a, \varepsilon_b \geq 0$ such that

$$\begin{aligned} & a \in A^{\varepsilon_a}(y), \\ & \text{either } \; b \in C(x) + F_1(x) + F_2^{\varepsilon_b}(x) \; \text{ or } \; b \in (C + F_1 + F_2)^{\varepsilon_b}(x), \\ & \gamma\|a + b\| = \|x - y\| \leq \rho, \;\; \varepsilon_a + \varepsilon_b \leq \epsilon, \end{aligned} \tag{80}$$

where $\gamma > 0$. Note that (i) for $\rho = \epsilon = 0$, the above conditions imply that $z^* := x = y$ is a solution of the monotone inclusion (76); (ii) the second inclusion in (80), which will appear in the ergodic iteration complexity, is potentially weaker than the first one (see Proposition 2.1(b)), which will appear in the corresponding pointwise iteration complexity of the proposed method.

We also mention that problem (76) falls in the framework of the monotone inclusion (28) due to the facts that, in view of assumptions (E1), (E2) and (E3), the operator A is maximal monotone, and the operator $F_1 + F_2$ is monotone and $(L + 1/\eta)$–Lipschitz continuous on the closed convex set $\Omega \supset D(C)$, which combined with the assumption on the operator C in (E1) and with [31, Proposition A.1] implies that the operator B defined in (79) is maximal monotone as well. These facts combined with assumption (E5) give that conditions (D1) and (D2) of Section 3 hold for A and B as in (E1) and (79), respectively. In particular, it gives that Algorithm 3 may be applied to solve the four-operator monotone inclusion (76).

In this regard, we emphasize that any implementation of Algorithm 3 will heavily depend on specific strategies for solving each subproblem in (30), since (y_k, a_k) required in (31) can be computed by using the resolvent operator of A, available in closed form in many important cases. In the next subsection, we show how the specific structure (76) allows for an application of a Tseng's F-B splitting type method for solving each subproblem in (30).

4.1 Solving the subproblems in (30) for B as in (79)

In this subsection, we present and study a Tseng's F-B splitting type method [2, 8, 31, 42] for solving the corresponding proximal subproblem in (30) at each (outer) iteration of Algorithm 3, when used to solve (76). To begin with, first consider the (strongly) monotone inclusion

$$0 \in B(z) + \frac{1}{\gamma}(z - \mathring{z}) \tag{81}$$

where B is as in (79), $\gamma > 0$ and $\mathring{z} \in \mathcal{H}$, and note that the task of finding $(x_k, b_k, \varepsilon_{b,k})$ satisfying (30) is related to the task of solving (81) with $\mathring{z} := z_{k-1}$.

In the remaining part of this subsection, we present and study a Tseng's F-B splitting type method for solving (81). As we have mentioned before, the resulting algorithm will be used as an inner procedure for solving the subproblems (30) at each iteration of Algorithm 3, when applied to solve (76).

Algorithm 4. A Tseng's F-B splitting type method for (81)

Input: C, F_1, Ω, L, F_2 and η as in conditions (E1)–(E5), $\mathring{z} \in \mathcal{H}$, $\mathring{\tau} > 0$, $\sigma \in (0,1)$ and γ such that

$$0 < \gamma \leq \frac{4\eta\sigma^2}{1 + \sqrt{1 + 16L^2\eta^2\sigma^2}}. \tag{82}$$

(0) Set $z_0 \leftarrow \mathring{z}$ and $j \leftarrow 1$.

(1) Let $z'_{j-1} \leftarrow P_\Omega(z_{j-1})$ and compute

$$\begin{aligned} \widetilde{z}_j &= \left(\frac{\gamma}{2}C + I\right)^{-1}\left(\frac{\mathring{z} + z_{j-1} - \gamma(F_1 + F_2)(z'_{j-1})}{2}\right), \\ z_j &= \widetilde{z}_j - \gamma\left(F_1(\widetilde{z}_j) - F_1(z'_{j-1})\right). \end{aligned} \tag{83}$$

(2) If

$$\|z_{j-1} - z_j\|^2 + \frac{\gamma\|z'_{j-1} - \widetilde{z}_j\|^2}{2\eta} \leq \mathring{\tau}, \tag{84}$$

then **terminate**. Otherwise, set $j \leftarrow j + 1$ and go to step 1.

Output: $(z_{j-1}, z'_{j-1}, z_j, \widetilde{z}_j)$.

Remark.

Algorithm 4 combines ideas from the standard Tseng's F-B splitting algorithm [42] as well as from recent insights on the convergence and iteration complexity of some variants the latter method [1, 8, 31]. In this regard, evaluating the cocoercive component F_2 just once per iteration (see [8, Theorem 1]) is potentially important in many applications, where the evaluation of cocoercive operators is in general computationally expensive (see [8] for a discussion). Nevertheless, we emphasize that the results obtained in this paper regarding the analysis of Algorithm 4 do not follow from any of the just mentioned references.

Next corollary ensures that Algorithm 4 always terminates with the desired output.

Corollary 4.1. *Assume that* $(1 - \sigma^2)^{-1} = \mathcal{O}(1)$ *and let* $d_{\mathring{z},b}$ *denote the distance of* $\mathring{z}$ *to* $B^{-1}(0) \neq \emptyset$. *Then,* Algorithm 4 *terminates with the desired output after performing no more than*

$$\mathcal{O}\left(1 + \log^+\left(\frac{d_{\mathring{z},b}}{\sqrt{\mathring{\tau}}}\right)\right) \tag{85}$$

iterations.

Proof. See Subsection 4.3. □

4.2 A Douglas-Rachford-Tseng's F-B four-operator splitting method

In this subsection, we present and study the iteration complexity of the main algorithm in this work, for solving (76), namely Algorithm 5, which combines Algorithm 3, used as an outer iteration, and Algorithm 4, used as an inner iteration, for solving the corresponding subproblem in (30). Algorithm 5 will be shown to be a special instance of Algorithm 3, for which pointwise and ergodic iteration complexity results are available in Section 3. Corollary 4.1 will be specially important to compute a bound on the total number of inner iterations performed by Algorithm 5 to achieve prescribed tolerances.

Algorithm 5. A Douglas-Rachford-Tseng's F-B splitting type method for (76)

(0) Let $z_0 \in \mathcal{H}$, $\tau_0 > 0$ and $0 < \sigma, \theta < 1$ be given, let C, F_1, Ω, L, F_2 and η as in conditions (E1)–(E5) and γ satisfying condition (82), and set $k \leftarrow 1$.

(1) Call Algorithm 4 with inputs C, F_1, Ω, L, F_2 and η, $(\mathring{z}, \mathring{\tau}) := (z_{k-1}, \tau_{k-1})$, σ and γ to obtain as output $(z_{j-1}, z'_{j-1}, z_j, \widetilde{z}_j)$, and set

$$x_k = \widetilde{z}_j, \qquad b_k = \frac{z_{k-1} + z_{j-1} - (z_j + \widetilde{z}_j)}{\gamma}, \qquad \varepsilon_{b,\,k} = \frac{\|z'_{j-1} - \widetilde{z}_j\|^2}{4\eta}. \tag{86}$$

(2) Compute $(y_k, a_k) \in \mathcal{H} \times \mathcal{H}$ such that

$$a_k \in A(y_k), \quad \gamma a_k + y_k = x_k - \gamma b_k. \tag{87}$$

(3) (3.a) If

$$\|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma \varepsilon_{b,k} \leq \sigma^2 \|\gamma b_k + y_k - z_{k-1}\|^2, \tag{88}$$

then

$$z_k = z_{k-1} - \gamma(a_k + b_k), \quad \tau_k = \tau_{k-1} \qquad \text{[extragradient step]}. \tag{89}$$

(3.b) Else

$$z_k = z_{k-1}, \quad \tau_k = \theta\, \tau_{k-1} \qquad \text{[null step]}. \tag{90}$$

(4) Set $k \leftarrow k + 1$ and go to step 1.

In what follows we present the pointwise and ergodic iteration complexities of Algorithm 5 for solving the four-operator monotone inclusion problem (76). The results will follow essentially from the corresponding ones for Algorithm 3 previously obtained in Section 3. On the other hand, bounds on the number of inner iterations executed before achieving prescribed tolerances will be proved by using Corollary 4.1.

We start by showing that Algorithm 5 is a special instance of Algorithm 3.

Proposition 4.2. *The triple* $(x_k, b_k, \varepsilon_{b,k})$ *in* (86) *satisfies condition* (30) *in* Step 1 *of* Algorithm 3*, i.e.,*

$$b_k \in C(x_k) + F_1(x_k) + F_2^{\varepsilon_{b,k}}(x_k) \subset B^{\varepsilon_{b,k}}(x_k), \quad \|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma\varepsilon_{b,k} \le \tau_{k-1}, \tag{91}$$

where B is as in (79)*. As a consequence,* Algorithm 5 *is a special instance of* Algorithm 3 *for solving* (28) *with B as in* (79)*.*

Proof. Using the first identity in (103), the definition of b_k in (86) as well as the fact that $\mathring{z} := z_{k-1}$ in Step 1 of Algorithm 5 we find

$$b_k = v_j - \frac{1}{\gamma}(\widetilde{z}_j - z_{k-1}) = v_j - \frac{1}{\gamma}(\widetilde{z}_j - \mathring{z}). \tag{92}$$

Combining the latter identity with the second inclusion in (104), the second identity in (103) and the definitions of x_k and $\varepsilon_{b,k}$ in (86) we obtain the first inclusion in (91). The second desired inclusion follows from (79) and Proposition 2.1(b). To finish the proof of (91), note that from the first identity in (92), the definitions of x_k and $\varepsilon_{b,k}$ in (86), the definition of v_j in (103) and (84) we have

$$\|\gamma b_k + x_k - z_{k-1}\|^2 + 2\gamma\varepsilon_{b,k} = \|z_{j-1} - z_j\|^2 + \frac{\gamma\|z'_{j-1} - \widetilde{z}_j\|^2}{2\eta} \le \mathring{\tau} = \tau_{k-1}, \tag{93}$$

which gives the inequality in (91). The last statement of the proposition follows from (91), (30)–(34) and (87)–(90). □

Theorem 4.3. (pointwise iteration complexity of Algorithm 5) *Let the operator B and $d_{0,\gamma}$ be as in* (79) *and* (43)*, respectively, and assume that* $\max\{(1-\sigma)^{-1}, \sigma^{-1}\} = \mathcal{O}(1)$*. Let also $d_{0,b}$ be the distance of z_0 to $B^{-1}(0) \neq \emptyset$. Then, for given tolerances $\rho, \epsilon > 0$, the following hold:*

(a) Algorithm 5 *finds $a, b, x, y \in \mathcal{H}$ and $\varepsilon_b \ge 0$ such that*

$$a \in A(y),\ b \in C(x) + F_1(x) + F_2^{\varepsilon_b}(x), \quad \gamma\|a + b\| = \|x - y\| \le \rho,\ \varepsilon_b \le \epsilon \tag{94}$$

after performing no more than

$$k_{\text{p; outer}} := \mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}^2}{\rho^2}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\} + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{95}$$

outer iterations.

(b) *Before achieving the desired tolerance $\rho, \epsilon > 0$, each iteration of* Algorithm 5 *performs at most*

$$k_{\text{inner}} := \mathcal{O}\left(1 + \log^+\left(\frac{d_{0,\gamma} + d_{0,b}}{\sqrt{\tau_0}}\right) + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{96}$$

inner iterations; and hence evaluations of the η–cocoercive operator F_2.

As a consequence, Algorithm 5 *finds $a, b, x, y \in \mathcal{H}$ and $\varepsilon_b \ge 0$ satisfying* (94) *after performing no more than $k_{\text{p; outer}} \times k_{\text{inner}}$ inner iterations.*

Proof. (a) The desired result is a direct consequence of the last statements in Proposition 4.2 and Theorem 3.5, and the inclusions in (91).

(b) Using Step 1's definition and Corollary 4.1 we conclude that, at each iteration $k \geq 1$ of Algorithm 5, the number of inner iterations is bounded by

$$\mathcal{O}\left(1 + \log^+\left(\frac{d_{z_{k-1},b}}{\sqrt{\tau_{k-1}}}\right)\right) \tag{97}$$

where $d_{z_{k-1},b}$ denotes the distance of z_{k-1} to $B^{-1}(0)$. Now, using the last statements in Propositions 4.2 and 3.2, Proposition 2.4 and a simple argument based on the triangle inequality we obtain

$$d_{z_{k-1},b} \leq 2d_{0,\gamma} + d_{0,b} \qquad \forall k \geq 1. \tag{98}$$

By combining (97) and (98) and using (38) we find that, at every iteration $k \geq 1$, the number of inner iterations is bounded by

$$\mathcal{O}\left(1 + \log^+\left(\frac{d_{0,\gamma} + d_{0,b}}{\sqrt{\theta^{\beta_{k-1}}\tau_0}}\right)\right) = \mathcal{O}\left(1 + \log^+\left(\frac{d_{0,\gamma} + d_{0,b}}{\sqrt{\tau_0}}\right) + \beta_{k-1}\right). \tag{99}$$

Using the latter bound, the last statement in Proposition 4.2, the bound on the number of null steps of Algorithm 3 given in Theorem 3.5, and (37) we conclude that, before achieving the prescribed tolerance $\rho, \epsilon > 0$, each iteration Algorithm 5 performs at most the number of iterations given in (96). This concludes the proof of (b).

To finish the proof, note that the last statement of the theorem follows directly from (a) and (b). □

Theorem 4.4. (ergodic iteration complexity of Algorithm 5) *For given tolerances $\rho, \epsilon > 0$, under the same assumptions of* Theorem 4.3 *the following hold:*

(a) Algorithm 5 *provides $a, b, x, y \in \mathcal{H}$ and $\varepsilon_a, \varepsilon_b \geq 0$ such that*

$$a \in A^{\varepsilon_a}(y),\ b \in (C + F_1 + F_2)^{\varepsilon_b}(x), \quad \gamma\|a + b\| = \|x - y\| \leq \rho,\ \varepsilon_a + \varepsilon_b \leq \epsilon \tag{100}$$

after performing no more than

$$k_{\text{e; outer}} := \mathcal{O}\left(1 + \max\left\{\frac{d_{0,\gamma}}{\rho}, \frac{d_{0,\gamma}^2}{\gamma\epsilon}\right\} + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{101}$$

outer iterations.

(b) *Before achieving the desired tolerance $\rho, \epsilon > 0$, each iteration of* Algorithm 5 *performs at most*

$$k_{\text{inner}} := \mathcal{O}\left(1 + \log^+\left(\frac{d_{0,\gamma} + d_{0,b}}{\sqrt{\tau_0}}\right) + \max\left\{\log^+\left(\frac{\sqrt{\tau_0}}{\rho}\right), \log^+\left(\frac{\tau_0}{\gamma\epsilon}\right)\right\}\right) \tag{102}$$

inner iterations; and hence evaluations of the η–cocoercive operator F_2.

As a consequence, Algorithm 5 *provides $a, b, x, y \in \mathcal{H}$ and $\varepsilon_b \geq 0$ satisfying* (100) *after performing no more than $k_{\text{e; outer}} \times k_{\text{inner}}$ inner iterations.*

Proof. The proof follows the same outline of Theorem 4.3's proof. □

4.3 Proof of Corollary 4.1

We start this subsection by showing that Algorithm 4 is a special instance of Algorithm 2 for solving the strongly monotone inclusion (81).

Proposition 4.5. *Let* $\{z_j\}$, $\{z'_j\}$ *and* $\{\widetilde{z}_j\}$ *be generated by* Algorithm 4 *and let the operator* B *be as in* (79). *Define,*

$$v_j := \frac{z_{j-1} - z_j}{\gamma}, \qquad \varepsilon_j := \frac{\|z'_{j-1} - \widetilde{z}_j\|^2}{4\eta}, \quad \forall j \geq 1. \tag{103}$$

Then, for all $j \geq 1$,

$$v_j \in (1/\gamma)(\widetilde{z}_j - \mathring{z}) + C(\widetilde{z}_j) + F_1(\widetilde{z}_j) + F_2^{\varepsilon_j}(\widetilde{z}_j) \subset (1/\gamma)(\widetilde{z}_j - \mathring{z}) + B^{\varepsilon_j}(\widetilde{z}_j), \tag{104}$$

$$\|\gamma v_j + \widetilde{z}_j - z_{j-1}\|^2 + 2\gamma\varepsilon_j \leq \sigma^2\|\widetilde{z}_j - z_{j-1}\|^2, \tag{105}$$

$$z_j = z_{j-1} - \gamma v_j. \tag{106}$$

As a consequence, Algorithm 4 *is a special instance of* Algorithm 2 *with* $\lambda_j \equiv \gamma$ *for solving* (22) *with* $S(\cdot) := (1/\gamma)(\cdot - \mathring{z})$.

Proof. Note that the first identity in (83) gives

$$\frac{z_{j-1} - \widetilde{z}_j}{\gamma} - F_1(z'_{j-1}) \in (1/\gamma)(\widetilde{z}_j - \mathring{z}) + C(\widetilde{z}_j) + F_2(z'_{j-1}).$$

Adding $F_1(\widetilde{z}_j)$ in both sides of the above identity and using the second and first identities in (83) and (103), respectively, we find

$$v_j = \frac{z_{j-1} - z_j}{\gamma} \in (1/\gamma)(\widetilde{z}_j - \mathring{z}) + C(\widetilde{z}_j) + F_1(\widetilde{z}_j) + F_2(z'_{j-1}), \tag{107}$$

which, in turn, combined with Lemma A.2 and the definition of ε_j in (103) proves the first inclusion in (104). Note now that the second inclusion in (104) is a direct consequence of (79) and Proposition 2.1(b). Moreover, (106) is a direct consequence of the first identity in (103).

To prove (105), note that from (103), the second identity in (83), (82) and (77) we have

$$\begin{aligned}\|\gamma v_j + \widetilde{z}_j - z_{j-1}\|^2 + 2\gamma\varepsilon_j &= \gamma^2\|F_1(\widetilde{z}_j) - F_1(z'_{j-1})\|^2 + \frac{\gamma\|z'_{j-1} - \widetilde{z}_j\|^2}{2\eta} \\ &\leq \left(\gamma^2 L^2 + \frac{\gamma}{2\eta}\right)\|z'_{j-1} - \widetilde{z}_j\|^2 \\ &\leq \sigma^2\|z_{j-1} - \widetilde{z}_j\|^2,\end{aligned}$$

which is exactly the desired inequality, where we also used the facts that $z'_{j-1} = P_\Omega(z_{j-1})$, $\widetilde{z}_j \in D(C) \subset \Omega$ and that P_Ω is nonexpansive. The last statement of the proposition follows from (104)–(106), (81), (24) and (25). □

Proof of Corollary 4.1. Let, for all $j \geq 1$, $\{v_j\}$ and $\{\varepsilon_j\}$ be defined in (103). Using the last statement in Proposition 4.5 and Proposition 2.5 with $\mu := 1/\gamma$ and $\underline{\lambda} := \gamma$ we find

$$\|\gamma v_j\|^2 + 2\gamma\varepsilon_j \leq \frac{((1+\sigma)^2 + \sigma^2)(1-\alpha)^{j-1}\|\mathring{z} - z^*_\gamma\|^2}{1 - \sigma^2}, \tag{108}$$

where $z_\gamma^* := (S+B)^{-1}(0)$ with $S(\cdot) := (1/\gamma)(\cdot - \mathring{z})$, i.e., $z_\gamma^* = (\gamma B + I)^{-1}(\mathring{z})$. Now, using (108), (103) and Lemma A.1 we obtain

$$\|z_{j-1} - z_j\|^2 + \frac{\gamma\|z'_{j-1} - \widetilde{z}_j\|^2}{2\eta} \leq \frac{((1+\sigma)^2 + \sigma^2)(1-\alpha)^{j-1} d^2_{\mathring{z},b}}{1-\sigma^2}, \tag{109}$$

which in turn combined with (84), after some direct calculations, gives (85).

5 Numerical experiments

In this section, we perform simple numerical experiments on the family of (convex) constrained quadratic programming problems

$$\begin{aligned} &\text{minimize } \frac{1}{2}\langle Qz, z\rangle + \langle e, z\rangle \\ &\text{subject to } Kz = 0, z \in X, \end{aligned} \tag{110}$$

where $Q \in \mathbb{R}^{n\times n}$ is symmetric and either positive definite or positive semidefinite, $e = (1, \ldots, 1) \in \mathbb{R}^n$, $K = (k_j) \in \mathbb{R}^{1\times n}$, with $k_j \in \{-1, +1\}$ for all $j = 1, \ldots, n$, and $X = [0, 10]^n$ is a box in $\mathbb{R}^n$. Problem (110) appears, for instance, in support vector machine classifiers (see, e.g., [13, 16]). Here, $\langle\cdot,\cdot\rangle$ denotes the usual inner product in $\mathbb{R}^n$. A vector $z^* \in \mathbb{R}^n$ is a solution of (110), if and only if it solves the MIP

$$0 \in N_{\mathcal{M}}(z) + N_X(z) + Qz + e, \tag{111}$$

where $\mathcal{M} := \mathcal{N}(K) := \{z \in \mathbb{R}^n \,|\, Kz = 0\}$. Problem (111) is clearly a special instance of (76), in which

$$A(\cdot) := N_{\mathcal{M}}(\cdot), \quad C(\cdot) := N_X(\cdot), \quad F_1(\cdot) := 0 \text{ and } F_2(\cdot) := Q(\cdot) + e. \tag{112}$$

Moreover, in this case, $J_{\gamma A} = P_{\mathcal{M}}$ and $J_{\gamma C} = P_X$.

In what follows, we analyze the numerical performance of the following three algorithms for solving the MIP (111):

- The Douglas-Rachford-Tseng's F-B splitting method (Algorithm 5 (ALGO 5)) proposed in Section 4. We set $\sigma = 0.99$, $\theta = 0.01$, the operators A, C, F_1 and F_2 as in (112), and $\Omega = \mathbb{R}^n$, $L = 0$ and $\eta = 1/(\sup_{\|z\|\leq 1}\|Qz\|)$ (which clearly satisfy the conditions (E1)–(E5) of Section 4). We also have set $\gamma = 2\eta\sigma^2$ (see (82)) and $\tau_0 = \|z_0 - P_X(z_0) + Qz_0\|^3 + 1$.
- The relaxed forward-Douglas-Rachford splitting (rFDRS) from [16, Algorithm 1] (originally proposed in [6]). We set (in the notation of [16]) $\beta_V = 1/(\sup_{\|z\|\leq 1}\|(P_{\mathcal{M}} \circ Q \circ P_{\mathcal{M}})z\|)$, $\gamma = 1.99\beta_V$ and $\lambda_k \equiv 1$.
- The three-operator splitting scheme (TOS) from [17, Algorithm 1]. We set (in the notation of [17]) $\beta = 1/(\sup_{\|z\|\leq 1}\|Qz\|)$, $\gamma = 1.99\beta$ and $\lambda_k \equiv 1$.

For each dimension $n \in \{100, 500, 1000, 2000, 6000\}$, we analyzed the performance of each the above mentioned algorithms on a set of 100 randomly generated instances of (110). All the experiments were performed on a laptop equipped with an Intel i7 7500U CPU, 8 GB DDR4 RAM and a nVidia

GeForce 940MX. In order to allow performance comparison of ALGO 5, rFDRS and TOS, we adopted the stopping criterion

$$\|z_k - z_{k-1}\| \leq 10^{-6}, \tag{113}$$

for which we considered only extragradient steps when analyzing the performance of ALGO 5. The corresponding experiments are displayed in Tables 1 (Q positive definite), 2 (Table 1 continued), 3 (Q positive semidefinite) and 4 (Table 3 continued).

Now note that by using (87) and (89), we conclude that (113) is equivalent to

$$\gamma\|a_k + b_k\| = \|x_k - y_k\| \leq 10^{-6}. \tag{114}$$

Motivated by the above observation, we analyzed the performance of ALGO 5 on solving (110) while using the stopping criterion (114), for which both extragradient and null steps are considered. The corresponding results are displayed on Tables 5 and 6.

Finally, we mention that (111) consists of a *three-operator* MIP. For future research, we intend to study the numerical performance of Algorithm 5 in (true) *four-operator* MIPs. One possibility would be to consider structured minimization problems of the form

$$\min_{x\in\mathcal{H}} \left\{ f(x) + g(x) + \varphi(\widetilde{K}x) + h(x) \right\} \tag{115}$$

where $f, g, \varphi : \mathcal{H} \to (-\infty, +\infty]$ are proper closed convex functions, $h : \mathcal{H} \to \mathbb{R}$ is convex and differentiable and $\widetilde{K} : \mathcal{H} \to \mathcal{H}$ is a bounded linear operator. Under certain qualification conditions, (115) is equivalent to the MIP

$$0 \in \partial f(x) + \partial g(x) + \widetilde{K}^*\partial\varphi(\widetilde{K}x) + \nabla h(x) \tag{116}$$

which, in turn, is clearly equivalent to

$$\begin{aligned} &0 \in \partial f(x) + \partial g(x) + \widetilde{K}^* y + \nabla h(x) \\ &0 \in \partial\varphi^*(y) - \widetilde{K}x, \end{aligned} \tag{117}$$

where φ^* denotes the Fenchel-conjugate of φ. We now note that (117) is a special instance of (76) where, by letting $z = (x, y)$,

$$\begin{aligned} &A(z) := \partial f(x) \times \partial\varphi^*(y), \quad C(z) := \partial g(x) \times \{0\}, \\ &F_1(z) := (\widetilde{K}^* y, -\widetilde{K}x), \quad F_2(z) := (\nabla h(x), 0). \end{aligned} \tag{118}$$

Hence, under mild conditions on (115) (specially regarding conditions (E1)–(E5) on Section 4), Algorithm 5 is potentially applicable to solve (117) (i.e., (115)).

We also mention that while the variational problem (115) appears in different applications in Imaging and related fields, the primal-dual formulation (117) has been widely used in nowadays research in designing efficient primal-dual methods for, in particular, solving (115) (see, e.g., [3, 5, 12, 14, 15]).

Table 1: Running time (in seconds) and number of iterations performed by ALGO 5, rFDRS and TOS to reach the stopping criterion (113) on a set of 100 randomly generated instances of (110) with the matrix Q **positive definite**, with $n \in \{100, 500, 1000, 2000, 6000\}$. We can see that either ALGO 5 or TOS outperform the rFDRS in terms of (mean) running time, while ALGO 5 shows a slightly superior performance on large dimensions. Moreover, – see Table 2 – when compared to TOS, ALGO 5 provides a much more accurate approximate solution to the (unique) solution of (110).

		Time			Iterations		
n	Algorithm	Min	Max	Mean	Min	Max	Mean
100	ALGO 5	0.0014	0.0132	0.0017	12	21	15.21
	rFDRS	0.0014	0.0170	0.0018	6	15	**8.31**
	TOS	0.0010	0.0097	**0.0012**	7	16	9.37
500	ALGO 5	0.0302	0.0451	**0.0334**	15	22	17.24
	rFDRS	0.0517	0.0843	0.0626	8	15	**10.26**
	TOS	0.0288	0.0434	0.0340	9	16	11.28
1000	ALGO 5	0.3177	0.4190	**0.3540**	14	20	17.14
	rFDRS	0.4768	0.7304	0.5771	9	14	**10.84**
	TOS	0.3103	0.4504	0.3760	10	15	11.86
2000	ALGO 5	3.6411	3.9397	**3.7648**	19	21	19.80
	rFDRS	5.0062	5.5711	5.2795	11	12	**11.70**
	TOS	3.6632	4.0798	3.7703	12	13	12.70
6000	ALGO 5	94.7551	121.1123	**101.6311**	18	20	18.81
	rFDRS	107.3812	125.9018	115.0631	11	13	**12.20**
	TOS	94.7152	123.6527	104.0517	12	15	13.18

Table 2: Table 1 continued. Here (1) we show the number of extragradient and null steps performed by ALGO 5 while reaching the stopping criterion (113); (2) we evaluate the absolute error between the provided iterate z_k and the unique solution z^* of (110). We can see that, when compared to TOS, both ALGO 5 and rFDRS provide a much more accurate approximate solution.

		Extragradient steps			Null steps			$\|\|z_k - z_*\|\|$		
n	Algorithm	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
100	ALGO 5	8	13	10.24	3	4	3.58	0.0033	0.7829	**0.2714**
	rFDRS							0.0014	2.0768	0.3401
	TOS							2.8702	5.7959	4.2820
500	ALGO 5	9	14	11.64	3	5	4.30	0.0009	1.0619	**0.2004**
	rFDRS							0.0029	1.0275	0.3336
	TOS							5.7643	10.0704	7.8648
1000	ALGO 5	10	16	12.55	4	5	4.53	0.0036	0.5649	**0.1798**
	rFDRS							0.0008	0.9373	0.2763
	TOS							7.9120	13.3011	10.5356
2000	ALGO 5	11	19	13.30	4	6	4.70	0.0766	0.5499	0.2795
	rFDRS							0.1004	0.3433	**0.2278**
	TOS							13.3085	16.2610	14.4403
6000	ALGO 5	13	18	15.20	4	7	5.20	0.0437	0.6245	**0.2333**
	rFDRS							0.1626	1.0021	0.4523
	TOS							19.9698	24.4657	22.9383

Table 3: Running time (in seconds) and number of iterations performed by ALGO 5, rFDRS and TOS to reach the stopping criterion (113) on a set of 100 randomly generated instances of (110) with the matrix Q **positive semidefinite**, with $n \in \{100, 500, 1000, 2000, 6000\}$. Similarly to the case of Q positive semidefinite, we can see that either ALGO 5 or TOS outperform the rFDRS in terms of (mean) running time, while ALGO 5 shows a slightly superior performance on large dimensions.

		Time			Iterations		
n	Algorithm	Min	Max	Mean	Min	Max	Mean
	ALGO 5	0.0013	0.0130	0.0018	11	20	15.31
100	rFDRS	0.0013	0.0119	0.0017	6	18	**10.16**
	TOS	0.0008	0.0056	**0.0011**	7	19	11.19
	ALGO 5	0.0249	0.0409	**0.0283**	15	24	17.99
500	rFDRS	0.0466	0.0895	0.0563	9	18	**12.13**
	TOS	0.0236	0.0432	0.0291	10	19	13.13
	ALGO 5	0.3013	0.4061	**0.3229**	17	24	19.88
1000	rFDRS	0.4604	0.6780	0.5443	10	16	**13.18**
	TOS	0.2910	0.4354	0.3348	11	17	14.18
	ALGO 5	3.6649	4.0284	**3.7989**	18	25	21.75
2000	rFDRS	5.0060	5.3364	5.1248	13	15	**14.20**
	TOS	3.6933	4.0949	3.8607	14	16	15.27
	ALGO 5	101.0412	111.8105	**107.9423**	20	23	21.80
6000	rFDRS	115.9101	146.0409	130.2121	13	16	**15.01**
	TOS	105.1307	116.9615	110.3801	14	17	16.05

Table 4: Table 3 continued. Here we provide the number of extragradient and null steps performed by ALGO 5 while reaching the stopping criterion (113).

		Extragradient steps			Null steps		
n	Algorithm	Min	Max	Mean	Min	Max	Mean
100	ALGO 5	8	19	12.23	3	5	3.77
500	ALGO 5	11	21	14.14	4	5	4.31
1000	ALGO 5	12	20	15.42	4	5	4.54
2000	ALGO 5	14	20	15.95	4	6	4.70
6000	ALGO 5	14	21	16.55	4	7	5.35

Table 5: Running time (in seconds) and number of iterations performed by ALGO 5 to reach the stopping criterion (114) on a set of 100 randomly generated instances of (110) with the matrix Q **positive definite**, with $n \in \{100, 500, 1000, 2000, 6000\}$. We can see a slight improvement when compared to the results obtained via the stopping criterion (113) – cf. Table 1.

	Time			Iterations		
n	Min	Max	Mean	Min	Max	Mean
100	0.0012	0.0121	0.0016	11	17	13.66
500	0.0251	0.0492	0.0314	14	19	16.10
1000	0.3000	0.3539	0.3201	15	20	17.32
2000	3.5538	3.7583	3.5914	16	20	17.72
6000	98.8411	102.9118	99.7147	18	21	19.20

Table 6: Table 5 continued. Here, we show the number of extragradient and null steps performed by ALGO 5 while reaching the stopping criterion (114) and evaluate the absolute error between the provided iterate z_k and the unique solution z^* of (110) – cf. Table 2.

	Extragradient steps			Null steps			Absolute Error		
n	Min	Max	Mean	Min	Max	Mean	Min	Max	Mean
100	8	16	10.49	3	4	3.48	0.0007	0.6949	0.2354
500	9	15	11.82	3	5	4.22	0.0007	0.7109	0.2134
1000	10	16	12.61	4	5	4.42	0.0011	0.6628	0.1989
2000	12	16	13.46	4	6	4.64	0.0061	0.4537	0.1596
6000	13	17	14.50	4	6	4.80	0.0557	0.3666	0.2236

A Auxiliary results

Lemma A.1. ([1, Lemma 3.1]) *Let $z^*_\gamma := (\gamma B + I)^{-1}(\mathring{z})$ be the (unique) solution of* (81). *Then,*

$$\|\mathring{z} - z^*_\gamma\| \leq \|\mathring{z} - x^*\| \qquad \forall x^* \in B^{-1}(0). \tag{119}$$

Lemma A.2. ([41, Lemma 2.2]) *Let $F : \mathcal{H} \to \mathcal{H}$ be η–cocoercive, for some $\eta > 0$, and let $z', \widetilde{z} \in \mathcal{H}$. Then,*

$$F(z') \in F^{\varepsilon}(\widetilde{z}) \quad \textit{where} \quad \varepsilon := \frac{\|z' - \widetilde{z}\|^2}{4\eta}.$$

References

[1] M. Marques Alves, R. D. C. Monteiro, and B. F. Svaiter. Regularized HPE-type methods for solving monotone inclusions with improved pointwise iteration-complexity bounds. *SIAM J. Optim.*, 26(4):2730–2743, 2016.

[2] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011. With a foreword by Hédy Attouch.

[3] R. I. Bo¸t, E. R. Csetnek, A. Heinrich, and C. Hendrich. On the convergence rate improvement of a primal-dual splitting algorithm for solving monotone inclusion problems. *Math. Program.*, 150(2, Ser. A):251–279, 2015.

[4] R. I. Boţ and E. R. Csetnek. A hybrid proximal-extragradient algorithm with inertial effects. *Numer. Funct. Anal. Optim.*, 36(8):951–963, 2015.

[5] R. I. Boţ, E. R. Csetnek, and C. Hendrich. Recent developments on primal-dual splitting methods with applications to convex minimization. In *Mathematics without boundaries*, pages 57–99. Springer, New York, 2014.

[6] L. M. Briceño Arias. Forward-Douglas-Rachford splitting and forward-partial inverse method for solving monotone inclusions. *Optimization*, 64(5):1239–1261, 2015.

[7] L. M. Briceño Arias. Forward-partial inverse-forward splitting for solving monotone inclusions. *J. Optim. Theory Appl.*, 166(2):391–413, 2015.

[8] L. Briceno-Arias and D. Davis. Forward-backward-half forward algorithm with non self-adjoint linear operators for solving monotone inclusions. *preprint*, 2017.

[9] R. S. Burachik, A. N. Iusem, and B. F. Svaiter. Enlargement of monotone operators with applications to variational inequalities. *Set-Valued Anal.*, 5(2):159–180, 1997.

[10] R. S. Burachik, C. A. Sagastizábal, and B. F. Svaiter. ϵ-enlargements of maximal monotone operators: theory and applications. In *Reformulation: nonsmooth, piecewise smooth, semismooth and smoothing methods (Lausanne, 1997)*, volume 22 of *Appl. Optim.*, pages 25–43. Kluwer Acad. Publ., Dordrecht, 1999.

[11] L. C. Ceng, B. S. Mordukhovich, and J. C. Yao. Hybrid approximate proximal method with auxiliary variational inequality for vector optimization. *J. Optim. Theory Appl.*, 146(2):267–303, 2010.

[12] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vision*, 40(1):120–145, 2011.

[13] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, (2):27:1–27:27, 2011.

[14] P. L. Combettes. Systems of structured monotone inclusions: duality, algorithms, and applications. *SIAM J. Optim.*, 23(4):2420–2447, 2013.

[15] P. L. Combettes and J.-C. Pesquet. Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators. *Set-Valued Var. Anal.*, 20(2):307–330, 2012.

[16] D. Davis. Convergence rate analysis of the forward-Douglas-Rachford splitting scheme. *SIAM J. Optim.*, 25(3):1760–1786, 2015.

[17] D Davis and W. Yin. A three-operator splitting scheme and its optimization applications. *Set-valued and variational analysis*, 2017.

[18] J. Douglas, Jr. and H. H. Rachford, Jr. On the numerical solution of heat conduction problems in two and three space variables. *Trans. Amer. Math. Soc.*, 82:421–439, 1956.

[19] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programming*, 55(3, Ser. A):293–318, 1992.

[20] J. Eckstein and P. J. S. Silva. A practical relative error criterion for augmented Lagrangians. *Math. Program.*, 141(1-2, Ser. A):319–348, 2013.

[21] J. Eckstein and W. Yao. Relative-error approximate versions of Douglas-Rachford splitting and special cases of the ADMM. *Mathematical Programming*, 2017.

[22] R. Glowinski, S. J. Osher, and W. Yin, editors. *Splitting methods in communication, imaging, science, and engineering.* Scientific Computation. Springer, Cham, 2016.

[23] M. L. N. Gonçalves, J. G. Melo, and R. D. C. Monteiro. Improved pointwise iteration-complexity of a regularized ADMM and of a regularized non-Euclidean HPE framework. *SIAM J. Optim.*, 27(1):379–407, 2017.

[24] Y. He and R. D. C. Monteiro. An accelerated HPE-type algorithm for a class of composite convex-concave saddle-point problems. *SIAM J. Optim.*, 26(1):29–56, 2016.

[25] A. N. Iusem and W. Sosa. On the proximal point method for equilibrium problems in Hilbert spaces. *Optimization*, 59(8):1259–1274, 2010.

[26] P.-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM J. Numer. Anal.*, 16(6):964–979, 1979.

[27] P. A. Lotito, L. A. Parente, and M. V. Solodov. A class of variable metric decomposition methods for monotone variational inclusions. *J. Convex Anal.*, 16(3-4):857–880, 2009.

[28] J.-E. Martinez-Legaz and M. Théra. ϵ-subdifferentials in terms of subdifferentials. *Set-Valued Anal.*, 4(4):327–332, 1996.

[29] R. D. C. Monteiro, C. Ortiz, and B. F. F. Svaiter. Implementation of a block-decomposition algorithm for solving large-scale conic semidefinite programming problems. *Comput. Optim. Appl.*, 57(1):45–69, 2014.

[30] R. D. C. Monteiro, C. Ortiz, and B. F. F. Svaiter. An adaptive accelerated first-order method for convex optimization. *Comput. Optim. Appl.*, 64(1):31–73, 2016.

[31] R. D. C. Monteiro and B. F. Svaiter. Complexity of variants of Tseng's modified F-B splitting and Korpelevich's methods for hemivariational inequalities with applications to saddle point and convex optimization problems. *SIAM Journal on Optimization*, 21:1688–1720, 2010.

[32] R. D. C. Monteiro and B. F. Svaiter. On the complexity of the hybrid proximal extragradient method for the iterates and the ergodic mean. *SIAM Journal on Optimization*, 20:2755–2787, 2010.

[33] R. D. C. Monteiro and B. F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *SIAM J. Optim.*, 23(1):475–507, 2013.

[34] R.D.C Monteiro and S. Chee-Khian. Complexity of the relaxed Peaceman-Rachford splitting method for the sum of two maximal strongly monotone operators. *Submitted*, 2017.

[35] R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM J. Control Optimization*, 14(5):877–898, 1976.

[36] M. V. Solodov and B. F. Svaiter. A hybrid approximate extragradient-proximal point algorithm using the enlargement of a maximal monotone operator. *Set-Valued Anal.*, 7(4):323–345, 1999.

[37] M. V. Solodov and B. F. Svaiter. A hybrid projection-proximal point algorithm. *J. Convex Anal.*, 6(1):59–70, 1999.

[38] M. V. Solodov and B. F. Svaiter. An inexact hybrid generalized proximal point algorithm and some new results on the theory of Bregman functions. *Math. Oper. Res.*, 25(2):214–230, 2000.

[39] M. V. Solodov and B. F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numer. Funct. Anal. Optim.*, 22(7-8):1013–1035, 2001.

[40] B. F. Svaiter. On weak convergence of the Douglas-Rachford method. *SIAM J. Control Optim.*, 49(1):280–287, 2011.

[41] B. F. Svaiter. A class of Fejér convergent algorithms, approximate resolvents and the hybrid proximal-extragradient method. *J. Optim. Theory Appl.*, 162(1):133–153, 2014.

[42] P. Tseng. A modified forward-backward splitting method for maximal monotone mappings. *SIAM J. Control Optim.*, 38(2):431–446 (electronic), 2000.