Journal of Global Optimization (2021) 79:257–277
https://doi.org/10.1007/s10898-020-00940-w

Weak convergence of an extended splitting method for monotone inclusions

Yunda Dong[1]

Received: 12 June 2019 / Accepted: 2 August 2020 / Published online: 7 August 2020

Abstract
In this article, we consider the problem of finding zeros of monotone inclusions of three operators in real Hilbert spaces, where the first operator's inverse is strongly monotone and the third is linearly composed, and we suggest an extended splitting method. This method allows relative errors and is capable of decoupling the third operator from linear composition operator well. At each iteration, the first operator can be processed with just a single forward step, and the other two need individual computations of the resolvents. If the first operator vanishes and linear composition operator is the identity one, then it coincides with a known method. Under the weakest possible conditions, we prove its weak convergence of the generated primal sequence of the iterates by developing a more self-contained and less convoluted techniques. Our suggested method contains one parameter. When it is taken to be either zero or two, our suggested method has interesting relations to existing methods. Furthermore, we did numerical experiments to confirm its efficiency and robustness, compared with other state-of-the-art methods.

Keywords Monotone inclusions · Splitting method · Scaling factors · Weak convergence

Mathematics Subject Classification 58E35 · 65K15

1 Introduction

Let $\mathcal{H}$, $\mathcal{G}$ be infinite-dimensional Hilbert spaces. In this article, we are mainly concerned with the following problem of finding an x in $\mathcal{H}$ such that

$$0 \in C(x) + A(x) + Q^* B(Qx - q), \tag{1}$$

where the inverse C^{-1} of the operator $C : \mathcal{H} \to \mathcal{H}$ is strongly monotone, $A : \mathcal{H} \rightrightarrows \mathcal{H}$ and $B : \mathcal{G} \rightrightarrows \mathcal{G}$ are maximal monotone, and $Q : \mathcal{H} \to \mathcal{G}$ is nonzero bounded linear with its

✉ Yunda Dong
ydong@zzu.edu.cn

[1] School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, People's Republic of China

adjoint Q^*, and $q \in \mathcal{G}$. Throughout this article, the problem's solution set is always assumed to be nonempty.

As shown in [1,2], the monotone inclusion above provides a simple but powerful framework of reformulating practical problems. In particular, it includes convex minimization problems, complementarity problems, monotone variational inequality problems and so on.

A simplified form of the monotone inclusion above is the following problem of finding an $x \in \mathcal{H}$ such that

$$0 \in A(x) + B(x). \tag{2}$$

To date, there have been three main splitting methods for solving this monotone inclusion of two operators. The first one is the forward-backward splitting method [3,4], whose recursive formula (in the case of A being single-valued) reads

$$x^{k+1} = (I + \mu_k B)^{-1}(x^k - \mu_k A(x^k)),$$

where $\mu_k > 0$. Its weak convergence can be proved either A^{-1} is strongly monotone [5,6] or both A is Lipschitz continuous and monotone and $A + B$ is strongly monotone [7] provided that the steplength is limited to some conservative constant. Since these assumptions are rather restrictive, one may resort to the second splitting method [3] proposed in the year 1979

$$\begin{aligned} x^k &= (I + \mu A)^{-1}(z^k), \\ y^k &= (I + \mu B)^{-1}(2x^k - z^k), \\ z^{k+1} &= z^k - \gamma(x^k - y^k), \end{aligned}$$

where $\mu > 0$ is scaling factor and the parameter $\gamma \in (0, 2)$. Lions and Mercier [3] analyzed weak convergence of the auxiliary sequence $\{z^k\}$ and [8] proved weak convergence of the main sequence $\{x^k\}$ to the solution point of the problem (2) above. In the $\gamma = 1$ case, Lions and Mercier [3] called it the Douglas–Rachford algorithm. This is because it has its root in the alternating-direction implicit iterative method for solving special systems of linear equations, due to the work of Douglas and Rachford [9] in the year 1956. Sometimes it is called "the Douglas–Rachford splitting method" and this is even used for the case where $\gamma \in (0, 2)$. In the forbidden case of $\gamma = 2$, it corresponds to the Peaceman–Rachford algorithm [10]. The third one is Tseng's splitting method [1] proposed in the year 2000

$$\begin{aligned} y^k &:= (I + \mu_k B)^{-1}(x^k - \mu_k A(x^k)), \\ x^{k+1} &= y^k - \mu_k A(y^k) + \mu_k A(x^k), \end{aligned}$$

where $\mu_k > 0$. If the forward operator A is (locally) Lipschitz continuous and monotone and the backward operator B is maximal monotone, and if μ_k is chosen in some proper way, then its weak convergence can be guaranteed. For a practical relaxed version, we refer to [11] for detailed discussions. If A is further linear, then the interested reader may see [12] (i.e., [13, Algorithm 3.0]) for a first full splitting method.

Note that, in the context of the three dominating splitting methods just mentioned above, the scaling factors for A and B are identical. Desirably, the method of [14, Algorithm 2] overcomes this issue by turning to solve prima-dual system of the primal problem (2)

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} A & 0 \\ 0 & B^{-1} \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} + \begin{pmatrix} 0 & I^* \\ -I & 0 \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix},$$

where (x, u) is the primal-dual variable. Choose $t \in \mathcal{R}$, for the current primal-dual iterate (x^k, u^k), compute

$$\begin{aligned} y^k &= (\alpha I + A)^{-1}(\alpha x^k - u^k), \\ \hat{y}^k &= (1-t)x^k + t y^k, \\ v^k &= (\beta I + B)^{-1}(\beta \hat{y}^k + u^k), \end{aligned}$$

where $\alpha > 0$ and $\beta > 0$ are the scaling factors. Then, make use of all these information to update the current iterate (x^k, u^k) to get the new one, respectively. While the weak convergence of most splitting methods is obtained by either using a fixed-point approach (e.g. [3]) or introducing an appropriate descent direction (e.g. [8]), the method of [14, Algorithm 2] is designed and analyzed as a different way to create a sequence of separating hyperplanes between the current iterate and the primal-dual solution set. For other discussions of the iterative schemes of primal-dual systems, the reader may consult [1,2,12,13,15–20].

In this article, we aim at extending the method of [14, Algorithm 2] for solving monotone inclusions from two operators to three ones. For our extended splitting method, its main recursive formulae are as follows. Choose $t \in \mathcal{R}$, for the current primal-dual iterate (x^k, u^k), compute

$$\begin{aligned} y^k &= (\alpha I + A)^{-1}(\alpha x^k - C(x^k) - Q^* u^k), \\ \hat{y}^k &:= (1-t)x^k + t y^k, \\ v^k &= (\beta I + B)^{-1}(\beta(Q\hat{y}^k - q) + u^k), \end{aligned}$$

where $\alpha > 0$ and $\beta > 0$ are the scaling factors. Then, we construct

$$-\left(\alpha(x^k - y^k) + \beta Q^*(Q\hat{y}^k - q - v^k),\ v^k - Qy^k + q\right)$$

and confirm that it can serve as a descent direction of $\|(x, u) - (x^*, u^*)\|^2$ at the current iterate, if the primal-dual solution point is unknown. Finally, by developing a more self-contained and less convoluted techniques, we prove weak convergence of our suggested splitting method.

On the one hand, when C vanishes and $Qx - q$ corresponds to the identity map, the main iterative formulae of our suggested splitting method coincide with the ones in [14, Algorithm 2]. Thus it can be viewed as an extension of their method. Be aware that our extension is of linear composition and is certainly different from [16], where a general algorithmic framework for finding a zero of the sum of $n \geq 3$ maximal monotone operators over a real Hilbert space was proposed, without first reducing the problem to some appropriate monotone inclusions of two operators.

On the other hand, when $t = 2$, our suggested splitting method is much like the method of [21,22], whose main iterative formulae include:

$$\begin{aligned} y^k &= (\alpha I + A)^{-1}(\alpha x^k - C(x^k) - Q^* u^k), \\ \hat{y}^k &:= 2y^k - x^k, \\ v^k &= (I + \beta B^{-1})^{-1}(\beta(Q\hat{y}^k - q) + u^k). \end{aligned}$$

Yet, the descent direction there is $(x^k - y^k, u^k - v^k)$ and thus is widely different. Moreover, our assumptions on the scaling factors are by far weaker than their ones; see Sect. 5 for more details. For very recent discussions of splitting methods for the monotone inclusions of three operators when linear composition reduces to identity, we refer to [20,23,24], to cite a few.

Notice that, our suggested splitting method shares a nice property with the method of [21,22]: decouple the third operator B from linear composition operator Q well. Therefore, each can solve the problem such as (1) in a practically efficient way. Furthermore, it allows relative errors as done in [16]. In contrast. the method of [21,22] requires absolutely summable errors.

The rest of this article is organized as follows. In Sect. 2, we give some useful concepts and preliminary results. In Sect. 3, we fully state our suggested splitting method in Hilbert spaces. In Sect. 4, under the weakest possible conditions, we prove its weak convergence of the generated primal sequence of the iterates. In particular, our proof is different from the proceeding ones [14,21,25]. In Sect. 5, we discuss the relations of our suggested splitting method to some existing ones. In In Sect. 6, we did numerical experiments to confirm the method's efficiency and robustness for our test problems, compared with some state-of-the-art methods [21,22]. In Sect. 7, we close this article by some concluding remarks.

2 Preliminary results

In this section, we first give some basic definitions and then provide some auxiliary results for later use.

Let $\mathcal{H}$ be an infinite-dimensional Hilbert space, in which $\langle x, y\rangle$ stands for the usual inner product and $\|x\| := \sqrt{\langle x, x\rangle}$ for the induced norm for any $x, y \in \mathcal{H}$. I stands for the identity operator, i.e., $Ix = x$ for all $x \in \mathcal{H}$. Let $T : \mathcal{H} \rightrightarrows \mathcal{H}$ be a possibly multi-valued operator. dom$T$ stands for the effective domain of T, i.e., $\mathrm{dom}T := \{x \in \mathcal{H} : Tx \neq \emptyset\}$.

Definition 2.1 Let $T : \mathcal{H} \to \mathcal{H}$ be a single-valued operator. If there exists some constant number $\kappa > 0$ such that

$$\|T(x) - T(y)\| \leq \kappa\|x - y\|, \quad \forall x, y \in \mathcal{H},$$

then T is called κ-Lipschitz continuous.

To concisely give the following definition, we agree on that the notation $(x, w) \in T$ and $x \in \mathcal{H}$, $w \in T(x)$ have the same meaning. Moreover, $w \in Tx$ if and only if $x \in T^{-1}w$, where T^{-1} stands for the inverse of T.

Definition 2.2 Let $T : \mathcal{H} \rightrightarrows \mathcal{H}$ be an operator. It is called monotone if and only if

$$\langle x - x',\ w - w'\rangle \geq 0, \qquad \forall (x, w) \in T,\ \ \forall (x', w') \in T;$$

maximal monotone if and only if it is monotone and for given $\hat{x} \in \mathcal{H}$ and $\hat{w} \in \mathcal{H}$ the following implication relation holds

$$\langle x - \hat{x},\ w - \hat{w}\rangle \geq 0, \quad \forall (x, w) \in T \quad \Rightarrow \quad (\hat{x}, \hat{w}) \in T.$$

Definition 2.3 Let $T : \mathcal{H} \rightrightarrows \mathcal{H}$ be an operator. It is called μ-strongly monotone if and only if there exists $\mu > 0$ such that

$$\langle x - x',\ w - w'\rangle \geq \mu\|x - x'\|^2, \qquad \forall (x, w) \in T,\ \ \forall (x', w') \in T.$$

Definition 2.4 Let $C : \mathcal{H} \to \mathcal{H}$ be an operator. C^{-1} is called c-strongly monotone if there exists some $c > 0$ such that

$$\langle x - y, C(x) - C(y)\rangle \geq c\|C(x) - C(y)\|^2, \forall x, y \in \mathcal{H}.$$

In particular, if $C(x) = Mx + q$, where M is an $n \times n$ positive semi-definite matrix and q is an n-dimensional vector, then

$$\langle x, Mx\rangle \geq \lambda_{\max}^{-1} \|Mx\|^2, \quad \forall x \in \mathcal{R}^n,$$

where $\lambda_{\max}$ is the largest eigenvalue of M.

Notice that Definition 2.4 is an instance of the celebrated Baillon-Haddad theorem (cf. [26, Remark 3.5.2]). Sometimes, we also call that the operator C given in this definition is c-cocoercive or c-inverse strongly monotone. In this case, C must be Lipschitz continuous and monotone.

Definition 2.5 Let $f : \mathcal{H} \to (-\infty, +\infty]$ be a closed proper convex function. Then for any given $x \in \mathcal{H}$ the sub-differential of f at x is defined by

$$\partial f(x) := \{s \in \mathcal{H} : f(y) - f(x) \geq \langle s, y - x\rangle, \ \forall y \in \mathcal{H}\}.$$

Each s is called a sub-gradient of f at x. Moreover, if f is further continuously differentiable, then $\partial f(x) = \{\nabla f(x)\}$, where $\nabla f(x)$ is the gradient of f at x.

As is well known, the sub-differential of any closed proper convex function in an infinite-dimensional Hilbert space is maximal monotone as well. An example is the following indicator function

$$\delta_{\mathcal{C}}(x) = \begin{cases} 0, & \text{if } x \in \mathcal{C}, \\ +\infty, & \text{if } x \notin \mathcal{C}. \end{cases}$$

where $\mathcal{C}$ is some nonempty closed convex set in $\mathcal{H}$, and its sub-differential must be closed, proper convex. Furthermore, for any given positive number $\lambda > 0$, we have $P_{\mathcal{C}} = (I + \lambda\partial\delta_{\mathcal{C}})^{-1}$, where $P_{\mathcal{C}}$ is usual projection onto $\mathcal{C}$.

For any given maximal monotone operator $T : \mathcal{H} \rightrightarrows \mathcal{H}$, it is Minty [27] who proved that there must exist a unique $y \in \mathcal{H}$ such that $(I + \lambda T)(y) \ni x$ for all $x \in \mathcal{H}$ and $\lambda > 0$. This implies that the corresponding operator $(I + \lambda T)^{-1}$, known as the resolvent operator with respect to T for any given $\lambda > 0$, is single-valued.

For any given maximal monotone operator $T : \mathcal{H} \rightrightarrows \mathcal{H}$, there are other related properties. (i) For all $x \in \mathcal{H}$, the set $T(x)$ must be either empty or nonempty closed convex; see [28, Proposition 3, §6.7]. (ii) The solution set $\{x : 0 \in T(x)\}$ is either empty or nonempty closed convex.

An important instance of the problem (1) above is

$$C := \nabla h, \quad A = \partial f, \quad B = \partial g,$$

where $h, f : \mathcal{H} \to (-\infty, +\infty]$, $g : \mathcal{G} \to (-\infty, +\infty]$ are closed proper convex functions, and ∇h is assumed to be Lipschitz continuous. Under suitable conditions, (1) corresponds to optimality condition of the following convex minimization

$$\min_{x \in \mathcal{H}} \ h(x) + f(x) + g(Qx - q). \tag{3}$$

Note that the problem (1) results in the associated Kuhn-Tucker set

$$Z := \left\{(x, u) \in (\mathcal{H}, \mathcal{G}) : -Q^* u \in C(x) + A(x), \quad Qx - q \in B^{-1}(u) \quad \right\}, \tag{4}$$

where (x, u) is called primal-dual variable. Next, we will show that such a set must be closed convex according to the following Lemma 2.1.

Lemma 2.1 *Let A, B, L be operators defined in the problem (1). Then the resulting operator by Attouch-Thera duality principle*

$$T(x,u) = \begin{pmatrix} C+A & 0 \\ 0 & B^{-1} \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} + \begin{pmatrix} 0 & Q^* \\ -Q & 0 \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} + \begin{pmatrix} 0 \\ q \end{pmatrix} \tag{5}$$

must be maximal monotone.

Proof Note that both $C + A$ and B are maximal monotone. Thus, the first operator on the right-hand side must be maximal monotone. Meanwhile, the linearity of Q means that the second must be maximal monotone as well [27]. Maximality of T follows from [29, Theorem 1]. □

Although Lemma 2.1 is a known result, we would like to follow [30, Proposition 2] or [31, Corollary 4.2] to give such a short proof for completeness.

Remark 2.1 In his PhD dissertation, the author [12, Chapter 4] suggested using some splitting methods such as [12, Algorithm 4.2.1] (i.e., [13, Algorithm 3.0]) to instead solve primal-dual system (5) of the primal problem (1). Notice that this dissertation can be found in the author's Researchgate.

Next, we would like to mention the Douglas-Rachford splitting method once again. Its main recursive formulae may be:

$$\begin{aligned} (I + \mu A)(x^k) &\ni z^k, \\ (I + \mu B)(y^k) &\ni 2x^k - z^k, \\ z^{k+1} &= z^k - \gamma(x^k - y^k), \end{aligned}$$

where $\mu > 0$ and $\gamma \in (0, 2)$. On the other hand, [32] suggested an equivalent version: for known $x^k \in \mathcal{H}$, $a^k \in A(x^k)$, it has the following form

$$\begin{aligned} (I + \mu B)(y^k) &\ni x^k - \mu a^k, \\ (I + \mu A)(x^{k+1}) &\ni x^k + \mu a^k - \gamma(x^k - y^k). \end{aligned}$$

Recently, [8] proved weak convergence of the main sequence $\{x^k\}$ generated by the Douglas–Rachford splitting method. Impressively, the case of $\gamma \geq 2$ was discussed there for the first time. In the year 2012, the author also confirmed that, if A is further Lipschitz continuous, then the set sequence $\{(A + B)(y^k)\}$ asymptotically includes the origin and the speed of inclusion is at $o(1/k)$ in a sense. This result was first cited as [33, Theorem 2.2.4] in Zhou's master's thesis, completed in March of 2012. For an explanation, we refer to the manuscript entitled "An asymptotic inclusion speed for the Douglas–Rachford splitting method in Hilbert spaces", accepted by Optimization Online in December of 2014. For pertinent discussions, we refer to [34–39] for more details.

3 Splitting methods

In this section, we will give a detailed description of our extension of the method of [14, Algorithm 2] to solving the monotone inclusions of three operators (1). Moreover, we also make some remarks.

Algorithm 3.1

Step 0. Choose $x^0 \in \mathcal{H}$, $u^0 \in \mathcal{G}$, $t \in \mathcal{R}$ and $\theta \in (0, 2)$. Calculate c. Choose $\sigma_1, \sigma_2 \in [0, 1)$ and choose α and β such that

$$\alpha > \sigma_1 + 1/(4c), \quad \beta > \sigma_2$$
$$4\left(\alpha - \frac{1}{4c} - \sigma_1\right) > \frac{t^2\beta^2\|Q\|^2}{\beta - \sigma_2}. \tag{6}$$

Set $k := 0$.

Step 1. For known x^k, u^k, compute

$$(\alpha I + A)(y^k) \ni \alpha x^k - C(x^k) - Q^* u^k + \xi^k, \tag{7}$$
$$\hat{y}^k := (1-t)x^k + t y^k, \tag{8}$$
$$(\beta I + B)(v^k) \ni \beta(Q\hat{y}^k - q) + u^k + \eta^k, \quad v^k := Qz^k - q, \tag{9}$$

where $\xi^k, \eta^k \in \mathcal{H}$ are errors and they satisfy

$$\|\xi^k\| \le \sigma_1\|x^k - y^k\|, \quad \|\eta^k\| \le \sigma_2\|Q(x^k - z^k)\|. \tag{10}$$

Compute $\gamma_k = \theta t_{1,k}/t_{2,k}$, where $t_{1,k}$ $t_{2,k}$ are given by

$$t_{1,k} = \left(\alpha - \frac{1}{4c}\right)\|x^k - y^k\|^2 + \langle x^k - y^k, \xi^k\rangle + \beta\|Q(x^k - z^k)\|^2$$
$$+ \langle Q(x^k - z^k), \eta^k\rangle - t\beta\langle Q(x^k - y^k), Q(x^k - z^k)\rangle$$
$$t_{2,k} = \|d^k\|^2 + \|Q(z^k - y^k)\|^2,$$

where d^k is given by

$$d^k := \alpha(x^k - y^k) + \beta Q^* Q(\hat{y}^k - z^k) + \xi^k + Q^*\eta^k$$
$$= \alpha(x^k - y^k) + \beta Q^*(Q\hat{y}^k - q - v^k) + \xi^k + Q^*\eta^k.$$

Step 2. Compute

$$x^{k+1} = x^k - \gamma_k d^k \tag{11}$$
$$u^{k+1} = u^k - \gamma_k Q(z^k - y^k) = u^k - \gamma_k(v^k - Qy^k + q). \tag{12}$$

Set $k := k + 1$.

Notice that, for Algorithm 3.1, one may adopt $J_A \leftarrow \alpha I$, $J_B \leftarrow \beta I$ for the (7) and (9), where each J is bounded linear and strongly monotone, and the iterative formulae (11) and (12) are replaced by

$$\begin{pmatrix} x^{k+1} \\ u^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ u^k \end{pmatrix} - \gamma_k W \begin{pmatrix} d^k \\ Q(z^k - y^k) \end{pmatrix},$$

where W is bounded linear and strongly monotone. Such idea was given in [40, Algorithm 2.2].

Remark 3.1 Now let us make some remarks on the condition (6). In some practical applications, the setting becomes in finite-dimensional Euclidean spaces and Q corresponds to an

$m \times n$ matrix, say (q_{ij}). Since it follows from [41, Sect. 2.3] that $\|Q\| \le \vartheta$, where ϑ is the positive square root of

$$\max_{j=1,\ldots,n} \sum_{i=1}^{m} |q_{ij}| \cdot \max_{i=1,\ldots,m} \sum_{j=1}^{n} |q_{ij}|,$$

$\|Q\|^2$ in the condition (6) can be replaced by ϑ^2. In practice, if necessary, we may reformulate

$$Qx - q \quad \text{into} \quad (Q/\vartheta)x - q/\vartheta,$$

or something like this, in some cases.

Remark 3.2 Be care of that, in Algorithm 3.1, the scaling factors and the coefficients may vary from iteration to iteration. Yet, for notational simplicity, we fix them throughout this article.

4 Convergence

In this section, for the primal sequence generated by Algorithm 3.1, we analyze its weak convergence to an element of the associated solution set. Our proof techniques are different from existing ones [14,25].

The following lemma is a well-known result, we also refer to [44] for a discussion of a special case.

Lemma 4.1 *Consider any maximal monotone mapping $T : \mathcal{H} \rightrightarrows \mathcal{H}$. Assume that the sequence $\{w^k\}$ in $\mathcal{H}$ converges weakly to w, and the sequence $\{s^k\}$ on domT converges strongly to s. If $T(w^k) \ni s^k$ for all k, then the relation $T(w) \ni s$ must hold.*

Lemma 4.2 *Let $Q : \mathcal{H} \to \mathcal{G}$ be nonzero, bounded and linear operator, and let $\alpha > 0$, $t \in \mathcal{R}$. If $4\alpha > t^2\beta\|Q\|^2$, then the following*

$$\langle x, \alpha x\rangle + \langle u, \beta u\rangle - t\langle Qx, \beta u\rangle \ge \varphi(\alpha, \beta, tQ)\left(\|x\|^2 + \|u\|^2\right)$$

holds for all $x \in \mathcal{H}$ and all $v \in \mathcal{G}$, where

$$\varphi(\alpha, \beta, tQ) := \frac{1}{2}\left(\alpha + \beta - \sqrt{(\alpha-\beta)^2 + t^2\beta^2\|Q\|^2}\right).$$

Proof If $t = 0$, then the assertion holds. Below we assume $t \neq 0$. Since the following inequality

$$\langle x, \alpha x\rangle + \langle u, \beta u\rangle - t\langle Qx, \beta u\rangle \ge \left(\alpha - \frac{|t|\beta\|Q\|^2}{2\varepsilon}\right)\|x\|^2 + \left(\beta - \frac{|t|\beta\varepsilon}{2}\right)\|u\|^2$$

holds for any positive number ε, then taking

$$\varepsilon = \frac{\sqrt{(\alpha-\beta)^2 + t^2\beta^2\|Q\|^2} - (\alpha-\beta)}{|t|\beta}$$

to maximize

$$\min\left\{\alpha - \frac{|t|\beta\|Q\|^2}{2\varepsilon},\ \beta - \frac{|t|\beta\varepsilon}{2}\right\}$$

and thus to yield the desired relation (draw the graph of the min function!). □

To the author's best knowledge, Lemma 4.2 or its equivalent version was given in [14, Sect. 3] and [23, Lemma 5.1]. Very recently, such a nice result was used in [24] and generalized in the author's manuscript entitled "A new splitting method for systems of monotone inclusions in Hilbert spaces" in 2017.

Lemma 4.3 *For any given positive numbers $\alpha > 0$ and $\beta > 0$, if C^{-1} is c-strongly monotone and*

$$y = (\alpha I + A)^{-1}(\alpha x - C(x) - Q^* u + \xi), \tag{13}$$

$$\hat{y} = (1-t)x + ty, \tag{14}$$

$$v = (\beta I + B)^{-1}(\beta(Q\hat{y} - q) + u + \eta), \quad v := Qz - q, \tag{15}$$

then

$$\begin{aligned}&\langle x - x^*,\ \alpha(x-y) + \beta Q^* Q(\hat{y} - z) + \xi + Q^*\eta\rangle + \langle u - u^*,\ Q(z-y)\rangle \\ &\geq \left(\alpha - \frac{1}{4c}\right)\|x-y\|^2 + \langle x-y, \xi\rangle + \beta\|Q(x-z)\|^2 + \langle Q(x-z), \eta\rangle \\ &- t\beta\langle Q(x-y), Q(x-z)\rangle\end{aligned}$$

Proof Since there must be x^*, u^* such that

$$-Q^*u^* - C(x^*) \in A(x^*), \quad Qx^* - q \in B^{-1}(u^*), \tag{16}$$

it follows from (16) to (13) that

$$A(y) \ni \alpha(x-y) - C(x) - Q^*u + \xi, \quad A(x^*) \ni -C(x^*) - Q^*u^*,$$

which, together with monotonicity of A, imply

$$\langle y - x^*,\ \alpha(x-y) - (C(x) - C(x^*)) - Q^*(u-u^*) + \xi\rangle \geq 0. \tag{17}$$

Meanwhile, it follows from (16) to (15) that

$$B(Qz - q) \ni \beta Q(\hat{y} - z) + u + \eta, \quad B(Qx^* - q) \ni u^*,$$

which, together with monotonicity of B, imply

$$\langle Q(z - x^*),\ \beta Q(\hat{y} - z) + u - u^* + \eta\rangle \geq 0.$$

Summing up this inequality and (17) yields

$$\begin{aligned}&\langle x - x^*,\ \alpha(x-y) + \beta Q^* Q(\hat{y} - z) + \xi + Q^*\eta\rangle + \langle u - u^*,\ Q(z-y)\rangle \\ &\geq \alpha\|x-y\|^2 + \langle x-y, \xi\rangle + \beta\langle Q(x-z), Q(\hat{y}-z)\rangle \\ &+ \langle y - x^*, C(x) - C(x^*)\rangle + \langle Q(x-z), \eta\rangle.\end{aligned} \tag{18}$$

Next, we bound the inner product with respect to C. In fact, since C^{-1} is c-strongly monotone, we get

$$\begin{aligned}&\langle y - x^*, C(x) - C(x^*)\rangle \\ &= \langle x - x^*, C(x) - C(x^*)\rangle - \langle x - y, C(x) - C(x^*)\rangle \\ &\geq c\|C(x) - C(x^*)\|^2 - \frac{1}{2}\left(\frac{1}{2c}\|x-y\|^2 + 2c\|C(x) - C(x^*)\|^2\right) \\ &= -\frac{1}{4c}\|x-y\|^2.\end{aligned}$$

Combining this with (14) and (18) yields the desired result. □

Theorem 4.1 *Let $\{(x^k, u^k)\}$ be the sequence of the primal-dual iterates generated by Algorithm 3.1. Then $\{x^k\}$ weakly converges to an element of the solution to the problem (1) whenever the condition (6) is satisfied.*

Proof From Algorithm 3.1 and Lemmas 4.2 and 4.3, we have

$$\begin{aligned}
&\langle x^k - x^*, d^k\rangle + \langle u^k - u^*,\ Q(z^k - y^k)\rangle \\
&\quad= \langle x^k - x^*,\ \alpha(x^k - y^k) + \beta Q^*Q(\hat{y}^k - z^k) + \xi^k + Q^*\eta^k\rangle + \langle u^k - u^*,\ Q(z^k - y^k)\rangle \\
&\quad\ge \left(\alpha - \frac{1}{4c}\right)\|x^k - y^k\|^2 + \langle x^k - y^k, \xi^k\rangle + \beta\|Q(x^k - z^k)\|^2 + \langle Q(x^k - z^k), \eta^k\rangle \\
&\qquad - t\beta\langle Q(x^k - y^k), Q(x^k - z^k)\rangle\ (:= t_{1,k}) \qquad (19)\\
&\quad\ge \left(\alpha - \frac{1}{4c} - \sigma_1\right)\|x^k - y^k\|^2 + (\beta - \sigma_2)\|Q(x^k - z^k)\|^2 \\
&\qquad - t\beta\langle Q(x^k - y^k), Q(x^k - z^k)\rangle \\
&\quad\ge \varphi\left(\alpha - \frac{1}{4c} - \sigma_1, \beta - \sigma_2, \frac{t\beta}{\beta - \sigma_2}Q\right)\left(\|x^k - y^k\|^2 + \|Q(x^k - z^k)\|^2\right), \qquad (20)
\end{aligned}$$

where we have made use of the aforementioned relative error criteria (10).

By (11) and (12), we get

$$\begin{aligned}
\|x^{k+1} - x^*\|^2 &= \|x^k - x^* - \gamma_k d^k\|^2, \\
\|u^{k+1} - u^*\|^2 &= \|u^k - u^* - \gamma_k Q(z^k - y^k)\|^2.
\end{aligned}$$

From these two relations, we further have

$$\begin{aligned}
&\|x^{k+1} - x^*\|^2 + \|u^{k+1} - u^*\|^2 \\
&\quad= \|x^k - x^*\|^2 + \|u^k - u^*\|^2 - 2\gamma_k\left(\langle x^k - x^*, d^k\rangle + \langle u^k - u^*,\ Q(z^k - y^k)\rangle\right) \\
&\qquad + \gamma_k^2\left(\|d^k\|^2 + \|Q(z^k - y^k)\|^2\right) \\
&\quad\le \|x^k - x^*\|^2 + \|u^k - u^*\|^2 - 2\gamma_k t_{1,k} + \gamma_k^2\left(\|d^k\|^2 + \|Q(z^k - y^k)\|^2\right) \\
&\quad= \|x^k - x^*\|^2 + \|u^k - u^*\|^2 - (2 - \theta)\gamma_k t_{1,k},
\end{aligned}$$

where the inequality uses (19). So, it follows from (19) to (20) and the formula for γ_k in Algorithm 3.1 that

$$\begin{aligned}
&\|x^{k+1} - x^*\|^2 + \|u^{k+1} - u^*\|^2 \\
&\quad\le \|x^k - x^*\|^2 + \|u^k - u^*\|^2 - (2 - \theta)\gamma_k\varphi\left(\alpha - \frac{1}{4c} - \sigma_1, \beta - \sigma_2, \frac{t\beta}{\beta - \sigma_2}Q\right) \\
&\qquad \cdot\left(\|x^k - y^k\|^2 + \|Q(x^k - z^k)\|^2\right). \qquad (21)
\end{aligned}$$

Moreover, as long as (6) is satisfied, it is not difficult to confirm that γ_k is bounded below by some positive number. Consequently, it follows from (21) that

$$(i)\ \{(x^k, u^k) - (x^*, u^*)\} \text{ converges in norm, thus } \|(x^k, u^k)\| \text{ is bounded}; \qquad (22)$$

$$(ii)\ \|x^k - y^k\| \to 0,\quad \|Q(x^k - z^k)\| \to 0, \qquad (23)$$

which, together with (10) and strong monotonicity of C^{-1}, implies

$$\begin{aligned} &(iii)\ \ \|\xi^k\| \to 0, \quad \|\eta^k\| \to 0, \\ &(iv)\ \ \|C(x^k) - C(y^k)\| \le c^{-1}\|x^k - y^k\| \to 0, \end{aligned} \tag{24}$$

respectively.

On the other hand, from (5), the set $T(y^k,\ \beta Q(\hat{y}^k - z^k) + u^k + \eta^k)$ includes

$$\begin{pmatrix} C(y^k) + A(y^k) + \beta Q^* Q(\hat{y}^k - z^k) + Q^* u^k + Q^* \eta^k \\ B^{-1}(\beta Q(\hat{y}^k - z^k) + u^k) - Q y^k + q \end{pmatrix},$$

which, together with (7) and (9), i.e.,

$$\begin{aligned} &A(y^k) \ni \alpha(x^k - y^k) - C(x^k) - Q^* u^k + \xi^k, \\ &B^{-1}(\beta Q(\hat{y}^k - z^k) + u^k) \ni Q z^k - q, \end{aligned}$$

implies that the set $T(y^k,\ \beta Q(\hat{y}^k - z^k) + u^k)$ includes

$$\begin{pmatrix} \alpha(x^k - y^k) - C(x^k) + C(y^k) + \beta Q^* Q(\hat{y}^k - z^k) + \xi^k + Q^* \eta^k \\ Q(z^k - y^k) \end{pmatrix}.$$

This is equivalent to saying that it includes

$$\begin{pmatrix} \alpha(x^k - y^k) - C(x^k) + C(y^k) + \beta Q^*(Q(x^k - z^k) - t Q(x^k - y^k)) + \xi^k + Q^* \eta^k \\ Q(x^k - y^k) + Q(z^k - x^k) \end{pmatrix}. \tag{25}$$

The property above (22) tells us that $\{(x^k, u^k)\}$ has at least one weak cluster point, say $\{(x^\infty, u^\infty)\}$.

To invoke Lemma 4.1, let us to check $\{\beta Q(\hat{y}^k - z^k) + u^k + \eta^k\}$ and $\{y^k\}$ do converge to zero in norm. In fact, since we have known that $\{(x^k, u^k)\}$ has one weak cluster point $\{(x^\infty, u^\infty)\}$, there must exist some subsequence of $\{(x^k, u^k)\}$ such that

$$x^{k_j} \rightharpoonup x^\infty, \quad u^{k_j} \rightharpoonup u^\infty, \quad \text{as}\ \ k_j \to +\infty,$$

where the notation "$\rightharpoonup$" stands for weak convergence. It follows from this and (23) that

$$y^{k_j} \rightharpoonup x^\infty, \quad \text{as}\ \ k_j \to +\infty.$$

Since the operator Q is bounded and linear and (24) holds, a similar discussion yields

$$\beta Q(\hat{y}^{k_j} - z^{k_j}) + u^{k_j} + \eta^{k_j} \rightharpoonup u^\infty, \quad \text{as}\ \ k_j \to +\infty.$$

Meanwhile, by (23) and boundedness and linearity of Q, we can see that the upper part and the lower part of (25) converge in norm strongly to zero, respectively, as $k \to +\infty$. Combining these facts and Lemma 4.1 implies that, if $\{(x^\infty, u^\infty)\}$ is any weak cluster point of $\{(x^k, u^k)\}$, then it must be zero of T. Meanwhile, the proof of uniqueness of weak cluster point is standard [43], thus is omitted. So the whole sequence $\{x^k\}$ weakly converges to x^∞, which is the solution to the problem (1) above. □

Remark 4.1 For the primal sequence generated by Algorithm 3.1 addressed in real Hilbert spaces, here we have made use of Lemmas 2.1 and 4.1 to analyze weak convergence to an element of the primal solution set. Of course, we may invoke an alternative proof technique [25, Proposition 2.4] as well. Yet, we feel that our approach is not only more self-contained but is less convoluted than this other approach. Our proof techniques were originally developed for the earliest draft of this article at the beginning of the year 2017 and also used in the author's manuscript mentioned before Lemma 4.3.

5 Relations to other existing methods

In this section, we discuss relations of the above-mentioned Algorithm 3.1 to other existing ones, when it removes errors.

5.1 Case 1

In the case of $Q = I$ and $q = 0$ and $t = 1$, if C vanishes, then Algorithm 3.1 is somewhat related to the Spingarn splitting method [15, Sect. 5], which can written as

$$(\mu I + A)(x^k) \ni \mu z^k - v^k, \quad a^k \in A(x^k),$$
$$(\mu I + B)(y^k) \ni \mu z^k + v^k, \quad b^k \in B(y^k),$$
$$z^{k+1} = \frac{1}{2}(x^k + y^k), \quad v^{k+1} = \frac{1}{2}(b^k - a^k),$$

where $\mu > 0$ is the scaling factor. See [14, Sect. 4.2] for a detailed explanation of why it turns out to be an instance of the Douglas-Rachford splitting method.

5.2 Case 2

In the case of $Q = I$ and $q = 0$, if C vanishes, then Algorithm 3.1 reduces to the method of [14, Algorithm 2]. As far as the corresponding conditions on the scaling factors are concerned, (6) becomes $4\alpha > \beta t^2 \|Q\|^2$. See the full, compact set of recursions described at the end of [14, Sect. 3] there.

5.3 Case 3

In the case of $t = 0$, if C vanishes, then elementary calculations can indicate that Algorithm 3.1 is equivalent to the one in [25, Proposition 3.5] in theory, and the latter further subsumes [13, Algorithm 3.1].

5.4 Case 4

In the case of $t = 2$, Algorithm 3.1 is reminiscent of the method proposed by Vu [21] and Condat [22]. In our notation here, the main iterative formulae are as follows. Choose $\alpha > 0$ and $\beta > 0$. For known $x^k, \ u^k$, first compute

$$(\alpha I + A)(y^k) \ni \alpha x^k - C(x^k) - Q^* u^k, \tag{26}$$
$$\hat{y}^k := 2y^k - x^k, \tag{27}$$
$$(I + \beta B^{-1})(v^k) \ni \beta(Q\hat{y}^k - q) + u^k. \tag{28}$$

Then, choose $\gamma \in (0, 2)$ and compute

$$(x^{k+1}, u^{k+1}) = (x^k, u^k) - \gamma(x^k - y^k, u^k - v^k).$$

Obviously, (26) and (28) is the same as the counterparts of Algorithm 3.1 in the case of $t = 2$. Yet, the way of updating x^k to get the new iterate x^{k+1} is widely different. In addition, the reader may resort to the following Moreau identity

$$(I + \beta B^{-1})^{-1}(w) \equiv w - \beta(I + \frac{1}{\beta}B)^{-1}(\frac{w}{\beta}), \quad \forall \beta > 0, \ \forall w \in \mathcal{H}$$

to check some other relations of (28) to the counterpart of Algorithm 3.1, and we will not detail them here.

As to the convergence conditions on α, β and c, in the case of $t = 2$, we require

$$4\left(\alpha - \frac{1}{4c}\right) > \beta t^2 \|Q\|^2 \quad \Rightarrow \quad 4(\alpha - \beta\|Q\|^2) > 1/c$$

whereas their ones in [21, Theorem 3.1] and [22, Theorem 3.1] are

$$2\min\{\alpha, \beta^{-1}\}\left(1 - \sqrt{\alpha^{-1}\beta\|Q\|^2}\right) > 1/c$$

and

$$2(\alpha - \beta\|Q\|^2) > 1/c, \tag{29}$$

(in the case of $C := \nabla h$, where h is continuously differentiable, Lipschitz constant of ∇h can take $1/c$ by the Baillon-Haddad theorem) respectively. Obviously, in the common case of $\|Q\| = 1$, we have

$$4(\alpha - \beta) > 2(\alpha - \beta) > 2\min\{\alpha, \beta^{-1}\}\left(1 - \sqrt{\alpha^{-1}\beta}\right)$$

because it is easy to check

$$\min\{\alpha, \beta^{-1}\}\left(1 - \sqrt{\alpha^{-1}\beta}\right) = \frac{\min\{\alpha, \beta^{-1}\}}{\sqrt{\alpha}(\sqrt{\alpha} + \sqrt{\beta})}(\alpha - \beta), \quad \frac{\min\{\alpha, \beta^{-1}\}}{\sqrt{\alpha}(\sqrt{\alpha} + \sqrt{\beta})} < 1.$$

In this sense, the first is much weaker than the other two. Notice that, for the choice of γ, it is required in [21, Theorem 3.1] that $\gamma \in (0, 2)$ whereas in [22, Theorem 3.1]

$$0 < \gamma < 2 - \frac{1}{2c}(\alpha - \beta\|Q\|^2)^{-1} \in [1, 2).$$

In contrast, our choice of γ_k is flexible and self-adaptive, and it can be larger than 2; see Table 1 below.

If C further vanishes, the main iterative formulae of their method [21,22] mentioned above becomes

$$\begin{aligned}
&(\alpha I + A)(y^k) \ni \alpha x^k - Q^* u^k, \\
&\hat{y}^k := 2y^k - x^k, \\
&(I + \beta B^{-1})(v^k) \ni u^k + \beta(Q\hat{y}^k - q), \\
&x^{k+1} = x^k - \gamma(x^k - y^k), \\
&u^{k+1} = u^k - \gamma(u^k - v^k),
\end{aligned}$$

where $\gamma \in (0, 2)$. If $\gamma = 1$ and $\alpha > \beta\|Q\|^2$, then it can be viewed as a special case of the proximal point algorithm [42,43] (also see [35,36,45–49] for further discussions) in a sense. In contrast, Algorithm 3.1 seems beyond such framework.

5.5 Case 5

When the author was finalizing this revised version, the manuscript [20] appeared. A special case of the problem there is optimality condition the aforementioned convex minimization (3). The main body of the method of [20, Algorithm 1] can be stated as follows. Choose $\alpha_i \in (0, 1]$, $\rho_i > 0$, $i = 1, 2, 3$. Choose starting points $x_1^0, x_2^0, x_3^0, w_1^1, w_2^1, w_3^1 \in \mathcal{H}$, $z^1 \in \mathcal{G}$. At k-th iteration, compute in order

$$
\begin{aligned}
& t_1^k := (1-\alpha_1)x_1^{k-1} + \alpha_1 z^k - \rho_1(Cx_1^{k-1} - w_1^k),\ x_1^k = t_1^k,\ y_1^k = Cx_1^k,\\
& t_2^k := (1-\alpha_2)x_2^{k-1} + \alpha_2 z^k + \rho_2 w_2^k,\ (I+\rho_2 A)x_2^k \ni t_2^k,\ y_2^k = (t_2^k - x_2^k)/\rho_2,\\
& t_3^k := (1-\alpha_3)x_3^{k-1} + \alpha_3(Qz^k - q) + \rho_3 w_3^k,\ (I+\rho_3 B)x_3^k \ni t_3^k,\\
& y_3^k = (t_3^k - x_3^k)/\rho_3.
\end{aligned}
$$

Then, compute via an appropriate way to get the new iterates. For additional conditions on these parameters, we refer to [20, Sect. 3] for more details. Clearly, Algorithm 3.1 is different from this method. For example, our primal starting point is x^0 whereas their method has six primal starting points.

6 Rudimentary numerical experiments

In this section, we implemented Algorithm 3.1 (NEW for short) to solve three test examples to confirm its efficiency and robustness, compared with other state-of-the-art methods [21,22] (VC for short). Since computations of the resulting resolvents are not costly due to practically useful reformulation of these examples, we did not use error criteria for the methods. In our writing style, rather than striving for maximal test problems, we tried to make the basic ideas and techniques as clear as possible.

All numerical experiments were run in MATLAB R2014a (8.3.0.532) with 32-bit (win32) on a desktop computer with an Intel(R) Core(TM) i3-2120 CPU 3.30 GHz and 2 GB of RAM. The operating system is Windows XP Professional.

Our first test problem is to find an $x \in \mathcal{R}^m$ such that

$$0 \in Dx - d + Q^* \partial \delta_{\mathcal{C}}(Qx - q),$$

where $q = (0, \ldots, 0, 1/m)^T \in \mathcal{R}^{m+1}$ and

$$
D = \begin{pmatrix}
b & -1 & & & & \\
a & b & -1 & & & \\
 & \cdot & \cdot & \cdot & & \\
 & & \cdot & \cdot & \cdot & \\
 & & & a & b & -1 \\
 & & & & a & b
\end{pmatrix},\quad
Q = \begin{pmatrix}
1 & & & & & \\
 & 1 & & & & \\
 & \cdot & \cdot & \cdot & & \\
 & & \cdot & \cdot & \cdot & \\
 & & & & 1 & \\
-\frac{1}{m} & \cdot & \cdot & \cdot & \cdot & \cdot & -\frac{1}{m}
\end{pmatrix},
$$

where $a := -1 - h$, $b := 4 + 2h$, $h := 1/(m+1)$, and $\mathcal{C} \subseteq \mathcal{R}^{m+1}$ is the first orthant. To ensure that $e_1 = (1, 0, \ldots, 0)^T$ solves it, we set $d = -De_1$ in our practical implements. In addition, we chose

$$Cx = 0.5(D + D^T)x - d, \quad Ax = 0.5(D - D^T)x$$

to match the problem (1).

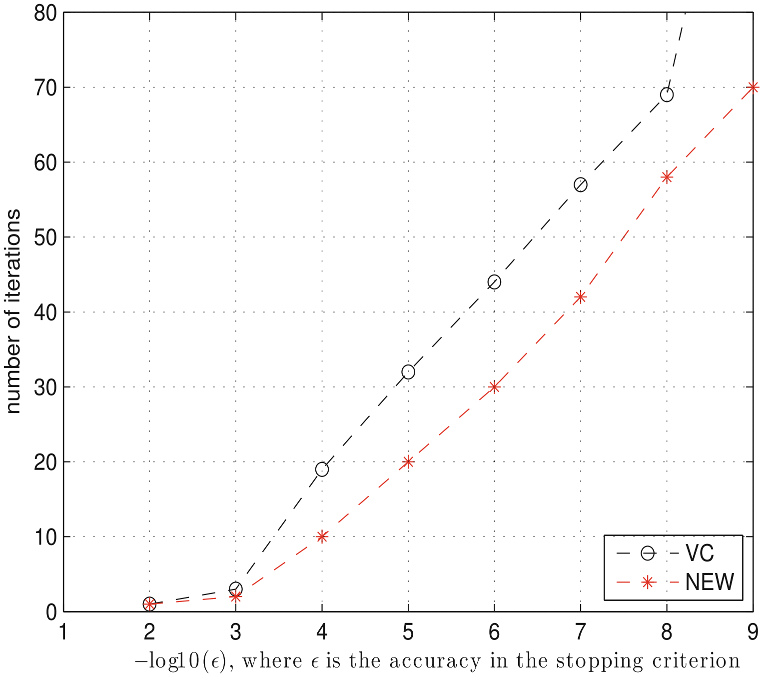

Fig. 1 Numerical results on the first test problem

Since $c = 1/6$, we suggested choosing α to be around $1/c = 6$. In practical implementations, for NEW and VC, we chose $\alpha \in \{4, 6, 8, 10, 12\}$ and

$$\beta = \rho(\alpha - 1/(4c)), \quad t = 2, \quad \theta = 1.8,$$
$$\beta = \rho/3, \quad \gamma = 1.8,$$

respectively. The reason why we adopted $\beta = \rho/3$ is: when $m = 1000$, $\|Q\|^2 \approx 1$, the condition in [21]

$$2\min\{\alpha, \beta^{-1}\}\left(1 - \sqrt{\alpha^{-1}\beta\|Q\|^2}\right) > 1/c \quad \Rightarrow \quad \min\{\alpha, \beta^{-1}\} > 3$$

implies $\alpha > 3$ and $\beta < 1/3$.

We adopted the stopping criterion $\|x_k - x^*\| \leq \epsilon\|x_0 - x^*\|$. After trials, we further chose $\alpha = 8, \rho = 0.9$ and $\alpha = 6, \rho = 0.5$ for VC and NEW, respectively. The corresponding numerical results were reported in Figs. 1 and 2.

From Figs. 1 and 2, we can see that NEW stably tends to achieve higher accuracy by using less number of iterations (and more elapsed time in most cases). In particular, it can achieve the accuracy of 10^{-9} order that VC fails to achieve.

Below, we further studied other two test problems, where C vanishes. To make NEW more practical, we chose $\alpha_0 > 0$ and, at k-th iteration, updated the involved scaling factor in the following way. First calculate

$$\phi_k := \frac{\alpha_k\|x_k - x_{k-1}\|}{\|A(x_k) - A(x_{k-1})\|},$$

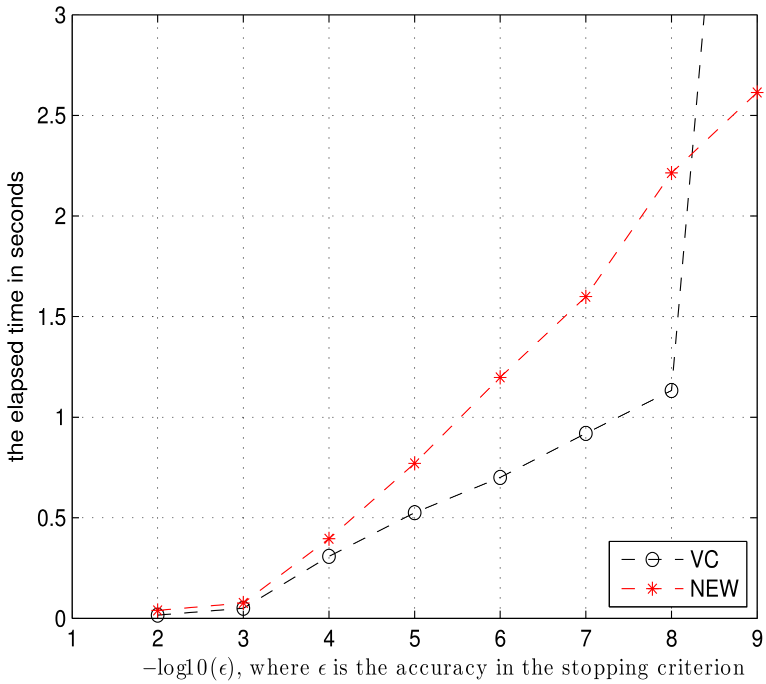

Fig. 2 Numerical results on the first test problem

where we have assumed that A is continuous. Then, update α_k via

$$\alpha_{k+1} = \begin{cases} 0.9\alpha_k & \text{if } \phi_k \geq 2, \\ 1.1\alpha_k & \text{if } \phi_k \leq 0.5, \\ \alpha_k & \text{otherwise.} \end{cases} \tag{30}$$

Here there is no worry about loss of convergence since it can be merely done for the first N (say, $N = 500$) iterations. In addition, it remains an open problem whether or not one can self-adaptively update the scaling factor for A when C exists as in the first test problem.

Our second test problem is monotone variational inequality problem, which is from [50] and is a user-optimized traffic pattern for the simple network with only two nodes x, y and five links a_1, a_2, a_3, b_1, b_2, where a_1, a_2, a_3 are directed from x to y and b_1, b_2 are the return of a_1, a_2, respectively.

The travel cost function and the constraint set are given by

$$F(x) = \begin{pmatrix} 10 & 0 & 0 & 5 & 0 \\ 0 & 15 & 0 & 0 & 5 \\ 0 & 0 & 20 & 0 & 0 \\ 2 & 0 & 0 & 20 & 0 \\ 0 & 1 & 0 & 0 & 25 \end{pmatrix} \begin{pmatrix} x_{a_1} \\ x_{a_2} \\ x_{a_3} \\ x_{b_1} \\ x_{b_2} \end{pmatrix} + \begin{pmatrix} 1000 \\ 950 \\ 3000 \\ 1000 \\ 1300 \end{pmatrix} \tag{31}$$

and

$$\mathcal{C} = \{x \in \mathcal{R}^5 : x \geq 0, x_{a_1} + x_{a_2} + x_{a_3} = 210, x_{b_1} + x_{b_2} = 120\},$$

respectively.

Table 1 Numerical results on γ_k in Algorithm 3.1

k	53	54	55	56	57	58	59	60
γ_k	1.2852	1.4687	1.6262	1.0989	1.9726	2.0878	1.4284	2.2592

This variational inequality problem corresponds to the monotone inclusion (1) via $F := A$, B is taken to be the sub-differential of the indicator function of $\{w \in \mathcal{R}^5 : w \geq 0\} \times \{(0, 0)^T\}$, and

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad q = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 210 \\ 120 \end{pmatrix}.$$

By the way, it seems a new idea, which occurs to the author in January of 2019, of reformulating the original problem into the monotone inclusion (1) under consideration. Such doing avoids the computation of usually expensive projection on the involved constraint set.

In practical implementations, we updated $F \leftarrow F/25$ and chose the starting points as

$$x^0 = (210, 0, 0, 120, 0)^T, \ u^0 = \text{zeros}(7, 1), \ \theta = 1.8$$

and adopted the ways of (30) to update α_k. As we know, $x^* = (120, 90, 0, 70, 50)^T$ is the unique solution to the variational inequality problem above. Furthermore, by Remark 3.1, we adopted the following preconditioning technique

$$\vartheta := sqrt(norm(Q, 1) * norm(Q^T, 1)), \quad (Q, q) \leftarrow (Q, q)/\vartheta.$$

After trials, we chose $\beta_k = \alpha_k$ for both VC and NEW. Then, we continued to try

$$\alpha_0 \in \{0.1, 1, 10\}, \quad \gamma \in \{0.9, 1.1, 1.3, 1.5, 1.7, 1.9\}$$

for VC and

$$\alpha_0 \in \{0.1, 1, 10\}, \quad t_k \in \{0, 0.5, 1, 1.5, 2, 2.1\}$$

for NEW. Finally, we further chose $\alpha_0 = 10$, $\gamma = 1.7$ and $\alpha_0 = 10$, $t_k = 2$ for VC and NEW, respectively.

Next, we numerically checked the computed values of γ_k in NEW. In the case of $\alpha_0 = 10$ and $t = 2$, we ran NEW and the returned last values of γ_k were listed in Table 1.

Our third second test problem is to solve the following complementarity problem of finding an $x \in \mathcal{R}^n$ such that

$$0 \in Mx + \partial\delta_{\mathcal{C}}(x),$$

where M is an $n \times n$ Hilbert matrix whose entries are

$$m_{i,j} = \frac{1}{i + j + 1}, \quad i, j = 0, \ldots, n - 1,$$

and $\mathcal{C}$ is the first orthant. Obviously, $x^* = (0, \ldots, 0)^T$ is the unique solution. The starting point and the stopping criterion are

$$(n = 10) \ \ x^0 = ones(n, 1), \quad u^0 = zeros(n, 1),$$

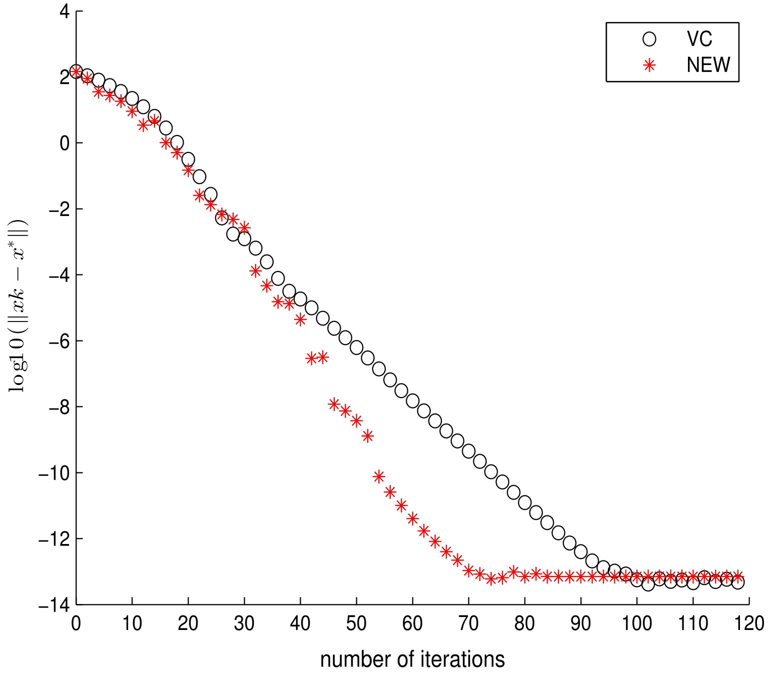

Fig. 3 Numerical results on the second test problem

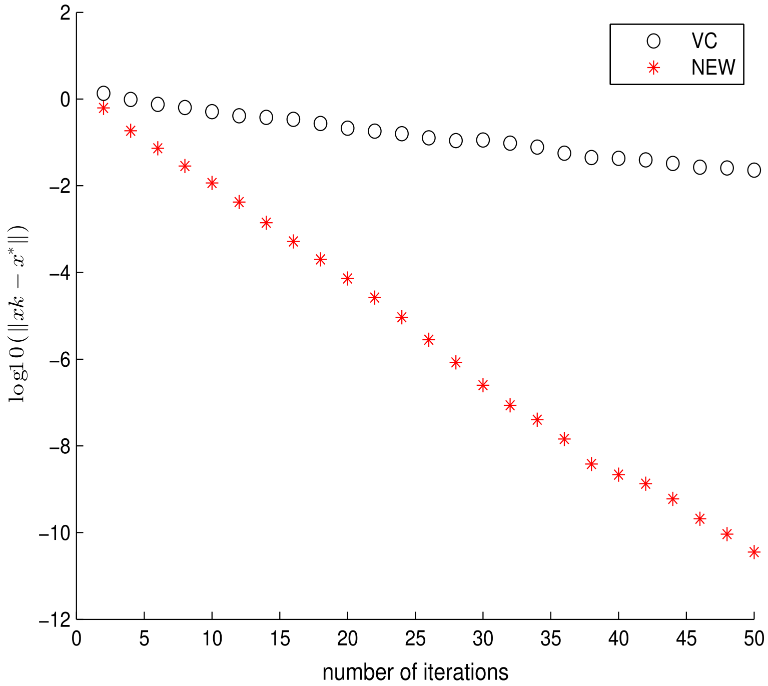

Fig. 4 Numerical results on the third test problem

respectively. For VC, in views of $c = 1$ and (29), we adopted

$$\begin{aligned} &\beta = \rho(\alpha - 0.5), \quad (\Rightarrow \ \alpha > 0.5), \quad \alpha \in \{0.8, 1, 5, 10\}, \\ &\rho \in \{0.01, 0.05, 0.1, 0.5, 0.9\}, \\ &\gamma = 2 - \frac{1}{2c}(\alpha - \beta)^{-1} - 0.0001. \end{aligned} \tag{32}$$

For NEW, we chose $\theta = 1$ and adopted the way of updating α_k as in (30) and calculated

$$\beta_k = \rho(\alpha_k - 1/(4c)).$$

After trials, we chose $\alpha = 5$, $\rho = 0.05$ and $\alpha_0 = 1$, $\rho = 0.5$ for VC and NEW, respectively.

The corresponding numerical results on the second and third test problems were reported in Figs. 3 and 4, respectively, where the elapsed time using tic and toc was not listed because it is always negligible in each case.

From Figs. 3 and 4, we can see that NEW outperformed VC clearly. Furthermore, NEW appears more robust. In addition, for the choice of γ in (32), we also tried to decrease it by a multiple of 0.9, 0.5, 0.1 and the results were still similar.

7 Conclusions

In this article, we have considered the problem of finding zeros of monotone inclusions of three operators in real Hilbert spaces, where the first operator's inverse is strongly monotone and the third is linearly composed, and we have suggested an extended splitting method, which allows relative errors, for its primal-dual system. In theory, under the weakest possible conditions, we have proved its weak convergence of the generated primal sequence of the iterates by developing a more self-contained and less convoluted techniques. In practice, we have done numerical experiments to confirm its efficiency and robustness, compared with other state-of-the-art methods.

There are three interesting topics arising in this article. The first would be to check whether or not it is possible to devise a practically useful way to calculate local approximation of strongly monotone constant of C^{-1}. The second would be to further update the scaling factor for B in a desirable way. The third would be to follow the scheme [51] to add inertial term to our extended splitting method. Resolving these issues are outside the scope of this article, but we hope to address them in future research.

Acknowledgements The author is greatly indebted to two referees for their careful reading and helpful suggestions. Special thanks go to Xiao Zhu for her help in doing numerical experiments.

References

1. Tseng, P.: A modified forward-backward splitting method for maximal monotone mapping. SIAM J. Control Optim. **38**, 431–446 (2000)
2. Combettes, P.L., Eckstein, J.: Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions. Math. Program. **168**(1–2), 645–672 (2018)
3. Lions, P.L., Mercier, B.: Splitting algorithms for the sum of two nonlinear operators. SIAM J. Numer. Anal. **16**, 964–979 (1979)
4. Passty, G.B.: Ergodic convergence to a zero of the sum of monotone oprators in Hilbert space. J. Math. Anal. Appl. **72**, 383–390 (1979)
5. Mercier, B.: Inéquations Variationnelles de la Mécanique, Publ. Math. Orsay, 80.01, Université de Paris-Sud, Orsay (1980)

6. Gabay, D.: Applications of the method of multipliers to variational inequalities. In: Fortin, M., Glowinski, R. (eds.) Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems, pp. 299–331. North-Holland, Amsterdam (1983)
7. Chen, H., Rockafellar, R.T.: Convergence rates in forward-backward splitting. SIAM J. Optim. **7**, 421–444 (1997)
8. Dong, Y.D., Fischer, A.: A family of operator splitting methods revisited. Nonlinear Anal. **72**, 4307–4315 (2010)
9. Douglas, J., Rachford, H.H.: On the numerical solution of the heat conduction problems in two and three space variables. Trans. Am. Math. Soc. **82**, 421–439 (1956)
10. Peaceman, D., Rachford, H.H.: The numerical solution of parabolic and elliptic differential equations. J. Soc. Ind. Appl. Math. **3**, 28–41 (1955)
11. Huang, Y.Y., Dong, Y.D.: New properties of forward-backward splitting and a practical proximal-descent algorithm. Appl. Math. Comput. **237**, 60–68 (2014)
12. Dong, Y.D.: Splitting Methods for Monotone Inclusions. PhD Dissertation, Nanjing University (2003)
13. Dong, Y.D.: An LS-free splitting method for composite mappings. Appl. Math. Lett. **18**(8), 843–848 (2005)
14. Eckstein, J., Svaiter, B.F.: A family of projective splitting methods for the sum of two maximal monotone operators. Math. Program. **111**, 173–199 (2008)
15. Spingarn, J.E.: Partial inverse of a monotone operator. Appl. Math. Optim. **10**(1), 247–265 (1983)
16. Eckstein, J., Svaiter, B.F.: General projective splitting methods for sums of maximal monotone operators. SIAM J. Control Optim. **48**(2), 787–811 (2009)
17. Eckstein, J.: A simplified form of block-iterative operator splitting and an asynchronous algorithm resembling the multi-block alternating direction method of multipliers. J. Optim. Theory Appl. **173**(1), 155–182 (2017)
18. Johnstone, P.R., Eckstein, J.: Convergence rates for projective splitting (2018). arXiv:1806.03920
19. Johnstone, P.R., Eckstein, J.: Projective splitting with forward steps: asynchronous and blockiterative operator splitting (2018). arXiv:1803.07043
20. Johnstone, P.R., Eckstein, J.: Single-forward-step projective splitting: exploiting cocoercivity (2019). arXiv:1902.09025
21. Vu, B.C.: A splitting algorithm for dual monotone inclusions involving cocoercive operators. Adv. Comput. Math. **38**(3), 667–681 (2013)
22. Condat, L.: A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. J. Optim. Theory Appl. **158**(2), 460–479 (2013)
23. Latafat, P., Patrinos, P.: Asymmetric forward-backward-adjoint splitting for solving monotone inclusions involving three operators. Comput. Optim. Appl. **68**(1), 57–93 (2017)
24. Dong, Y.D., Yu, X.H.: A new splitting method for monotone inclusions of three operators. Calcolo **56**(1), 25 (2019)
25. Alotaibi, A., Combettes, P.L., Shahzad, N.: Solving coupled composite monotone inclusions by successive Fejér approximations of their Kuhn-Tucker set. SIAM J. Optim. **24**(4), 2076–2095 (2014)
26. Zalinescu, C.: Convex Analysis in General Vector Spaces. World Scientific Publishing, River Edge (2002)
27. Minty, G.J.: Monotone (nonlinear) operators in Hilbert space. Duke Math. J. **29**, 341–346 (1962)
28. Aubin, J.P., Ekeland, I.: Applied Nonlinear Analysis. Interscience Publications. Wiley, New York (1984)
29. Rockafellar, R.T.: On the maximality of sums of nonlinear monotone operators. Trans. Am. Math. Soc. **149**(1), 75–88 (1970)
30. Eckstein, J., Ferris, M.C.: Smooth methods of multipliers for complementarity problems. Math. Program. **86**, 65–90 (1999)
31. Pennanen, T.: Dualization of generalized equations of maximal monotone type. SIAM J. Optim. **10**(3), 809–835 (2000)
32. Eckstein, J., Bertsekas, D.P.: On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. Math. Program. **55**(3), 293–318 (1992)
33. Zhou, S.F.: Operator Splitting Algorithms for ℓ_1-ℓ_2 Problem and Interior Point Method for Its Duality Problem. Thesis Dissertation, Zhengzhou University (2012)
34. Bello Cruz, J.Y., Nghia Tran, T.A.: On the convergence of the forward-backward splitting method with linesearches. Optim. Method Softw. **31**(6), 1209–1238 (2016)
35. Dong, Y.D.: The proximal point algorithm revisited. J. Optim. Theory Appl. **161**, 478–489 (2014)
36. Dong, Y.D.: Comments on "the proximal point algorithm revisited". J. Optim. Theory Appl. **166**, 343–349 (2015)
37. Dong, Y.D.: Douglas-Rachford splitting method for semi-definite programming. J. Appl. Math. Comput. **51**, 569–591 (2016)

38. Kanzow, C., Shehu, Y.: Generalized Krasnoselskii-Mann-type iterations for nonexpansive mappings in Hilbert spaces. Comput. Optim. Appl. **67**(3), 595–620 (2017)
39. Matsushita, S.Y.: On the convergence rate of the Krasnosel'skiĭ-Mann iteration. Bull. Austr. Math. Soc. **96**(1), 162–170 (2017)
40. Solodov, M.V., Tseng, P.: Modified projection-type methods for monotone variational inequalities. SIAM J. Control Optim. **34**, 1814–1830 (1996)
41. Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. Johns Hopkins University Press, Baltimore Maryland (1996)
42. Moreau, J.J.: Proximité et dualité dans un espace Hilbertien. Bull. Soc. Math. Fr. **93**, 273–299 (1965)
43. Rockafellar, R.T.: Monotone operators and the proximal point algorithm. SIAM J. Control Optim. **14**, 877–898 (1976)
44. Browder, F.E.: Convergence theorems for sequences of nonlinear operators in Banach spaces. Math. Z. **100**, 201–225 (1967)
45. Brézis, H., Lions, P.L.: Produits infinis de resolvantes. Isr. J. Math. **29**, 329–345 (1978)
46. Luque, F.J.: Asymptotic convergence analysis of the proximal point algorithm. SIAM J. Control Optim. **22**, 277–293 (1984)
47. Güler, O.: On the convergence of the proximal point algorithm for convex minimization. SIAM J. Control Optim. **29**, 403–419 (1991)
48. Pennanen, T.: Local convergence of the proximal point algorithm and multiplier methods without monotonicity. Math. Oper. Res. **27**, 193–202 (2002)
49. Dong, Y.D.: A new relative error criterion for the proximal point algorithm. Nonlinear Anal. **64**, 2143–2148 (2006)
50. Dafermos, S.: Traffic equilibrium and variational inequalities. Transp. Sci. **14**, 42–54 (1980)
51. Dong, Y.D.: New inertial factors of the Krasnosel'skiĭ-Mann iteration. Set-Valued Var. Anal. (2020). https://doi.org/10.1007/s11228-020-00541-5