Available online at www.sciencedirect.com
ScienceDirect

Mathematics and Computers in Simulation 203 (2023) 518–537

MATHEMATICS AND COMPUTERS IN SIMULATION

www.elsevier.com/locate/matcom

Original articles

# A new splitting method for systems of monotone inclusions in Hilbert spaces

Yunda Dong

*School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, PR China*

Received 28 July 2021; received in revised form 29 April 2022; accepted 23 June 2022
Available online 30 June 2022

**Abstract**

In this article, we consider the problem of finding a zero of systems of monotone inclusions in real Hilbert spaces. Furthermore, each monotone inclusion consists of three operators and the third is linearly composed. We suggest a splitting method for solving them: At each iteration, for each monotone inclusion, it mainly needs computations of three resolvents for individual operator. This method can be viewed as a powerful extension of the classical Douglas–Rachford splitting. Under the weakest possible assumptions, by introducing and using the characteristic operator, we analyze its weak convergence. The most striking feature is that it merely requires each scaling factor for individual operator be positive. Numerical results indicate practical usefulness of this method, together with its special cases, in solving our test problems of separable structure.


*Keywords:* Monotone inclusion; Characteristic operator; Splitting methods; Weak convergence; Scaling factor

## 1. Introduction

For $i = 1, \ldots, n$, let $\mathcal{H}_i$ and $\mathcal{G}$ be real infinite-dimensional Hilbert spaces. In this article, we are mainly concerned with the following system of monotone inclusions

$$0 \in \bar{A}_i(x_i) + A_i(x_i) + Q_i^* B\left(\sum_{i=1}^n Q_i x_i - q\right),\ i = 1, \ldots, n, \tag{1}$$

where $\bar{A}_i,\ A_i : \mathcal{H}_i \rightrightarrows \mathcal{H}_i$ are maximal monotone operators, $B : \mathcal{G} \rightrightarrows \mathcal{G}$ is maximal monotone operator, and each $Q_i : \mathcal{H}_i \to \mathcal{G}$ is nonzero bounded linear operator with its adjoint operator $Q_i^*$, and $q \in \mathcal{G}$ is a vector. The problem above models a wide range of problems arising from definite linear systems, linear/quadratic programming, complementarity problems, variational inequality problems and optimal control [19].

For an important case

$$0 \in \bar{A}(x) + A(x), \tag{2}$$

one may resort to the Douglas–Rachford splitting method (DR method for short) of Lions and Mercier [13]; see the Refs. [2,3,5,7,10,20] and Algorithm 3.

In the $n = 1$ case, the author discussed how to solve semi-definite programming by DR method and a modified version in a 2010 conference report, which is an early draft of [2].

*E-mail address:* ydong@zzu.edu.cn.

https://doi.org/10.1016/j.matcom.2022.06.023

In general $n$ case, He and Han [9] considered a class of separable convex minimization problems with linear constraints in Euclidean spaces, which is shown to be a special case of (1) below, and proposed an iterative scheme. Their proposed method contains the ingredient of the DR splitting, and at each iteration it merely needs to solve much easier subproblems.

Inspired by these two works, we in this article consider how to further apply the DR splitting to solving more general problem (1) above. As a result, we suggest a new splitting method. Under the weakest possible assumptions, by introducing and using the characteristic operator (see Lemma 1 below for more details), we analyze its weak convergence. One of the most striking features is that it merely requires each scaling factor for individual operator be positive.

For our suggested splitting method, it just reduces to the classical Douglas–Rachford splitting, when applied to solving the two-operator monotone inclusion (2) above.

Interestingly, in a very special case of $B$ being the sub-differential of the indicator function of the singleton set $\{0\}$ (and others), our suggested splitting method coincides with an equivalent version of the algorithm proposed by He and Han [9] . Yet, even in this case, our way of choosing the parameter involved in (37) is a new idea. It is given by (38) and essentially different from theirs.

Our suggested splitting method is reminiscent of an extended splitting method of [4, Algorithm 3.1], which is best suited for solving

$$0 \in C(x) + A(x) + Q^* B\,(Qx - q)\,, \tag{3}$$

corresponding to (1) with $n = 1$ and $\bar{A} := C$ being further inverse strongly monotone. In contrast, its novelties include: (i) There is no need to evaluate the constant with respect to inverse strong monotonicity; (ii) its scope of applications is much wider and it can solve all four test problems whereas the method of [4, Algorithm 3.1] fails to do so; see Section 7 for more details.

For our suggested splitting method, we also did rudimentary experiments to confirm that it, together with its special cases, is easily implementable and practically efficient for our test problems.

## 2. Preliminaries

In this section, we first give some basic definitions and then provide some auxiliary results for later use. Moreover, we formally state general systems of monotone inclusions in real Hilbert spaces under consideration.

Let $Q : \mathcal{H} \to \mathcal{G}$ be nonzero bounded linear operator with its adjoint operator $Q^*$. Then its norm $\|Q\|$ is given by

$$\|Q\| := \max\left\{\sqrt{\langle u, Q^*Qu\rangle} : \|u\| = 1,\ u \in \mathcal{H}\right\},$$

where $\langle\cdot,\cdot\rangle$ stands for usual inner product. If in finite-dimensional spaces, then $Q$ becomes an $m$-by-$n$ matrix. Thus, it is well-known that $\|Q\|^2 \le \|Q\|_1 \cdot \|Q\|_\infty$, where

$$\|Q\|_1 := \max_{j=1,\ldots,n} \textstyle\sum_{i=1}^m |q_{ij}|, \quad \|Q\|_\infty := \max_{i=1,\ldots,m} \textstyle\sum_{j=1}^n |q_{ij}|.$$

**Definition 1.** Let $\mathcal{H}$ be a real Hilbert space. 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

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

Each element $s$ 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 $x$.

To concisely give the following definition, we agree on that the notation $(x, a) \in A$ and $x \in \mathcal{H}$, $a \in A(x)$ have the same meaning.

**Definition 2.** Let $A : \mathcal{H} \rightrightarrows \mathcal{H}$ be an operator. It is called monotone iff

$$\langle x - x',\ a - a'\rangle \ge 0, \qquad \forall (x, a) \in A,\ \ \forall (x', a') \in A;$$

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

$$\langle x - \hat{x},\ a - \hat{a}\rangle \ge 0, \quad \forall (x, a) \in A \quad \Rightarrow \quad (\hat{x}, \hat{a}) \in A.$$

**Definition 3.** A single-valued operator $A : \mathcal{H} \to \mathcal{H}$ is called Lipschitz continuous with modulus $\kappa > 0$ if

$$\|A(x) - A(y)\| \leq \kappa \|x - y\|$$

holds for all $x, y \in \mathcal{H}$.

**Definition 4.** Let $C : \mathcal{H} \to \mathcal{H}$ be an operator. $C$ is called inverse strongly monotone if there exists some $c > 0$ such that

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

It is well known that the sub-differential of any closed, proper and convex function in a real Hilbert space is maximal monotone as well. An important instance is the indicator function $\delta_{\mathcal{C}}$ of a convex subset $\mathcal{C}$ in a real Hilbert space

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

The closedness of $\delta_{\mathcal{C}}$ is equivalent to the closedness of $C$. Thus, $\partial\delta_{\mathcal{C}}$ to $\mathcal{C}$ is maximal monotone when $\mathcal{C}$ is closed convex. Furthermore, the inverting operation of $I + \mu\partial\delta_{\mathcal{C}}$ equals the usual projection $P_{\mathcal{C}}$ onto the set $\mathcal{C}$ for any given positive number $\mu$. In addition, if we set $\mathcal{R}^n_+ := \{x \in \mathcal{R}^n : x \geq 0\}$, then $P_{\mathcal{R}^n_+}(x) = \max\{0, x\}$, which is of component-wise maximum.

For any given maximal monotone operator $A : \mathcal{H} \rightrightarrows \mathcal{H}$, its effective domain dom$A$ is defined by dom$A := \{x \in \mathcal{H} : A(x) \neq \emptyset\}$. A related property is that if $x \in \text{dom}A$ then the set $A(x)$ must be a nonempty closed convex set. A fundamental property is that, as proved by Minty [16], for any given positive number $\alpha > 0$ and $\hat{x} \in \mathcal{H}$, there exists a unique $x \in \mathcal{H}$ such that $(\alpha I + A)(x) \ni \hat{x}$ or $(I + \alpha A)(x) \ni \hat{x}$.

For any given maximal monotone operator $A : \mathcal{H} \rightrightarrows \mathcal{H}$, the basic iterative procedure of finding its zeros is the following proximal point algorithm [14,19]: Choose an initial point $x^0 \in \mathcal{H}$, and solve the monotone inclusion

$$(I + \mu_k A)(x) \ni x^k, \quad k = 0, 1, \ldots$$

to get the new iterate $x^{k+1}$, where the proximal parameter $\mu_k > 0$ can vary from iteration to iteration. For very recent discussions, we refer to [3,10,15,20] and the references cited therein.

Denote

$$x := \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad a := \begin{pmatrix} a_1 \\ \vdots \\ a_n \end{pmatrix}, \quad A := \begin{pmatrix} A_1 & & \\ & \ddots & \\ & & A_n \end{pmatrix}.$$

**Lemma 1.** *For systems of monotone inclusions (1), we introduce the dual variable $u \in \mathcal{G}$. Then*

$$T(x, a, u) := \begin{pmatrix} \bar{A} & & \\ & A^{-1} & \\ & & B^{-1} \end{pmatrix} \begin{pmatrix} x \\ a \\ u \end{pmatrix} + \begin{pmatrix} 0 & I & Q^* \\ -I & 0 & 0 \\ -Q & 0 & 0 \end{pmatrix} \begin{pmatrix} x \\ a \\ u \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ q \end{pmatrix} \tag{4}$$

*must be maximal monotone.*

**Proof.** Note that $\bar{A}$, $A$ and $B$ are maximal monotone. Thus, the first operator on the right-hand side is maximal monotone. Meanwhile, the linearity of $Q$ means that the second is also maximal monotone [16]. Maximality of $T$ follows from [18]. □

In this article, such $T$ is called the characteristic operator or eigenoperator with respect to the problem (1) above.

## 3. Main results

In this section, we describe our suggested splitting method for systems of monotone inclusions (1) in details.

The method's design is based on the following

**Assumption 1.** For system of monotone inclusions (1), we assume that there exist $x_1^* \in \mathcal{H}_1$, ...,$x_n^* \in \mathcal{H}_n$, $x_{n+1}^* \in \mathcal{G}$, $u^* \in \mathcal{G}$ such that they solve

$$0 \in \bar{A}_i(x_i) + A_i(x_i) + Q_i^* u, \quad i = 1, \ldots, n, \tag{5}$$

$$0 \in B(x_{n+1}) - u, \tag{6}$$

$$0 = \textstyle\sum_{i=1}^n Q_i x_i - q - x_{n+1}. \tag{7}$$

Furthermore, $\emptyset \neq \mathrm{dom}\bar{A}_i \subseteq \mathrm{dom}A_i$ for $i = 1, \ldots, n$, $\emptyset \neq \mathrm{dom}B$.

First of all, we would like to explain Assumption 1 a bit. For example, we consider

$$\min \ \bar{f}(x) + f(x) + g(Qx - q),$$

where $\bar{f}, f : \mathcal{R}^n \to \mathcal{R}$, $g : \mathcal{R}^m \to \mathcal{R}$ are closed, proper convex functions, $Q$ is an $m \times n$ matrix and $q \in \mathcal{R}^m$. If there exists an $x$ such that

$$x \in \mathrm{ri}\ \mathrm{dom}\bar{f} \cap \mathrm{ri}\ \mathrm{dom}f, \quad Qx - q \in \mathrm{ri}\ \mathrm{dom}g, \tag{8}$$

then its optimality condition is

$$0 \in \partial\bar{f}(x) + \partial f(x) + Q^T \partial g(Qx - q),$$

where ri stands for the relative interior; see [17] for more details. If $g$ is taken to be the indicator function $\delta_{\{0\}}$, then (8) reduces to

$$x \in \mathrm{ri}\ \mathrm{dom}\bar{f} \cap \mathrm{ri}\ \mathrm{dom}f, \quad Qx - q = 0 \tag{9}$$

because the set ri dom$g$ becomes $\{0\}$, and we further have

$$0 \in \partial\bar{f}(x) + \partial f(x) + Q^T \partial\delta_{\{0\}}(Qx - q).$$

Of course, we may replace (9) by

$$x \in \mathrm{int}\ \mathrm{dom}\bar{f} \cap \mathrm{int}\ \mathrm{dom}f, \quad Qx - q = 0, \tag{10}$$

where int stands for the interior. This is stronger but more convenient, and it suffices to cover optimality conditions of the first, second and fourth test problems.

Below we discuss how to use these conditions (5), (6) and (7) to design an iterative scheme for solving them. For known $x_i^k \in \mathcal{H}_i$, $a_i^k \in A_i(x_i^k)$, $i = 1, \ldots, n$, $x_{n+1}^k \in \mathcal{G}$, $u^k \in \mathcal{G}$. We first choose $\beta > 0$ and update the dual iterate by

$$\bar{u}^k = u^k - (x_{n+1}^k - \textstyle\sum_{i=1}^n Q_i x_i^k + q)/\beta$$

to get the intermediate point $\bar{u}^k$. Such an idea, at least in the setting of the Douglas–Rachford splitting method, first appeared in the aforementioned 2010 conference report. It plays a critical role in designing and analyzing the method. Then, we compute

$$(\alpha_i I + \bar{A}_i)(\bar{x}_i^k) \ni \alpha_i x_i^k - a_i^k - Q_i^* \bar{u}^k, \quad a_i^k \in A_i(x_i^k),$$
$$(\alpha_{n+1} I + B)(\bar{x}_{n+1}^k) \ni \alpha_{n+1} x_{n+1}^k + \bar{u}^k,$$

to get the intermediate points $\bar{x}_i^k$ and $\bar{x}_{n+1}^k$.

Denote

$$w := \begin{pmatrix} \alpha_1 x_1 + a_1 \\ \vdots \\ \alpha_{n+1} x_{n+1} + a_{n+1} \\ u \end{pmatrix}, \quad d := \begin{pmatrix} x_1 - \bar{x}_1 \\ \vdots \\ x_{n+1} - \bar{x}_{n+1} \\ -\sum_{i=1}^{n+1} Q_i \bar{x}_i + q \end{pmatrix}, \tag{11}$$

where

$$a_{n+1} := 0, \quad Q_{n+1} := -I. \tag{12}$$

We can prove that, if $\beta$ is properly chosen, then the following desired inequality holds $\langle w^k - w^*, d^k \rangle > 0$, where $w^*$ corresponds to the primal–dual solution. This indicates that $-d^k$ can provide a descent direction of $\|w - w^*\|^2$ at $w = w^k$. So, it is not difficult to get the following Algorithm 1.

**Algorithm 1.** Our suggested splitting algorithm

Step 0. For $i = 1, \ldots, n$, choose $x_i^0 \in \mathcal{H}_i$, $a_i^0 \in A_i(x_i^0)$, $x_{n+1}^0 \in \mathcal{G}$, $u^0 \in \mathcal{G}$. For $i = 1, \ldots, n+1$, choose $\alpha_i > 0$ and $\theta \in (0, 2)$. Set $k := 0$.

Step 1. Choose $\beta$ satisfying

$$\beta > \textstyle\sum_{i=1}^n \|Q_i\|^2/(4\alpha_i) + 1/(4\alpha_{n+1}). \tag{13}$$

For $x_i^k \in \mathcal{H}_i$, $a_i^k \in A_i(x_i^k)$, $i = 1, \ldots, n$, $x_{n+1}^k \in \mathcal{G}$, $u^k \in \mathcal{G}$. Compute

$$\bar{u}^k = u^k - (x_{n+1}^k - \textstyle\sum_{i=1}^n Q_i x_i^k + q)/\beta, \tag{14}$$

$$(\alpha_i I + \bar{A}_i)(\bar{x}_i^k) \ni \alpha_i x_i^k - a_i^k - Q_i^* \bar{u}^k, \tag{15}$$

$$(\alpha_{n+1} I + B)(\bar{x}_{n+1}^k) \ni \alpha_{n+1} x_{n+1}^k + \bar{u}^k. \tag{16}$$

If some stopping criterion is met, then stop. Otherwise, compute

$$\begin{aligned}
\phi_k &:= \textstyle\sum_{i=1}^{n+1} \alpha_i \|x_i^k - \bar{x}_i^k\|^2 + \langle \bar{x}_{n+1}^k - \sum_{i=1}^n Q_i \bar{x}_i^k + q,\ u^k - \bar{u}^k \rangle,\\
\varphi_k &:= \textstyle\sum_{i=1}^{n+1} \|x_i^k - \bar{x}_i^k\|^2 + \|\bar{x}_{n+1}^k - \sum_{i=1}^n Q_i \bar{x}_i^k + q\|^2,\\
\gamma_k &:= \theta \phi_k / \varphi_k.
\end{aligned} \tag{17}$$

Step 2. For $i = 1, \ldots, n$, compute in order

$$(\alpha_i I + A_i)(x_i^{k+1}) \ni \alpha_i x_i^k + a_i^k - \gamma_k (x_i^k - \bar{x}_i^k), \tag{18}$$

$$\alpha_{n+1} x_{n+1}^{k+1} = \alpha_{n+1} x_{n+1}^k - \gamma_k (x_{n+1}^k - \bar{x}_{n+1}^k), \tag{19}$$

$$u^{k+1} = u^k - \gamma_k (\bar{x}_{n+1}^k - \textstyle\sum_{i=1}^n Q_i \bar{x}_i^k + q),$$

$$a_i^{k+1} = \alpha_i (x_i^k - x_i^{k+1}) + a_i^k - \gamma_k (x_i^k - \bar{x}_i^k). \tag{20}$$

Set $k := k + 1$.

If it is not easy to evaluate $\|Q_i\|^2$ in practice, then we may turn to consider replacing it with $\|Q\|_1 \cdot \|Q\|_\infty$ as an alternative. In practical implementations, we usually calculate

$$\begin{aligned}
\Delta &:= \textstyle\sum_{i=1}^n \|Q_i\|_1 \|Q_i\|_\infty/(4\alpha_i) + 1/(4\alpha_{n+1}),\\
\Delta' &:= \min\{\|Q_1\|_1\|Q_1\|_\infty/(4\alpha_1), \ldots, \|Q_n\|_1\|Q_n\|_\infty/(4\alpha_n), 1/(4\alpha_{n+1})\}
\end{aligned}$$

and then we may choose

$$\beta = \kappa \Delta + 10^{-9} \Delta', \quad \kappa \geq 1. \tag{21}$$

In Algorithm 1, there are sub-problems (cf. (15) and (18)) of the following type

$$(\alpha I + A)(x) \ni w.$$

Now we discuss how to solve it. (i) If $A$ is further linear, then we may use Matlab solver via

$$x = (\alpha I + A) \setminus w. \tag{22}$$

(ii) If $A := \nabla f$ is the gradient of some continuously differentiable convex function $f$, then we may resort to quasi-Newton method with novel conditions using gradient only to locate steplength; see [22, Sect. 5.3]. (iii) If $A := F$ is continuously differentiable, then we may use some Newton-type method to solve this sub-problem. In (15), there are $a_i^k \in A_i(x_i^k)$, $i = 1, \ldots, n$. If $k = 0$, then each $a_i^0$ is chosen from the set $A_i(x_i^0)$, as stated above. If $k \geq 0$, then we shall make use of (20) to calculate each $a_i^{k+1}$, and the computational cost is tiny.

In Algorithm 1, there is also sub-problem (16) of the following type

$$(\alpha I + B)(y) \ni p.$$

For (16), if $B$ is taken to be the differential of the indictor function of some closed convex set $\mathcal{C}$, then it becomes

$$\bar{x}_{n+1}^k = P_{\mathcal{C}}(\alpha_{n+1} x_{n+1}^k + \bar{u}^k).$$

Of course, if $\mathcal{C}$ is taken to be $\mathcal{R}^n_+ := \{x \in \mathcal{R}^n : x \geq 0\}$, then it further reduces to

$$\bar{x}^k_{n+1} = \max\left\{0,\ \alpha_{n+1}x^k_{n+1} + \bar{u}^k\right\},$$

where 0 is $n$-dimensional zero vector. In particular, in the case of $\mathcal{C} = \{0\}$, we always have $\bar{x}^k_{n+1} = 0$.

## 4. Weak convergence

In this section, we analyze convergence properties of the primal sequence and the dual sequence generated by Algorithm 1. Under the weakest possible assumptions, we prove the former's weak convergence to a solution of the problem (1).

To simplify the proof of our main theorem, we introduce the following lemma, which is new and extends a result in the proof of [12] from $n = 1$ to general cases.

**Lemma 2.** *For $i = 1, \ldots, n$, let $Q_i : \mathcal{H}_i \to \mathcal{G}$ be nonzero bounded linear operators, and let $\alpha_i > 0$. If $\beta_i > \|Q_i\|^2/(4\alpha_i)$, then the following*

$$\begin{aligned}&\textstyle\sum_{i=1}^n \left(\alpha_i\|x_i\|^2 + \langle Q_i x_i, u\rangle + \beta_i\|u\|^2\right)\\ &\geq \frac{1}{2}\textstyle\sum_{i=1}^n \left(\alpha_i + n\beta_i - \sqrt{n\|Q_i\|^2 + (\alpha_i - n\beta_i)^2}\right)\left(\|x_i\|^2 + n^{-1}\|u\|^2\right)\end{aligned}$$

*holds for all $x_i \in \mathcal{H}_i$ and all $u \in \mathcal{G}$.*

Very recently, such a nice result was used in [3,6].

**Lemma 3.** *Consider any maximal monotone operator $T : \mathcal{H} \to \mathcal{H}$. 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.*

**Theorem 1.** *Let $\{x^k_i\}(i = 1, \ldots, n+1)$, $\{u^k\}$ be the sequences generated by Algorithm 1. If Assumption 1 holds and $\beta := \sum_{i=1}^{n+1} \beta_i$ and*

$$\beta_i > \frac{\|Q_i\|^2}{4\alpha_i},\ i = 1, \ldots, n, \quad \beta_{n+1} > \frac{1}{4\alpha_{n+1}}, \tag{23}$$

*then there exists some positive number $\hat{\gamma}$ such that*

$$\begin{aligned}&\|w^{k+1} - w^*\|^2\\ &\leq \|w^k - w^*\|^2 - \hat{\gamma}\left(\textstyle\sum_{i=1}^{n+1}\|x^k_i - \bar{x}^k_i\|^2 + \|u^k - \bar{u}^k\|^2\right),\end{aligned} \tag{24}$$

*where $w$ is defined in (11).*

**Proof.** For $i = 1, \ldots, n$, it follows from (15) that

$$\bar{A}_i(\bar{x}^k_i) \ni \alpha_i(x^k_i - \bar{x}^k_i) - a^k_i - Q^*_i\bar{u}^k,$$

which, together with (5)

$$\bar{A}_i(x^*_i) \ni -a^*_i - Q^*_i u^*,$$

and monotonicity of each $\bar{A}_i$, imply

$$\begin{aligned}0 &\leq \langle \bar{x}^k_i - x^*_i,\ \alpha_i(x^k_i - \bar{x}^k_i) - (a^k_i - a^*_i) - Q^*_i(\bar{u}^k - u^*)\rangle\\ &= \langle \bar{x}^k_i - x^*_i,\ \alpha_i(x^k_i - \bar{x}^k_i) - (a^k_i - a^*_i)\rangle\\ &\quad - \langle \bar{x}^k_i - x^*_i,\ Q^*_i(\bar{u}^k - u^*)\rangle\\ &= \langle x^k_i - x^*_i - (x^k_i - \bar{x}^k_i),\ \alpha_i(x^k_i - \bar{x}^k_i) - (a^k_i - a^*_i)\rangle\\ &\quad - \langle \bar{x}^k_i - x^*_i,\ Q^*_i(\bar{u}^k - u^*)\rangle.\end{aligned}$$

Rearranging all terms yields

$$\begin{aligned}&\langle \alpha_i x_i^k + a_i^k - (\alpha_i x_i^* + a_i^*),\ x_i^k - \bar{x}_i^k\rangle - \langle Q_i(\bar{x}_i^k - x_i^*),\ \bar{u}^k - u^*\rangle\\ &\geq \alpha_i \|x_i^k - \bar{x}_i^k\|^2 + \langle x_i^k - x_i^*,\ a_i^k - a_i^*\rangle\\ &\geq \alpha_i \|x_i^k - \bar{x}_i^k\|^2,\end{aligned} \tag{25}$$

where the last inequality follows from monotonicity of each $A_i$.

It follows from (16) that

$$B(\bar{x}_{n+1}^k) \ni \alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k) + \bar{u}^k,$$

which, together with (6)

$$B(x_{n+1}^*) \ni u^*,$$

and monotonicity of $B$, imply

$$\begin{aligned}0 &\leq \langle \bar{x}_{n+1}^k - x_{n+1}^*,\ \alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k) + \bar{u}^k - u^*\rangle\\ &= \langle \bar{x}_{n+1}^k - x_{n+1}^*,\ \alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k)\rangle\\ &\quad + \langle \bar{x}_{n+1}^k - x_{n+1}^*,\ \bar{u}^k - u^*\rangle\\ &= \langle x_{n+1}^k - x_{n+1}^* - (x_{n+1}^k - \bar{x}_{n+1}^k),\ \alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k)\rangle\\ &\quad + \langle \bar{x}_{n+1}^k - x_{n+1}^*,\ \bar{u}^k - u^*\rangle.\end{aligned}$$

Rearranging all terms yields

$$\begin{aligned}&\langle x_{n+1}^k - x_{n+1}^*,\ \alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k)\rangle + \langle \bar{x}_{n+1}^k - x_{n+1}^*,\ \bar{u}^k - u^*\rangle\\ &\geq \alpha_{n+1}\|x_{n+1}^k - \bar{x}_{n+1}^k\|^2.\end{aligned}$$

Combining this with (25) yields

$$\begin{aligned}&\textstyle\sum_{i=1}^n \langle \alpha_i x_i^k + a_i^k - (\alpha_i x_i^* + a_i^*),\ x_i^k - \bar{x}_i^k\rangle\\ &\quad + \langle x_{n+1}^k - x_{n+1}^*,\ \alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k)\rangle - \textstyle\sum_{i=1}^{n+1}\langle Q_i(\bar{x}_i^k - x_i^*),\ \bar{u}^k - u^*\rangle\\ &\geq \textstyle\sum_{i=1}^{n+1}\alpha_i\|x_i^k - \bar{x}_i^k\|^2,\end{aligned}$$

which, together with $\sum_{i=1}^{n+1} Q_i x_i^* = q$ (see (7) and (12)), implies

$$\begin{aligned}&\textstyle\sum_{i=1}^n \langle \alpha_i x_i^k + a_i^k - (\alpha_i x_i^* + a_i^*),\ x_i^k - \bar{x}_i^k\rangle\\ &\quad + \langle x_{n+1}^k - x_{n+1}^*,\ \alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k)\rangle - \langle \textstyle\sum_{i=1}^{n+1} Q_i\bar{x}_i^k - q,\ \bar{u}^k - u^*\rangle\\ &\geq \textstyle\sum_{i=1}^{n+1}\alpha_i\|x_i^k - \bar{x}_i^k\|^2,\end{aligned}$$

namely

$$\begin{aligned}&\textstyle\sum_{i=1}^n \langle \alpha_i x_i^k + a_i^k - (\alpha_i x_i^* + a_i^*),\ x_i^k - \bar{x}_i^k\rangle\\ &\quad + \langle u^k - u^*,\ -\textstyle\sum_{i=1}^{n+1} Q_i\bar{x}_i^k + q\rangle + \langle \alpha_{n+1}(x_{n+1}^k - x_{n+1}^*),\ x_{n+1}^k - \bar{x}_{n+1}^k\rangle\\ &\geq \textstyle\sum_{i=1}^{n+1}\alpha_i\|x_i^k - \bar{x}_i^k\|^2 - \langle \textstyle\sum_{i=1}^{n+1} Q_i\bar{x}_i^k - q,\ u^k - \bar{u}^k\rangle,\end{aligned}$$

which can be rewritten as

$$\langle w^k - w^*, d^k\rangle \geq \phi_k, \tag{26}$$

where $w$ and $d$ are defined in (11).

On the other hand, by (19) and (26), we get

$$\begin{aligned}&\|w^{k+1} - w^*\|^2\\ &= \|w^k - w^* - \gamma_k d^k\|^2\end{aligned}$$

$$
\begin{aligned}
&= \|w^k - w^*\|^2 - 2\gamma_k \langle w^k - w^*, d^k\rangle + \gamma_k^2 \|d^k\|^2 \\
&= \|w^k - w^*\|^2 - 2\gamma_k \langle w^k - w^*, d^k\rangle + \gamma_k^2 \varphi_k \\
&\le \|w^k - w^*\|^2 - 2\gamma_k \phi_k + \gamma_k^2 \varphi_k,
\end{aligned}
$$

which, together with (17), implies

$$
\|w^{k+1} - w^*\|^2 \le \|w^k - w^*\|^2 - (2-\theta)\gamma_k \phi_k. \tag{27}
$$

Since

$$
\begin{aligned}
&\bar{x}_{n+1}^k - \textstyle\sum_{i=1}^n Q_i \bar{x}_i^k + q \\
&= x_{n+1}^k - \textstyle\sum_{i=1}^n Q_i x_i^k + q + \bar{x}_{n+1}^k - x_{n+1}^k + \sum_{i=1}^n Q_i (x_i^k - \bar{x}_i^k) \\
&= \beta(u^k - \bar{u}^k) + \bar{x}_{n+1}^k - x_{n+1}^k + \textstyle\sum_{i=1}^n Q_i (x_i^k - \bar{x}_i^k)
\end{aligned} \tag{28}
$$

and $\beta := \sum_{i=1}^{n+1} \beta_i$ (see Theorem 1), it follows from Lemma 2 that

$$
\begin{aligned}
&\phi_k \\
&= \textstyle\sum_{i=1}^{n+1} \alpha_i \|x_i^k - \bar{x}_i^k\|^2 + \beta \|u^k - \bar{u}^k\|^2 + \sum_{i=1}^{n+1} \langle Q_i(x_i^k - \bar{x}_i^k),\ u^k - \bar{u}^k\rangle \\
&= \textstyle\sum_{i=1}^{n+1} \left(\alpha_i \|x_i^k - \bar{x}_i^k\|^2 + \langle Q_i(x_i^k - \bar{x}_i^k),\ u^k - \bar{u}^k\rangle\right) + \sum_{i=1}^{n+1} \beta_i \|u^k - \bar{u}^k\|^2 \\
&\ge \frac{1}{2} \textstyle\sum_{i=1}^{n+1} \left(\|x_i^k - \bar{x}_i^k\|^2 + (n+1)^{-1} \|u^k - \bar{u}^k\|^2\right) \\
&\quad \left(\alpha_i + (n+1)\beta_i - \sqrt{(n+1)\|Q_i\|^2 + (\alpha_i - (n+1)\beta_i)^2}\right),
\end{aligned}
$$

and the conditions (23) indicate that each

$$
\frac{1}{2}\left(\alpha_i + (n+1)\beta_i - \sqrt{(n+1)\|Q_i\|^2 + (\alpha_i - (n+1)\beta_i)^2}\right)
$$

must be positive. Let $\rho$ be their minimum. Thus, we further get

$$
\phi_k \ge \rho \left(\textstyle\sum_{i=1}^{n+1} \|x_i^k - \bar{x}_i^k\|^2 + \|u^k - \bar{u}^k\|^2\right). \tag{29}
$$

Meanwhile, we get

$$
\begin{aligned}
&\varphi_k \\
&= \textstyle\sum_{i=1}^{n+1} \|x_i^k - \bar{x}_i^k\|^2 + \|\bar{x}_{n+1}^k - \sum_{i=1}^n Q_i \bar{x}_i^k + q\|^2 \\
&= \textstyle\sum_{i=1}^{n+1} \|x_i^k - \bar{x}_i^k\|^2 + \|\beta(u^k - \bar{u}^k) + \sum_{i=1}^{n+1} Q_i (x_i^k - \bar{x}_i^k)\|^2 \\
&\le \textstyle\sum_{i=1}^{n+1} \|x_i^k - \bar{x}_i^k\|^2 + \left(\beta^2 + \sum_{i=1}^{n+1} \|Q_i\|^2\right)\left(\|u^k - \bar{u}^k\|^2 + \sum_{i=1}^{n+1} \|x_i^k - \bar{x}_i^k\|^2\right) \\
&\le \left(1 + \beta^2 + \textstyle\sum_{i=1}^{n+1} \|Q_i\|^2\right)\left(\sum_{i=1}^{n+1} \|x_i^k - \bar{x}_i^k\|^2 + \|u^k - \bar{u}^k\|^2\right).
\end{aligned}
$$

Thus, we can conclude that

$$
\gamma_k = \theta \phi_k / \varphi_k \ge \frac{\theta\rho}{1 + \beta^2 + \sum_{i=1}^{n+1} \|Q_i\|^2} > 0. \tag{30}
$$

Combining this with (27) and (29) yields the desired result. □

Note that, the relation (30) tells us that the sequence $\{\gamma_k\}$ has a positive lower bound, which is a desirable property and is beneficial to numerical stability.

**Theorem 2.** *Assume that Assumption 1 holds. Let $\{x_i^k\}(i = 1, \ldots, n+1)$, $\{u^k\}$ be the sequences generated by Algorithm 1, respectively. If the conditions (23) hold, then the corresponding primal sequences $\{x_i^k\}(i = 1, \ldots, n)$ weakly converge to a solution of system of monotone inclusions (1) mentioned above.*

**Proof.** It follows from (24) that

$$
(i)\ \ x_i^k - \bar{x}_i^k \to 0,\ i = 1, \ldots, n+1,\ \ u^k - \bar{u}^k \to 0; \tag{31}
$$

$$(ii)\ \{(x_i^k, a_i^k)\},\ \{x_{n+1}^k\},\ \{u^k\} \text{ are bounded in norm} \tag{32}$$

for $i = 1, \ldots, n$.

Next, we will make use of Lemma 3 to prove the remaining part. To this end, we follow the definition of the set $T$ to get

$$\begin{aligned} &T(\bar{x}^k, a^k, \alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k) + \bar{u}^k) \\ &= \begin{pmatrix} \vdots \\ \bar{A}_i(\bar{x}_i^k) + a_i^k + Q_i^*(\alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k) + \bar{u}^k) \\ \vdots \\ A_i^{-1}(a_i^k) - \bar{x}_i^k \\ \vdots \\ B^{-1}(\alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k) + \bar{u}^k) - \sum_{i=1}^n Q_i \bar{x}_i^k + q \end{pmatrix}. \end{aligned}$$

Meanwhile, by (15) and (16), we further have

$$\begin{aligned} &\bar{A}_i(\bar{x}_i^k) + a_i^k + Q_i^*(\alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k) + \bar{u}^k) \\ &\ni \alpha_i(x_i^k - \bar{x}_i^k) - a_i^k - Q_i^*\bar{u}^k + a_i^k + Q_i^*(\alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k) + \bar{u}^k) \\ &= \alpha_i(x_i^k - \bar{x}_i^k) + \alpha_{n+1}Q_i^*(x_{n+1}^k - \bar{x}_{n+1}^k) \end{aligned}$$

and

$$\begin{aligned} &B^{-1}\left(\alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k) + \bar{u}^k\right) - \textstyle\sum_{i=1}^n Q_i \bar{x}_i^k + q \\ &\ni \bar{x}_{n+1}^k - \textstyle\sum_{i=1}^n Q_i \bar{x}_i^k + q, \end{aligned}$$

respectively. So, the set

$$T(\bar{x}^k, a^k, \alpha_{n+1}(x_{n+1}^k - \bar{x}_{n+1}^k) + \bar{u}^k)$$

includes

$$\begin{pmatrix} \vdots \\ \alpha_i(x_i^k - \bar{x}_i^k) + \alpha_{n+1}Q_i^*(x_{n+1}^k - \bar{x}_{n+1}^k) \\ \vdots \\ x_i^k - \bar{x}_i^k \\ \vdots \\ \bar{x}_{n+1}^k - \sum_{i=1}^n Q_i \bar{x}_i^k + q \end{pmatrix},$$

which strongly converges to zero due to (28) and (31) and boundedness of each $Q_i$. On the other hand, we shall check weak convergence of each of the involved sequences. In fact, according to (32), there exists one weak cluster point such that

$$(x^{k_j}, a^{k_j}) \rightharpoonup (x^\infty, a^\infty), \quad u^{k_j} \rightharpoonup u^\infty,$$

which, together with (31), implies

$$\bar{x}^{k_j} \rightharpoonup x^\infty,\ a^{k_j} \rightharpoonup a^\infty,\ \alpha_{n+1}(x_{n+1}^{k_j} - \bar{x}_{n+1}^{k_j}) + \bar{u}^{k_j} \rightharpoonup u^\infty.$$

So, we can conclude that this cluster point solves $0 \in T(x, a, u)$ as desired and solves the problem (1) as well. The proof of uniqueness of weak cluster point is standard, see [5,19] for more details. □

Here we would like to stress that, one important contribution of this article is that we prove weak convergence of Algorithm 1 by developing more self-contained and less convoluted techniques. In essence, our proof is not a simple generalization since tendency of $\|w^k - w^*\|$ to zero is required in the proof of [9]. It is well known [8] that this tendency to zero may fail to hold for the proximal point algorithm (and the resulting splitting methods) in infinite-dimensional Hilbert spaces.

## 5. Special cases

In this section, we discuss several important cases of Algorithm 1.

### 5.1. Case 1

In the $n = 1$ case, the system of monotone inclusions (1) becomes the following monotone inclusion

$$0 \in \bar{A}(x) + A(x) + Q^* B(Qx - q). \tag{33}$$

Assume that $\emptyset \neq \mathrm{dom}\bar{A} \subseteq \mathrm{dom}A$, $\emptyset \neq \mathrm{dom}B$. In this case, Algorithm 1 reduces to

**Algorithm 2.**

Step 0. Choose $x^0 \in \mathcal{H}$, $a^0 \in A(x^0)$, $y^0 \in \mathcal{G}$, $u^0 \in \mathcal{G}$. Choose $\alpha > 0$, $\hat{\alpha} > 0$ and $\theta \in (0, 2)$. Set $k := 0$.
Step 1. Choose $\beta$ satisfying

$$\beta > \|Q\|^2/(4\alpha) + 1/(4\hat{\alpha}).$$

For $x^k \in \mathcal{H}$, $a^k \in A(x^k)$, $y^k \in \mathcal{G}$, $u^k \in \mathcal{G}$. Compute

$$\bar{u}^k = u^k - (y^k - Qx^k + q)/\beta,$$
$$(\alpha I + \bar{A})(\bar{x}^k) \ni \alpha x^k - a^k - Q^* \bar{u}^k,$$
$$(\hat{\alpha} I + B)(\bar{y}^k) \ni \hat{\alpha} y^k + \bar{u}^k.$$

If some stopping criterion is met, then stop. Otherwise, compute

$$\phi_k := \alpha\|x^k - \bar{x}^k\|^2 + \langle \bar{y}^k - Q\bar{x}^k + q,\ u^k - \bar{u}^k\rangle,$$
$$\varphi_k := \|x^k - \bar{x}^k\|^2 + \|\bar{y}^k - Q\bar{x}^k + q\|^2,$$
$$\gamma_k := \theta\phi_k/\varphi_k.$$

Step 2. Compute in order

$$(\alpha I + A)(x^{k+1}) \ni \alpha x^k + a^k - \gamma_k(x^k - \bar{x}^k),$$
$$\hat{\alpha} y^{k+1} = \hat{\alpha} y^k - \gamma_k(y^k - \bar{y}^k),$$
$$u^{k+1} = u^k - \gamma_k(\bar{y}^k - Q\bar{x}^k + q),$$
$$a^{k+1} = \alpha(x^k - x^{k+1}) + a^k - \gamma_k(x^k - \bar{x}^k).$$

Set $k := k + 1$.

### 5.2. Case 2

If we further assume that $B$, $Q$, $q$ vanish, then (33) reduces to

$$0 \in \bar{A}(x) + A(x). \tag{34}$$

Thus, Algorithm 2 becomes

**Algorithm 3.** A special case of Algorithm 2, i.e., Douglas–Rachford splitting method in [5]

Step 0. Choose $x^0 \in \mathcal{H}$, $\alpha > 0$. Choose $\theta \in (0, 2)$. Set $k := 0$.
Step 1. For $x^k \in \mathcal{H}$, $a^k \in A(x^k)$. Compute

$$(\alpha I + \bar{A})(\bar{x}^k) \ni \alpha x^k - a^k.$$

Step 2. Compute

$$(\alpha I + A)(x^{k+1}) \ni \alpha x^k + a^k - \theta\alpha(x^k - \bar{x}^k).$$

Set $k := k + 1$.

Obviously, if $\bar{A}$ vanishes and $\theta = 1$, then Algorithm 3 reduces to

$$(\alpha I + A)(x^{k+1}) \ni \alpha x^k.$$

This is nothing but the aforementioned proximal point algorithm.

### 5.3. Case 3

When $B$ is the sub-differential of the indicator function of the singleton set $\{0\}$. In this case, the system of monotone inclusions (1) becomes

$$0 \in \bar{A}_i(x_i) + A_i(x_i) + Q_i^* u, \ i = 1, \dots, n, \tag{35}$$
$$0 = \textstyle\sum_{i=1}^n Q_i x_i - q. \tag{36}$$

Meanwhile, Assumption 1 indicates that it has at least one solution. Thus, Algorithm 1 correspondingly reduces to the following Algorithm 4, which is particularly useful in practice.

**Algorithm 4.** A special case of Algorithm 1, used for solving (35)–(36)

Step 0. Choose $x^0 \in \mathcal{H}$, $\alpha > 0$. Choose $\theta \in (0, 2)$. Set $k := 0$.
Step 1. Choose $\beta$ satisfying

$$\beta > \textstyle\sum_{i=1}^n \|Q_i\|^2/(4\alpha_i).$$

For $x_i^k \in \mathcal{H}_i$, $a_i^k \in A_i(x^k)$, $i = 1, \dots, n$, $u^k \in \mathcal{G}$. Compute in order

$$\begin{aligned} &\bar{u}^k = u^k - (-\textstyle\sum_{i=1}^n Q_i x_i^k + q)/\beta, \\ &(\alpha_i I + \bar{A}_i)(\bar{x}_i^k) \ni \alpha_i x_i^k - a_i^k - Q_i^* \bar{u}^k. \end{aligned} \tag{37}$$

If some stopping criterion is met, stop. Otherwise, compute

$$\gamma_k := \theta \frac{\sum_{i=1}^n \alpha_i \|x_i^k - \bar{x}_i^k\|^2 + \langle -\sum_{i=1}^n Q_i \bar{x}_i^k + q, \ u^k - \bar{u}^k \rangle}{\sum_{i=1}^n \|x_i^k - \bar{x}_i^k\|^2 + \| - \sum_{i=1}^n Q_i \bar{x}_i^k + q\|^2}.$$

Step 2. Compute in order

$$\begin{aligned} &(\alpha_i I + A_i)(x_i^{k+1}) \ni \alpha_i x_i^k + a_i^k - \gamma_k (x_i^k - \bar{x}_i^k), \quad i = 1, \dots, n, \\ &u^{k+1} = u^k - \gamma_k (-\textstyle\sum_{i=1}^n Q_i \bar{x}_i^k + q), \\ &a_i^{k+1} = \alpha_i (x_i^k - x_i^{k+1}) + a_i^k - \gamma_k (x_i^k - \bar{x}_i^k), \quad i = 1, \dots, n. \end{aligned}$$

Set $k := k + 1$.

In practical implementations, we usually calculate

$$\begin{aligned} &\Delta := \textstyle\sum_{i=1}^n \|Q_i\|_1 \|Q_i\|_\infty/(4\alpha_i), \\ &\Delta' := \min\{\|Q_1\|_1 \|Q_1\|_\infty/(4\alpha_1), \dots, \|Q_n\|_1 \|Q_n\|_\infty/(4\alpha_n)\} \end{aligned}$$

and then by our numerical experience choose

$$\beta = \kappa \Delta + 10^{-9} \Delta', \quad \kappa \ge 1 \tag{38}$$

to guarantee the method's efficiency since it eventually satisfies the condition (13).

### 5.4. Case 4

Notice that a special case of (35)–(36) is optimality conditions of the following convex minimization problem

$$\begin{aligned} &\text{minimize} \ \textstyle\sum_{i=1}^n f_i(x_i), \\ &\text{subject to} \ \textstyle\sum_{i=1}^n Q_i x_i = q, \ x_i \in \mathcal{X}_i, \ i = 1, \dots, n, \end{aligned} \tag{39}$$

where, for $i = 1, \ldots, n$, $f_i$ is a closed, proper and convex function from $\mathcal{R}^{n_i}$ to $\mathcal{R}$, $Q_i$ is an $m$-by-$n_i$ matrix and $Q_i^T$ is its transpose, $q \in \mathcal{R}^m$, and $\mathcal{X}_i$ is a nonempty closed convex set in $\mathcal{R}^{n_i}$.

In this case, once we set

$$\bar{A}_i := \partial \delta_{\mathcal{X}_i}, \quad A_i := \partial f_i$$

and we further follow [9] to require the involved parameters to satisfy

$$\beta = \tilde{\alpha}, \quad \alpha_i = \tilde{\alpha}, \quad i = 1, \ldots, n, \quad \tilde{\alpha} > \sqrt{n} \max\{\|Q_i\| : i = 1, \ldots, n\}, \tag{40}$$

Algorithm 4 reduces to

**Algorithm 5.** A special case of Algorithm 4

Step 0. Choose $x^0 \in \mathcal{H}$, $\tilde{\alpha} > 0$. Choose $\theta \in (0, 2)$. Set $k := 0$.
Step 1. For $x_i^k \in \mathcal{H}_i$, $a_i^k \in \partial f_i(x^k)$, $i = 1, \ldots, n$, $u^k \in \mathcal{G}$. Compute in order

$$\bar{u}^k = u^k - (-\textstyle\sum_{i=1}^n Q_i x_i^k + q)/\tilde{\alpha},$$
$$\bar{x}_i^k = P_{\mathcal{X}_i}(\tilde{\alpha} x_i^k - a_i^k - Q_i^* \bar{u}^k).$$

If some stopping criterion is met, stop. Otherwise, compute

$$\gamma_k := \theta \frac{\sum_{i=1}^n \tilde{\alpha} \|x_i^k - \bar{x}_i^k\|^2 + \langle -\sum_{i=1}^n Q_i \bar{x}_i^k + q, \ u^k - \bar{u}^k \rangle}{\sum_{i=1}^n \|x_i^k - \bar{x}_i^k\|^2 + \| - \sum_{i=1}^n Q_i \bar{x}_i^k + q\|^2}.$$

Step 2. Compute in order

$$(\tilde{\alpha} I + \partial f_i)(x_i^{k+1}) \ni \tilde{\alpha} x_i^k + a_i^k - \gamma_k (x_i^k - \bar{x}_i^k), \quad i = 1, \ldots, n,$$
$$u^{k+1} = u^k - \gamma_k (-\textstyle\sum_{i=1}^n Q_i \bar{x}_i^k + q),$$
$$a_i^{k+1} = \tilde{\alpha}(x_i^k - x_i^{k+1}) + a_i^k - \gamma_k (x_i^k - \bar{x}_i^k), \quad i = 1, \ldots, n.$$

Set $k := k + 1$.

Algorithm 5 is actually an equivalent version of the algorithm proposed by He and Han [9]. In practical implementations, we may choose

$$\beta = \alpha_i = \tilde{\alpha} = 1.0001 \max_{i=1,\ldots,n} \left\{ \sqrt{n \|Q_i\|_1 \|Q_i\|_\infty} \right\}, \quad i = 1, \ldots, n. \tag{41}$$

## 6. Relations to other splitting methods

In this section, we compare our suggested splitting methods with other ones, which are well suited for solving general monotone inclusion (1). For simplicity, we only consider Euclidean spaces.

Consider the following monotone inclusion

$$0 \in Ax + 0.5 L_1^T B_1(L_1 x - r_1) + 0.5 L_2^T B_2(L_2 x - r_2) \quad \Leftrightarrow$$
$$0 \in 2Ax + L_1^T B_1(L_1 x - r_1) + L_2^T B_2(L_2 x - r_2),$$

where $A : \mathcal{R}^n \rightrightarrows \mathcal{R}^n$, $B_1 : \mathcal{R}^p \rightrightarrows \mathcal{R}^p$ and $B_2 : \mathcal{R}^q \rightrightarrows \mathcal{R}^q$ are maximal monotone, $L_1 : \mathcal{R}^n \to \mathcal{R}^p$ and $L_2 : \mathcal{R}^n \to \mathcal{R}^q$ are linear, and $r_1 \in \mathcal{R}^p$, $r_2 \in \mathcal{R}^q$.

A splitting method for this monotone inclusion, due to Vũ [21] and Condat [1], can be stated as follows.

**Algorithm 6.**

Step 0. Choose $x^0 \in \mathcal{R}^n$, $v_1^0 \in \mathcal{R}^p$, $v_2^0 \in \mathcal{R}^q$. Choose positive numbers $\tau, \sigma_1, \sigma_2$ such that

$$\sigma_1 \|L_1\|^2 + \sigma_2 \|L_2\|^2 < 2\tau^{-1}.$$

Set $k := 0$.

Step 1. Compute

$$p^k = (I + \tau A)^{-1}(x^k - 0.5\tau(L_1^T v_1^k + L_2^T v_2^k)),$$
$$y^k = 2p^k - x^k.$$

Choose $\lambda_k \in [0.01, 1.99]$ and compute

$$x^{k+1} = x^k + \lambda_k(p^k - x^k),$$
$$v_1^{k+1} = v_1^k + \lambda_k(I + \sigma_1 B_1^{-1})^{-1}(v_1^k + \sigma_1(L_1 y^k - r_1)) - \lambda_k v_1^k,$$
$$v_2^{k+1} = v_2^k + \lambda_k(I + \sigma_2 B_2^{-1})^{-1}(v_2^k + \sigma_2(L_2 y^k - r_2)) - \lambda_k v_2^k.$$

Set $k := k + 1$.

Note that the following Moreau identity

$$(I + \sigma B^{-1})^{-1}(u) \equiv u - \sigma(I + \frac{1}{\sigma}B)^{-1}(\frac{u}{\sigma}), \quad \sigma > 0.$$

is useful.

Consider the following monotone inclusion

$$0 \in G_1^T A_1 G_1 z + G_2^T A_2 G_2 z + A_3 z,$$

where $A_1 : \mathcal{R}^p \rightrightarrows \mathcal{R}^p$, $A_2 : \mathcal{R}^q \rightrightarrows \mathcal{R}^q$ and $A_3 : \mathcal{R}^n \rightrightarrows \mathcal{R}^n$ are maximal monotone, $G_1 : \mathcal{R}^n \to \mathcal{R}^p$ and $G_2 : \mathcal{R}^n \to \mathcal{R}^q$ are linear. Under suitable assumptions, it becomes

$$0 = G_1^T w_1 + G_2^T w_2 + w_3,$$
$$w_1 \in A_1 G_1 z,\ w_2 \in A_2 G_2 z,\ w_3 \in A_3 z.$$

A splitting method of [11, Algorithm 1] for this monotone inclusion, due to Johnstone and Eckstein, reads

**Algorithm 7.**

Step 0. Choose $z^1 \in \mathcal{R}^n,\ w^1 = (w_1^1, w_2^1, w_3^1)^T,\ w_1^1 \in \mathcal{R}^p,\ w_2^1 \in \mathcal{R}^q,\ w_3^1 \in \mathcal{R}^n$. Choose $x_1^0 \in \mathcal{R}^p,\ x_2^0 \in \mathcal{R}^q,\ x_3^0 \in \mathcal{R}^n$. Choose $0 < \alpha_i \le 1,\ \rho_i > 0,\ i = 1, 2, 3$ and $\gamma > 0$. Set $k := 1$.

Step 1. For $i = 1, 2, 3$, compute

$$t_i^k = (1 - \alpha_i)x_i^{k-1} + \alpha_i G_i z^k + \rho_i w_i^k,$$
$$x_i^k = (I + \rho_i A_i)^{-1}(t_i^k), \quad y_i^k = \rho_i^{-1}(t_i^k - x_i^k).$$

Calculate

$$u_1^k = x_1^k - G_1 x_3^k, \quad u_2^k = x_2^k - G_2 x_3^k, \quad v^k = G_1^T y_1^k + G_2^T y_2^k + y_3^k.$$

If $\pi_k := \|u^k\|^2 + \gamma^{-1}\|v^k\|^2 > 0$, then compute

$$\varphi(z^k, w^k) = \langle z^k, v^k\rangle + \langle w_1^k, u_1^k\rangle + \langle w_2^k, u_2^k\rangle - [\langle x_1^k, y_1^k\rangle + \langle x_2^k, y_2^k\rangle + \langle x_3^k, y_3^k\rangle].$$

Let $\tau_k = \frac{1}{\pi_k}\max\{0, \varphi(z^k, w^k)\}$, and go to Step 2. Otherwise, stop.

Step 2. Compute

$$z^{k+1} = z^k - \gamma^{-1}\tau_k v^k, \quad w_1^{k+1} = w_1^k - \tau_k u_1^k,$$
$$w_2^{k+1} = w_2^k - \tau_k u_2^k, \quad w_3^{k+1} = -G_1^T w_1^{k+1} - G_2^T w_2^{k+1}.$$

Set $k := k + 1$.

Be aware that Algorithms Algorithm 2, Algorithm 6, Algorithm 7 share a common feature: At each iteration, the resolvent of each operator has to be computed once.

## 7. Rudimentary experiments

In this section, we confirmed practical usefulness of Algorithm 1, together with its special cases. 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 R2020a(9.8.0) with 64-bit (win64) on a desktop computer with an Intel(R) Core(TM) i5-7400 CPU 3.00 GHz and 8.00 GB of RAM. The operating system is Windows 10.

VC splitting: The splitting method due to Vũ [21] and Condat [1], i.e., Algorithm 6.

He-Han: The aforementioned equivalent version of the algorithm proposed by He and Han [9], i.e., Algorithm 5.

Extended Splitting: An extended splitting method of [4, Algorithm 3.1], which is designed for solving (1) with $n = 1$ but $\bar{A}$ being further assumed to be inverse strongly monotone.

JE splitting: The splitting method of [11, Algorithm 1] due to Johnstone and Eckstein, i.e., Algorithm 7.

Our first test problem is to solve the following convex minimization with linear constraints

$$\text{minimize} \quad \textstyle\sum_{i=1}^{n} \vartheta|x_i| + \frac{1}{2}x_i^2, \quad \text{subject to} \quad \textstyle\sum_{i=1}^{n} Q_i x_i = q,$$

where $\vartheta > 0$, $Q_i \in \mathcal{R}^m$, $i = 1, \ldots, n$ are randomly-generated sparse vectors, with each entry being in the interval $(-1, 1)$. We set $q = \sum_{i=1}^{n} Q_i \sum_{i=1}^{n}$ so as to guarantee that this problem has at least one feasible point $(1, \ldots, 1)^T \in \mathcal{R}^n$.

This test problem can be rewritten as

$$\text{minimize} \quad \bar{f}(x) + f(x), \quad \text{subject to} \quad Qx = q,$$

where $Q := [Q_1, \ldots, Q_n]$ and

$$\bar{f}(x) := \textstyle\sum_{i=1}^{n}, \quad f(x) := \frac{1}{2}\|x\|^2.$$

Obviously, this feasible point above satisfies (10).

The resulting optimality conditions correspond to (35) with

$$\bar{A}_i(\cdot) := \vartheta\, \partial|\cdot|, \quad A_i(x_i) := x_i, \quad B = \partial\delta_{\{0\}}.$$

In practical implementations, we chose $n = 500$, $m = 10$ and $\vartheta = 100$ (be aware that $\vartheta$ shall be large so as to emphasize the role of $\sum_{i=1}^{n} |x_i|$ in the objective function). Furthermore, for Algorithm 2, we set

$$\tilde{x}_i^k := x_i^k - (a_i^k + Q_i^* \bar{u}^k)/\alpha_i,$$

and got

$$(\alpha_i I + \bar{A}_i)(\bar{x}_i^k) \ni \alpha_i \tilde{x}_i^k \quad \Rightarrow \quad \bar{x}_i^k = (I + \alpha_i^{-1}\bar{A}_i)^{-1}(\tilde{x}_i^k).$$

Thus, we further got

$$\bar{x}_i^k = (I + \alpha_i^{-1}\vartheta\, \partial|\cdot|)^{-1}(\tilde{x}_i^k) = \operatorname{sgn}(\tilde{x}_i^k) \max\left\{|\tilde{x}_i^k| - \alpha_i^{-1}\vartheta, 0\right\},$$

where the term on the righthand side is the so-called soft shrinkage function.

For Algorithm 4, we chose $\theta = 1.9$, and $\alpha_i = 1$ for $i = 1, \ldots, 500$. The primal and dual starting points were chosen as

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

To get $\beta$, we adopted (38) with $\kappa = 1$.

For VC splitting, we chose

$$2A = \partial f,\ B_1 = I,\ B_2 = \partial\delta_{\{0\}},\ L_1 = I,\ r_1 = zeros(n, 1),\ L_2 = Q,\ r_2 = q,$$

where $f(x) := \sum_{i=1}^{n} \vartheta|x_i| \sum_{i=1}^{n}$, and we chose

$$\tau = 0.02, \quad \lambda_k \equiv 1.2, \quad \sigma_1 = 0.1, \quad \sigma_2 = 0.1,$$
$$x^0 = ones(n, 1), \quad v_1^0 = zeros(n, 1), \quad v_2^0 = zeros(m, 1).$$

For He-Han, we made use of (41) and chose $\theta = 1.9$ for better numerical performance. Strictly speaking, this test problem is not a special case of (39). Yet, the corresponding method refers to Algorithm 4, but with (41).

For JE splitting, we chose

$$A_1 = \partial\delta_{\{q\}}, \quad A_2 = \partial f, \quad A_3 = I, \quad G_1 = Q, \quad G_2 = I,$$

where $f(x) := \sum_{i=1}^{n} \vartheta |x_i| \sum_{i=1}^{n}$ , and we chose

$$\begin{aligned}
&z^1 = ones(n, 1), \quad w_1^1 = zeros(m, 1), \quad w_2^1 = \vartheta * z^1, \\
&w_3^1 = -G_1^T * w_1^1 - G_2^T * w_2^1, \\
&x_1^0 = zeros(m, 1), \quad x_2^0 = zeros(n, 1), \quad x_3^0 = zeros(n, 1), \\
&\alpha = [0.9, 0.9, 0.9], \quad \rho = [1, 1, 1], \quad \gamma = 10,
\end{aligned}$$

To compare these algorithms, we mimic the style of [11] to introduce

$$f_k := \frac{\sum_{i=1}^{n} f_i(x_i^k) - f^*}{f^*} + \|Qx^k - q\|,$$

where $f^*$ is the optimal value of the objective function. To estimate it, we used the best feasible value returned by Algorithm 4 after 900 iterations. The corresponding numerical results were reported in Figure 2, where the $y$-axis was labeled as $\log 10(f_k)$.

Our second test problem is the following linear program:

$$\begin{aligned}
&\text{minimize} \quad \delta_{\mathcal{R}_+^3}(x_1) + c_1^T x_1 + \delta_{\mathcal{R}_+^2}(x_2) + c_2^T x_2, \\
&\text{subject to} \quad Q_1 x_1 + Q_2 x_2 - q = 0,
\end{aligned}$$

where

$$c_1 = (-5, -2, -3)^T, \quad c_2 = (1, -1)^T$$

and

$$Q_1 = \begin{pmatrix} 1 & 2 & 2 \\ 3 & 4 & 1 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad q = \begin{pmatrix} 8 \\ 7 \end{pmatrix}.$$

Notice that $x^* = (1.2, 0, 3.4, 0, 0)^T$ is the exact solution.

This linear program can be rewritten as

$$\text{minimize} \quad \delta_{\mathcal{R}_+^5}(x) + c^T x, \quad \text{subject to} \quad Qx - q = 0,$$

where $c := (-5, -2, -3, 1, -1)^T$, $Q = [Q_1, Q_2]$. Obviously, a feasible point $(1, 0.1, 3, 0.8, 0.6)^T$ satisfies (10). Thus, the resulting optimality conditions easily follows.

By taking

$$\bar{A}_1 = \partial\delta_{\mathcal{R}_+^3}, \quad A_1(x_1) = c_1, \ \bar{A}_2 = \partial\delta_{\mathcal{R}_+^2}, \ A_2(x_2) = c_2,$$

we applied Algorithm 4 to solving it. We chose

$$x_1^0 = (0, 0, 0)^T, \ \ x_2^0 = (0, 0)^T, \ \ u^0 = (0, 0)^T$$

as the starting points and chose $\theta = 1.4, \alpha_1 = 1, \alpha_2 = 2.5$. To get $\beta$, we adopted (38) with $\kappa = 1$.

For VC splitting, we chose

$$\begin{aligned}
&2Ax = c, \quad B_1 = \partial\delta_{\mathcal{R}_+^5}, \quad B_2 = \partial\delta_{\{0\}}, \\
&L_1 = I, \quad r_1 = zeros(5, 1), \quad L_2 = Q, \quad r_2 = q,
\end{aligned}$$

and we chose

$$\begin{aligned}
&\tau = 0.04, \quad \lambda_k \equiv 1.2, \quad \sigma_1 = 1, \quad \sigma_2 = 1, \\
&x^0 = zeros(5, 1), \quad v_1^0 = zeros(5, 1), \quad v_2^0 = zeros(2, 1).
\end{aligned}$$

For He-Han, we made use of (41) and chose $\theta = 1.4$ for better numerical performance.

For JE splitting, we chose

$$A_1 z = c, \quad A_2 = \partial\delta_{\{q\}}, \quad A_3 = \partial\delta_{\mathcal{R}^5_+}, \quad G_1 = I, \quad G_2 = Q,$$

and we chose

$$\begin{aligned}
&z^1 = zeros(5, 1), \quad w_1^1 = c, \quad w_2^1 = zeros(2, 1), \quad w_3^1 = zeros(5, 1),\\
&x_1^0 = zeros(5, 1), \quad x_2^0 = zeros(2, 1), \quad x_3^0 = zeros(5, 1),\\
&\alpha = [0.9, 0.9, 0.9], \quad \rho = [1, 1, 1], \quad \gamma = 10.
\end{aligned}$$

The corresponding numerical results were reported in Figure 2, where the $y$-axis was labeled as $\log10(\|x^k - x^*\|)$.

Our third test problem is from [4], which is to find an $x \in \mathcal{R}^m$ such that

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

where

$$D = \text{tridiag}\,(-1 - h, 4 + 2h, -1),$$

where $h := 1/(m + 1)$, and

$$Q = [eye(m); (-1/m) * ones(1, m)]; \quad q = [zeros(m, 1); -1/m]$$

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. Thus, the problem's unique solution $x^* = e_1$. We chose

$$\bar{A}x = 0.5(D + D^T)x - d, \quad A = 0.5(D - D^T), \quad B = \partial\delta_{\mathcal{C}}$$

to match the problem (1), as done in [4]. In addition, we chose $m = 1000$ and

$$x^0 = zeros(m, 1), \quad u^0 = zeros(m + 1, 1).$$

In practical implementations, for the parameters in Algorithm 2, we set $\alpha_1 = 1$, $\alpha_2 = 1$ and we first chose

$$\theta \in \{1.0, 1.2, 1.4, 1.6, 1.8, 1.99\}, \quad \kappa \in \{1, 2\}$$

for 12 times of trials and finally chose $\theta = 1.2$ and $\kappa = 2$ in (21) for better numerical performance.

Be aware that, for Step 2 in Algorithm 2, we directly used Matlab solver (cf. (22)) to solve

$$\big(\alpha I + 0.5(D - D^T)\big)\, x = \alpha x^k + a^k - \gamma_k(x^k - \bar{x}^k)$$

to get $x^{k+1}$.

For VC splitting, we chose

$$\begin{aligned}
&2Ax = 0.5(D + D^T)x - d, \quad B_1 = 0.5(D - D^T), \quad B_2 = \partial\delta_{\mathcal{C}},\\
&L_1 = I, \quad r_1 = zeros(m, 1), \quad L_2 = Q, \quad r_2 = q,
\end{aligned}$$

and we chose

$$\begin{aligned}
&\tau = 0.5, \quad \lambda_k \equiv 1.4, \quad \sigma_1 = 1.6, \quad \sigma_2 = 1.6,\\
&x^0 = zeros(m, 1), \quad v_1^0 = zeros(m, 1), \quad v_2^0 = zeros(m + 1, 1).
\end{aligned}$$

For Extended Splitting, we implemented it in the same way as [4].

For JE splitting, we chose

$$\begin{aligned}
&A_1 = \partial\delta_{\{q\}}, \quad A_2 z = 0.5(D + D^T)z - d, \quad A_3 = 0.5(D - D^T),\\
&G_1 = Q, \quad G_2 = I,
\end{aligned}$$

and we chose

$$\begin{aligned}
&z^1 = zeros(m, 1), \quad w_1^1 = zeros(m + 1, 1),\\
&w_2^1 = 0.5(D + D^T)z^1 - d, \quad w_3^1 = -G_1^T * w_1^1 - G_2^T * w_2^1,\\
&x_1^0 = zeros(m + 1, 1), \quad x_2^0 = zeros(m, 1), \quad x_3^0 = zeros(m, 1),
\end{aligned}$$

$$\alpha = [0.9, 0.9, 0.9], \quad \rho = [1, 1, 1], \quad \gamma = 10.$$

The corresponding numerical results were reported in Fig. 3, where the $y$-axis was labeled as $\log 10(\|x^k - x^*\|)$.

Our fourth test problem is to solve the following convex minimization with linear constraints

$$\text{minimize} \quad \delta_{\mathcal{X}_i}(x_i) + \textstyle\sum_{i=1}^n f_i(x_i), \quad \text{subject to} \quad \textstyle\sum_{i=1}^n i\, x_i = 1,$$

where

$$\mathcal{X}_i := \{x_i : 1 \geq x_i \geq 1/(n(n+1))\}, \;\; i = 1, \dots, n, \quad f_i(x_i) := x_i - t \ln x_i.$$

Its equivalent form is

$$\text{minimize} \quad \delta_{\mathcal{X}}(x) + f(x), \quad \text{subject to} \quad Qx = 1$$

where $Q = (1, \dots, n)$ and

$$\mathcal{X} := \{x : 1 \geq x_i \geq 1/(n(n+1)),\; i = 1, \dots, n\}, \quad f(x) := \textstyle\sum_{i=1}^n f_i(x_i).$$

Obviously, a feasible point $x^*$, with $x_i^* = 1/(ni), i = 1, \dots, n$, satisfies (10). Notice that this point is also the unique solution.

For this test problem, its optimality conditions correspond to (35)–(36) with

$$\bar{A}_i := \partial \delta_{\mathcal{X}_i}, \;\; A_i := \nabla f_i, \;\; Q_i = i, \;\; q = 1.$$

In practical implementations, set $n = 10$, $t = 1/(2n)$, $\theta = 1$ and

$$x^0 = ones(n, 1), \;\; u^0 = 0.$$

For Algorithm 4, we took each $\alpha_i$ to be some value equal to or close to local Lipschitz constant of $\nabla f_i$, where

$$\nabla f_i(x_i) = 1 - t/x_i.$$

Since $\nabla^2 f_i(x_i) = t/x_i^2$ (its absolute value corresponds to local Lipschitz "constant") is no longer constant, in practical implementations, we had to allow $\alpha_i$ to vary from iteration to iteration. At $k$th iteration, we took

$$\alpha_i \leftarrow \alpha_i^k = t/(x_i^k)^2$$

as an approximation of local Lipschitz constant of $\nabla f_i$ around $x_i^k$. To get $\beta$, we adopted (38) with $\kappa = 1$. There is no need to worry about convergence because we can keep $\alpha_i$ unchanged after the first finite iterations.

Be aware that, for Step 2 in Algorithm 4, in order to solve

$$(\alpha_i I + \nabla f_i)(x_i) = \alpha_i x_i^k + a_i^k - \gamma_k(x_i^k - \bar{x}_i^k), \;\; i = 1, \dots, n$$

to get $x_i^{k+1}$, we consider

$$\alpha_i x_i + 1 - \frac{t}{x_i} = \alpha_i x_i^k + a_i^k - \gamma_k(x_i^k - \bar{x}_i^k), \;\; i = 1, \dots, n.$$

Since it can be converted into quadratic equation with respect to the variable $x_i$ and $\ln x_i$ implies that $x_i$ must be positive, we took its unique positive root as $x_i^{k+1}$, i.e.,

$$x_i^{k+1} = \frac{w_i^k - 1 + \sqrt{(w_i^k - 1)^2 + 4\alpha_i t}}{2\alpha_i}, \quad i = 1, \dots, n,$$

where $w_i^k := \alpha_i x_i^k + a_i^k - \gamma_k(x_i^k - \bar{x}_i^k)$.

For VC splitting, we chose

$$2A = \nabla f, \quad B_1 = \partial \delta_{\mathcal{X}}, \quad B_2 = \partial \delta_{\{0\}},$$
$$L_1 = I, \quad r_1 = zeros(n, 1), \quad L_2 = (1, \dots, n), \quad r_2 = 1,$$

and we chose

$$\tau = 0.02, \quad \lambda_k \equiv 1.0, \quad \sigma_1 = 0.1, \quad \sigma_2 = 0.1,$$
$$x^0 = zeros(n, 1), \quad v_1^0 = zeros(n, 1), \quad v_2^0 = 0.$$

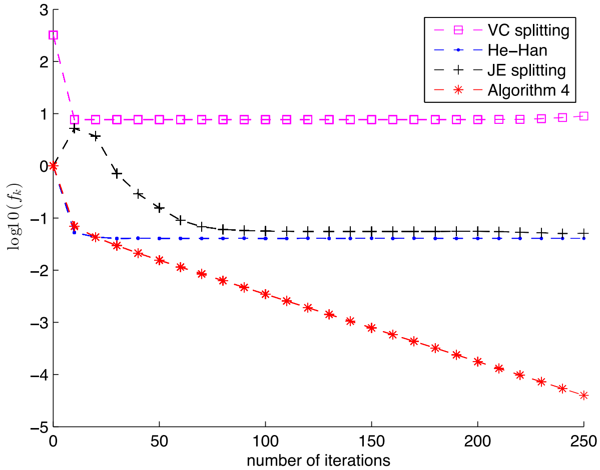


**Fig. 1.** Numerical results on the first test problem.

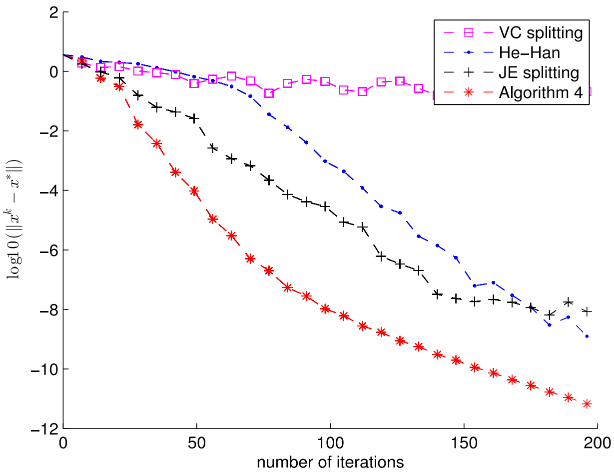


**Fig. 2.** Numerical results on the second test problem.

For He-Han, its parameters were chosen by (41) and $\theta = 1$ for better numerical performance.

For JE splitting, we chose

$$A_1 = \nabla f, \quad A_2 = \partial\delta_{\{1\}}, \quad A_3 = \partial\delta_{\mathcal{X}}, \quad G_1 = I, \quad G_2 = (1, \ldots, n),$$

and we chose

$$\begin{aligned} &z^1 = ones(n, 1), \quad w_1^1 = (1-t)z^1, \quad w_2^1 = 0, \\ &w_3^1 = -G_1^T * w_1^1 - G_2^T * w_2^1, \quad x_1^0 = zeros(n, 1), \quad x_2^0 = 0, \\ &x_3^0 = zeros(n, 1), \quad \alpha = [0.9, 0.9, 0.9], \quad \rho = [1, 1, 1], \quad \gamma = 10. \end{aligned}$$

The corresponding numerical results were reported in Fig. 4, where the $y$-axis was labeled as $\log 10(\|x^k - x^*\|)$.

From Figs. 1–4, we can see that all these four test problems were stably and efficiently solved by either Algorithm 2 or Algorithm 4. Furthermore, either always outperformed other state-of-the-art algorithms in achieving higher accuracy for the first, second and fourth. In particular, for the *hardest* fourth problem, it was significantly better than other three ones.

At the end of this section, we would like to stress two points. One is that Algorithm 1, together with its special case, has a nice feature: take full advantage of separable structures so that each subproblem is easier to solve. In

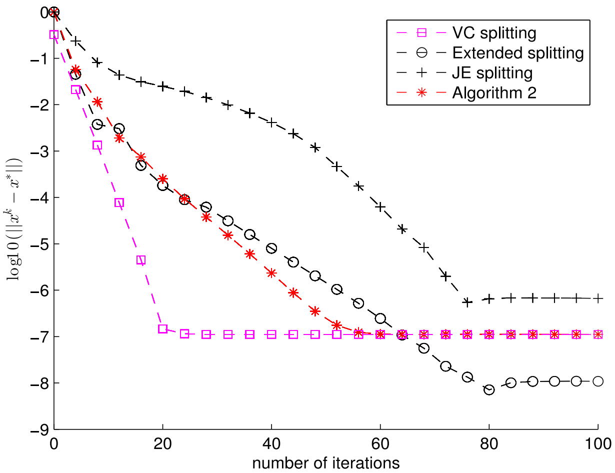


**Fig. 3.** Numerical results on the third test problem.

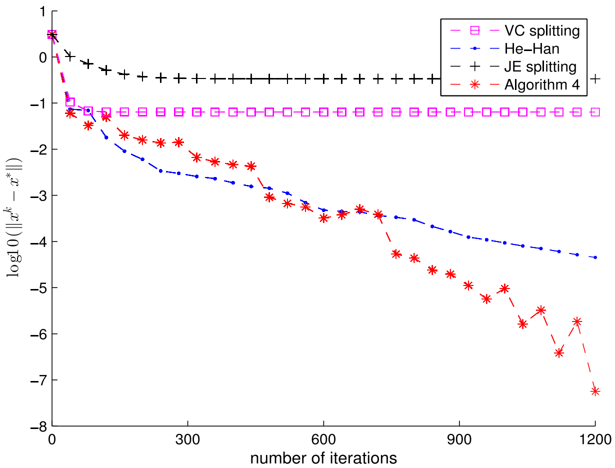


**Fig. 4.** Numerical results on the fourth test problem.

contrast, the classical augmented Lagrangian method does not have such a nice feature; see [9] for more details. The other is that Algorithm 1 is originally designed for solving more general monotone inclusions (1) than [2–6,10]. So, we have no intension to make numerical comparisons among these algorithms.

## 8. Conclusions

In this article, we have considered the problem of finding a zero of separable systems of monotone inclusions in real Hilbert spaces, and we have suggested a splitting method, which can be viewed as a powerful extension of the classical Douglas–Rachford splitting. Furthermore, under the weakest possible assumptions, we have proven its weak convergence by invoking more self-contained and less convoluted techniques. Impressively, our suggested splitting method inherits an appealing feature of the classical Douglas–Rachford splitting method: the involved scaling factors can be any given positive numbers, respectively. Furthermore, we have done numerical experiments to confirm practical usefulness of this method, together with its special cases, in solving our test problems of separable structure.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The author is greatly indebted to the editor and the referee for their suggestions, which improve the quality of this article. Special thanks go to Dr. Xue Zhang, Xiao Zhu, Mengdi Sun and Yiyi Li for writing Matlab codes. This work was partially supported by the National Natural Science Foundation of China (11901368).

## References

[1] L. Condat, A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms, J. Optim. Theory Appl. 158 (2) (2013) 460–479.
[2] Y.D. Dong, Douglas-Rachford splitting method for semi-definite programming, J. Appl. Math. Comput. 51 (2016) 569–591.
[3] Y.D. Dong, New inertial factors of the Krasnosel'skiĭ-Mann iteration, Set-Valued Var Anal. 29 (1) (2021) 145–161.
[4] Y.D. Dong, Weak convergence of an extended splitting method for monotone inclusions, J. Global Optim. 79 (2021) 257–277.
[5] Y.D. Dong, A. Fischer, A family of operator splitting methods revisited, Nonlinear Anal. 72 (2010) 4307–4315.
[6] Y.D. Dong, X.H. Yu, A new splitting method for monotone inclusions of three operators, Calcolo 56 (1) (2019) 3.
[7] J. Eckstein, D.P. Bertsekas, On the douglas-rachford splitting method and the proximal algorithm for maximal monotone operators, Math. Program. 55 (1992) 293–318.
[8] O. Güler, On the convergence of the proximal point algorithm for convex minimization, SIAM J. Control Optim. 29 (1991) 403–419.
[9] H.J. He, D.R. Han, A distributed Douglas-Rachford splitting method for multi-block convex minimization problems, Adv. Comput. Math. 42 (1) (2016) 27–53.
[10] O.S. Iyiola, C.D. Enyi, Y. Shehu, Reflected three-operator splitting method for monotone inclusion problem, Optim Methods Softw. http://dx.doi.org/10.1080/10556788.2021.1924715, in press.
[11] P.R. Johnstone, J. Eckstein, Single-forward-step projective splitting: exploiting cocoercivity, Comput. Optim. Appl. 78 (2021) 125–166.
[12] P. Latafat, P. Patrinos, Asymmetric forward–backward-adjoint splitting for solving monotone inclusions involving three operators, Comput. Optim. Appl. 68 (1) (2017) 57–93.
[13] P.L. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, SIAM J. Numer. Anal. 16 (1979) 964–979.
[14] B. Martinet, Regularisation d'inéquations variationelles par approximations successives, Rev Fr D'Inform. Recherche Opér. 4 (1970) 154–158.
[15] S.Y. Matsushita, On the convergence rate of the Krasnosel'skiĭ-Mann iteration, B Austr. Math. Soc. 96 (1) (2017) 162–170.
[16] G.J. Minty, Monotone (nonlinear) operators in Hilbert space, Duke Math. J. 29 (1962) 341–346.
[17] T. Pennanen, Dualization of generalized equations of maximal monotone type, SIAM J. Optim. 10 (3) (2000) 809–835.
[18] R.T. Rockafellar, On the maximality of sums of nonlinear monotone operators, Trans. Amer. Math. Soc. 149 (1) (1970) 75–88.
[19] R.T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM J. Control Optim. 14 (1976) 877–898.
[20] D.R. Sahu, Applications of accelerated computational methods for quasi-nonexpansive operators to optimization problems, Soft Comput. 24 (2020) 17887–17911.
[21] B. Vũ, A splitting algorithm for dual monotone inclusions involving cocoercive operators, Adv. Comput. Math. 38 (3) (2013) 667–681.
[22] D. Wicht, M. Schneider, T. Böhlke, An efficient solution scheme for small-strain crystal-elasto-viscoplasticity in a dual framework, Comput. Methods Appl. Mech. Engrg. 358 (2020) 112611.