




# Distributed forward-backward methods for ring networks

Francisco J. Aragón-Artacho<sup>1</sup> · Yura Malitsky<sup>2</sup> · Matthew K. Tam<sup>3</sup> ·  
David Torregrosa-Belén<sup>1</sup> 

Received: 2 December 2021 / Accepted: 11 July 2022  
© The Author(s) 2022

## Abstract

In this work, we propose and analyse forward-backward-type algorithms for finding a zero of the sum of finitely many monotone operators, which are not based on reduction to a two operator inclusion in the product space. Each iteration of the studied algorithms requires one resolvent evaluation per set-valued operator, one forward evaluation per cocoercive operator, and two forward evaluations per monotone operator. Unlike existing methods, the structure of the proposed algorithms are suitable for distributed, decentralised implementation in ring networks without needing global summation to enforce consensus between nodes.

**Keywords** Monotone operator · Monotone inclusion · Splitting algorithm · Forward-backward algorithm · Distributed optimisation

**Mathematics Subject Classification** 47H05 · 65K10 · 90C30

---

Dedicated to the memory of Asen L. Dontchev.

---

✉ David Torregrosa-Belén  
david.torregrosa@ua.es

Francisco J. Aragón-Artacho  
francisco.aragon@ua.es

Yura Malitsky  
yurii.malitskyi@liu.se

Matthew K. Tam  
matthew.tam@unimelb.edu.au

<sup>1</sup> Department of Mathematics, University of Alicante, San Vicente del Raspeig, Alicante 03690, Spain

<sup>2</sup> Department of Mathematics, Linköping University, Linköping 581 83, Sweden

<sup>3</sup> School of Mathematics & Statistics, University of Melbourne, Parkville, VIC 3010, Australia

## 1 Introduction

In this work, we propose algorithms of forward-backward-type for solving structured monotone inclusions in a real Hilbert space  $\mathcal{H}$ . Specifically, we consider the problem

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \left( \sum_{i=1}^n A_i + \sum_{i=1}^m B_i \right)(x), \quad (1)$$

where  $A_1, \dots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$  are maximally monotone operators, and  $B_1, \dots, B_m : \mathcal{H} \rightarrow \mathcal{H}$  are either cocoercive, or monotone and Lipschitz continuous. Inclusions in the form (1) arise in a number of settings of fundamental importance in mathematical optimisation. In what follows, we describe three such examples.

**Example 1** (Composite minimisation) Consider the minimisation problem given by

$$\min_{x \in \mathcal{H}} \sum_{i=1}^n g_i(x) + \sum_{i=1}^m f_i(x), \quad (2)$$

where  $g_1, \dots, g_n : \mathcal{H} \rightarrow (-\infty, +\infty]$  are proper, lsc and convex, and  $f_1, \dots, f_m : \mathcal{H} \rightarrow (-\infty, +\infty)$  are convex and differentiable with  $L$ -Lipschitz continuous gradients. Through its first order optimality condition, (2) can be posed as (1) with

$$A_i = \partial g_i \text{ and } B_i = \nabla f_i$$

where  $\partial g_i$  denotes the *subdifferential* of  $g_i$ . Note that the operators  $B_1, \dots, B_m$  are both  $L$ -Lipschitz and  $\frac{1}{L}$ -cocoercive, due to the *Baillon–Haddad theorem* [1, Corolaire 10].

**Example 2** (Structured saddle-point problems) Consider the saddle-point problem given by

$$\min_{x \in \mathcal{H}_1} \max_{y \in \mathcal{H}_2} \sum_{i=1}^n h_i(x) + \sum_{i=1}^m \Phi_i(x, y) - \sum_{i=1}^n g_i(y), \quad (3)$$

where  $h_1, \dots, h_n : \mathcal{H}_1 \rightarrow (-\infty, +\infty]$ ,  $g_1, \dots, g_n : \mathcal{H}_2 \rightarrow (-\infty, +\infty]$  are proper, lsc and convex, and  $\Phi_1, \dots, \Phi_m : \mathcal{H}_1 \times \mathcal{H}_2 \rightarrow (-\infty, +\infty]$  are differentiable convex-concave functions with Lipschitz continuous gradient. Assuming a saddle-point exists, (3) can be posed as (1) in the space  $\mathcal{H} := \mathcal{H}_1 \times \mathcal{H}_2$  with

$$A_i(x, y) = \begin{pmatrix} \partial h_i(x) \\ \partial g_i(y) \end{pmatrix} \text{ and } B_i(x, y) = \begin{pmatrix} \nabla_x \Phi_i(x, y) \\ -\nabla_y \Phi_i(x, y) \end{pmatrix},$$

where we note that the operators  $B_1, \dots, B_m : \mathcal{H} \rightarrow \mathcal{H}$  are monotone, due to [2, Theorem 2], and  $L$ -Lipschitz continuous, but generally not cocoercive.

**Example 3** (Structured variational inequalities) Consider the variational inequality problem given by

$$\text{find } x^* \in \mathcal{H} \text{ such that } \sum_{i=1}^n g_i(x) - \sum_{i=1}^n g_i(x^*) + \sum_{i=1}^m \langle B_i(x^*), x - x^* \rangle \geq 0 \quad \forall x \in \mathcal{H}, \quad (4)$$

where  $g_1, \dots, g_n : \mathcal{H} \rightarrow (-\infty, +\infty]$  are proper, lsc and convex, and  $B_1, \dots, B_m : \mathcal{H} \rightarrow \mathcal{H}$  are monotone and  $L$ -Lipschitz. Then (4) is of the form of (1) with  $A_i = \partial g_i$ . An important special case of (4) is the constrained variational inequality problem given by

$$\text{find } x^* \in \mathcal{H} \text{ such that } \sum_{i=1}^m \langle B_i(x^*), x - x^* \rangle \geq 0 \quad \forall x \in C := \bigcap_{i=1}^n C_i,$$

where  $C_1, \dots, C_n \subseteq \mathcal{H}$  are nonempty, closed and convex sets. This formulation allows one to exploit a representation of the set  $C$  in terms of the simpler sets  $C_1, \dots, C_n$ .

## 1.1 Splitting algorithms

We focus on *splitting algorithms* for solving (1) of forward-backward-type, by which we mean those whose iteration can be expressed in terms of the *resolvents* of the set-valued operators  $A_1, \dots, A_n$  and direct evaluations of the single-valued operators  $B_1, \dots, B_m$ . It is always possible to reduce this problem to the  $m = 1$  case by combining the single-valued operators into a single operator  $F := \sum_{i=1}^m B_i$  whilst preserving the above features. However, since the resolvent of a sum is generally not related to the individual resolvents, the same cannot be said for the set-valued operators, and so it makes sense to distinguish algorithms for (1) based on the value of  $n$ .

In the case  $n = 1$ , there are many methods satisfying the above criteria. Among them, the best known are arguably the *forward-backward method* given by

$$x^{k+1} = J_{\lambda A_1}(x^k - \lambda F(x^k)),$$

which can be used when  $F$  is cocoercive, and the *forward-backward-forward method* [3] given by

$$\begin{cases} y^k = J_{\lambda A_1}(x^k - \lambda F(x^k)) \\ x^{k+1} = y^k - \lambda F(y^k) + \lambda F(x^k), \end{cases}$$

which can be used when  $F$  is monotone and Lipschitz. When  $n = 2$ , there are also many methods. For instance, if  $F$  is cocoercive, *Davis–Yin splitting* [4–6] which takes the form

$$\begin{cases} x^k = J_{\lambda A_1}(z^k) \\ z^{k+1} = z^k + J_{\lambda A_2}(2x^k - z^k - \lambda F(x^k)) - x^k \end{cases}$$

can be applied, and if  $F$  is monotone and Lipschitz, then the *backward-forward-reflected-backward methods* [7] can be used.

However, for  $n > 2$ , the situation is drastically different. Most existing methods rely on a product space reformulation, either directly or implicitly. For instance, the iteration given by

$$\begin{cases} x^k = \frac{1}{n} \sum_{i=1}^n z_i^k \\ z_i^{k+1} = z_i^k + J_{\lambda A_i}(2x^k - z_i^k - \lambda B_i(x^k)) - x^k \quad \forall i \in \llbracket 1, n \rrbracket \end{cases} \quad (5)$$

for cocoercive  $B_1, \dots, B_n$ , where  $\llbracket 1, n \rrbracket$  denotes the integers between 1 and  $n$ , amounts to Davis–Yin splitting applied to the three operator inclusion

$$\text{find } \mathbf{x} = (x, \dots, x) \in \mathcal{H}^n \text{ such that } 0 \in (N_D + A + B)(\mathbf{x}), \quad (6)$$

where  $A := (A_1, \dots, A_n)$ ,  $B := (B_1, \dots, B_n)$  and  $N_D$  denotes the *normal cone* to the diagonal subspace  $D := \{(x_1, \dots, x_n) \in \mathcal{H}^n : x_1 = \dots = x_n\}$ . Other methods for (1) with  $n > 2$  include the *generalised forward-backward method* [8] and those from the *projective splitting family* [9, 10].

Indisputably, product space reformulations such as (6) provide a convenient tool that makes the derivation of algorithms for  $n > 2$  operators an almost mechanical procedure. It is therefore natural to consider whether this tool is the only one at our disposal. In addition to academic importance in its own right, the discovery of new algorithms that do not fall within standard categories can provide new possibilities, both in terms of mathematical techniques and potential applications. Sometimes these applications can be quite unexpected, as we demonstrate next.

## 1.2 Distributed algorithms

Advances in hardware (parallel computation) and increasing the size of datasets (decentralised storage) have made distributed algorithms one of the most prevalent trends in algorithm development. Such algorithms rely on a network of devices that perform subtasks and are able to communicate with each other. For details on the topic, the reader is referred to the book of Bertsekas & Tsitsiklis [11] as well as [12] for recent advances.

From the perspective of distributed computing, the product space formulation generally requires the computation of a global sum across all nodes in every iteration. To be more concrete, consider a distributed implementation of (5) in which node  $i$  performs the  $z_i$ -updates by using its operators,  $A_i$  and  $B_i$ . To perform the  $x$ -update, the local variables  $z_1, \dots, z_n$  must be aggregated and the result then broadcast to the entire network. There may be many reasons why this is not desirable, including default network setting, privacy or cost issues.

Another important aspect of distributed communication is parallelism and synchronisation. Returning to our example involving (5) from the previous paragraph, the product space reformulation provides a fully parallel algorithm in the sense that all nodes performing  $z$ -updates can compute their updates in parallel before sending to the central coordinator. This parallelisation comes at cost of requiring global synchronisation between nodes. Specially, the algorithm (5) cannot move from  $k$ -th to  $(k + 1)$ -th iteration until all nodes  $1, \dots, n$  have completed their computation. This can be overcome with *asynchronous algorithms*, that is, those which only require little or no global synchronisation. However, their development and mathematical analysis are significantly more delicate.

### 1.3 Our contribution

We propose and analyse algorithms of forward-backward-type for solving (1) which exploit problem structure. Note that by using the zero operator in (1) if necessary, we can always assume that  $m = n - 1$ . Applied to this problem with cocoercive operators  $B_1, \dots, B_{n-1}$ , our algorithm can be expressed as the fixed point iteration  $\mathbf{z}^{k+1} = T(\mathbf{z}^k)$  based on the operator  $T : \mathcal{H}^{n-1} \rightarrow \mathcal{H}^{n-1}$  given by

$$T(\mathbf{z}) := \mathbf{z} + \gamma \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix},$$

where  $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$  depends on  $\mathbf{z} = (z_1, \dots, z_{n-1}) \in \mathcal{H}^{n-1}$  and is given by

$$\begin{cases} x_1 = J_{\lambda A_1}(z_1), \\ x_i = J_{\lambda A_i}(z_i + x_{i-1} - z_{i-1} - \lambda B_{i-1}(x_{i-1})) \quad \forall i \in \llbracket 2, n-1 \rrbracket, \\ x_n = J_{\lambda A_n}(x_1 + x_{n-1} - z_{n-1} - \lambda B_{n-1}(x_{n-1})). \end{cases}$$

For the case where  $B_i$  are monotone and Lipschitz, the underlying operator is slightly more complicated and relies on an update similar to the one proposed in the *forward-reflected-backward* method [13].

Overall, the notable characteristics of the algorithms we propose are:

- They do not rely on existing product space reformulation: Instead, we extend the framework for backward operators, proposed in [14], which in turn is a generalisation of [15] for  $n > 3$ .
- They are decentralised and can be naturally implemented on a ring network for communication.
- The order in which variables are updated can vary significantly between executions:  $z_i^{k+1}$  can be computed before evaluation of  $z_{i+2}^k, z_{i+3}^{k-1}, \dots$

Importantly, we believe that our work is an important starting point towards a more general template that will allow for different network topologies.

The remainder of this work is structured as follows: In Sect. 2, we recall notation and preliminaries for later use. In Sect. 3, we introduce and analyse a forward-backward type algorithm for solving (1) with cocoercive operators. In Sect. 4, we introduce and analyse a modification of the algorithm from Sect. 3 which can be used when  $B_1, \dots, B_m$  are not necessarily cocoercive.

## 2 Preliminaries

Throughout this paper,  $\mathcal{H}$  denotes a real Hilbert space equipped with inner product  $\langle \cdot, \cdot \rangle$  and induced norm  $\| \cdot \|$ . A *set-valued operator* is a mapping  $A : \mathcal{H} \rightrightarrows \mathcal{H}$  that assigns to each point in  $\mathcal{H}$  a subset of  $\mathcal{H}$ , i.e.,  $A(x) \subseteq \mathcal{H}$  for all  $x \in \mathcal{H}$ . In the case when  $A$  always maps to singletons, i.e.,  $A(x) = \{u\}$  for all  $x \in \mathcal{H}$ ,  $A$  is said to be a *single-valued mapping* and is denoted by  $A : \mathcal{H} \rightarrow \mathcal{H}$ . In an abuse of notation, we may write  $A(x) = u$  when  $A(x) = \{u\}$ . The *domain*, the *graph*, the set of *fixed points* and the set of *zeros* of  $A$ , are denoted, respectively, by  $\text{dom } A$ ,  $\text{gra } A$ ,  $\text{Fix } A$  and  $\text{zer } A$ ; i.e.,

$$\begin{aligned}\text{dom } A &:= \{x \in \mathcal{H} : A(x) \neq \emptyset\}, & \text{gra } A &:= \{(x, u) \in \mathcal{H} \times \mathcal{H} : u \in A(x)\}, \\ \text{Fix } A &:= \{x \in \mathcal{H} : x \in A(x)\}, & \text{zer } A &:= \{x \in \mathcal{H} : 0 \in A(x)\}.\end{aligned}$$

The *inverse operator* of  $A$ , denoted by  $A^{-1}$ , is defined through  $x \in A^{-1}(u) \iff u \in A(x)$ . The *identity operator* is denoted by  $\text{Id}$ .

**Definition 1** An operator  $B : \mathcal{H} \rightarrow \mathcal{H}$  is said to be

- (i) *L-Lipschitz continuous* for  $L > 0$  if

$$\|B(x) - B(y)\| \leq L\|x - y\| \quad \forall x, y \in \mathcal{H};$$

- (ii)  $\frac{1}{L}$ -*cocoercive* for  $L > 0$  if

$$\langle B(x) - B(y), x - y \rangle \geq \frac{1}{L} \|B(x) - B(y)\|^2 \quad \forall x, y \in \mathcal{H}.$$

Note that, by the Cauchy–Schwarz inequality, a  $\frac{1}{L}$ -cocoercive operator is always  $L$ -Lipschitz continuous.

**Definition 2** An operator  $T : \mathcal{H} \rightarrow \mathcal{H}$  is said to be

- (i) *quasi-nonexpansive* if

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

- (ii) *nonexpansive* if it is 1-Lipschitz continuous, i.e.,

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

- (iii) *strongly quasi-nonexpansive* if there exists  $\sigma > 0$  such that

$$\|T(x) - y\|^2 + \sigma \|(\text{Id} - T)(x)\|^2 \leq \|x - y\|^2 \quad \forall x \in \mathcal{H}, \forall y \in \text{Fix } T;$$

(iv) *averaged nonexpansive* if there exists  $\alpha \in (0, 1)$  such that

$$\|T(x) - T(y)\|^2 + \frac{1 - \alpha}{\alpha} \|(\text{Id} - T)(x) - (\text{Id} - T)(y)\|^2 \leq \|x - y\|^2 \quad \forall x, y \in \mathcal{H}.$$

In particular, the following implications hold: (iv) $\Rightarrow$ (ii) $\Rightarrow$ (i) and (iv) $\Rightarrow$ (iii) $\Rightarrow$ (i).

When we wish to explicitly specify the constants involved, we refer to the operators in Definition 2(iii) and (iv), respectively, as  $\sigma$ -strongly quasi-nonexpansive and  $\alpha$ -averaged nonexpansive. Since the mapping  $\alpha \mapsto \frac{1-\alpha}{\alpha}$  is a bijection from  $(0, 1)$  to  $(0, +\infty)$ , there is a one-to-one relationship between the values of  $\sigma$  in (iii) and  $\alpha$  in (iv), with inverse relation given by  $\sigma \mapsto \frac{1}{1+\sigma}$ .

**Definition 3** A set-valued operator  $A : \mathcal{H} \rightrightarrows \mathcal{H}$  is *monotone* if

$$\langle x - y, u - v \rangle \geq 0 \quad \forall (x, u), (y, v) \in \text{gra } A.$$

Furthermore,  $A$  is said to be *maximally monotone* if there exists no monotone operator  $B : \mathcal{H} \rightrightarrows \mathcal{H}$  such that  $\text{gra } B$  properly contains  $\text{gra } A$ .

**Proposition 1** ([16, Corollary 20.28]) *Every continuous monotone operator with full domain is maximally monotone. In particular, every cocoercive operator is maximally monotone.*

The resolvent operator, whose definition is given next, is one of the main building blocks of splitting algorithms.

**Definition 4** Given an operator  $A : \mathcal{H} \rightrightarrows \mathcal{H}$ , the *resolvent* of  $A$  with parameter  $\gamma > 0$  is the operator  $J_{\gamma A} : \mathcal{H} \rightrightarrows \mathcal{H}$  defined by  $J_{\gamma A} := (\text{Id} + \gamma A)^{-1}$ .

**Proposition 2** ([17] or [16, Corollary 23.11]) *Let  $A : \mathcal{H} \rightrightarrows \mathcal{H}$  be monotone and let  $\gamma > 0$ . Then*

- (i)  $J_{\gamma A}$  is single-valued,
- (ii)  $\text{dom } J_{\gamma A} = \mathcal{H}$  if and only if  $A$  is maximally monotone.

### 3 A distributed forward-backward method

Let  $n \geq 2$  and consider the problem

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right)(x), \quad (7)$$

where  $A_1, \dots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$  are maximally monotone and  $B_1, \dots, B_{n-1} : \mathcal{H} \rightarrow \mathcal{H}$  are  $\frac{1}{L}$ -cocoercive.

For the case when  $B_1 = \dots = B_{n-1} = 0$ , Malitsky and Tam [14] proposed a splitting algorithm with  $(n-1)$ -fold lifting for finding a zero of the sum of  $n \geq 2$  maximally monotone operators; see also [18] for recent extensions. In this section, we adapt the methodology developed in [6] to obtain a splitting method of forward-backward-type for the inclusion (7) by modifying the splitting method in [14] without increasing the dimension of the ambient space.

Given  $\lambda \in (0, \frac{2}{L})$  and  $\gamma \in (0, 1 - \frac{\lambda L}{2})$  and an initial point  $\mathbf{z}^0 = (z_1^0, \dots, z_{n-1}^0) \in \mathcal{H}^{n-1}$ , our proposed algorithm for (7) generates two sequences,  $(\mathbf{z}^k) \subseteq \mathcal{H}^{n-1}$  and  $(\mathbf{x}^k) \subseteq \mathcal{H}^n$ , according to

$$\mathbf{z}^{k+1} = \mathbf{z}^k + \gamma \begin{pmatrix} x_2^k - x_1^k \\ x_3^k - x_2^k \\ \vdots \\ x_n^k - x_{n-1}^k \end{pmatrix} \quad (8a)$$

and

$$\begin{cases} x_1^k = J_{\lambda A_1}(z_1^k), \\ x_i^k = J_{\lambda A_i}(z_i^k + x_{i-1}^k - z_{i-1}^k - \lambda B_{i-1}(x_{i-1}^k)) \quad \forall i \in \llbracket 2, n-1 \rrbracket, \\ x_n^k = J_{\lambda A_n}(x_1^k + x_{n-1}^k - z_{n-1}^k - \lambda B_{n-1}(x_{n-1}^k)). \end{cases} \quad (8b)$$

The structure of (8) lends itself to a distributed decentralised implementation, similar to the one in [14, Algorithm 2]. More precisely, consider a cycle graph with  $n$  nodes labeled 1 through  $n$ . Each node in the graph represents an agent, and two agents can communicate only if their nodes are adjacent. In our setting, this means that Agent  $i$  can only communicate with Agents  $i-1$  and  $i+1 \pmod n$ , for  $i \in \llbracket 1, n \rrbracket$ . We assume that each agent only knows its operators in (1). Specifically, we assume that only Agent 1 knows the operator  $A_1$  and that, for each  $i \in \{2, \dots, n\}$ , only Agent  $i$  knows the operators  $A_i$  and  $B_{i-1}$ . The responsibility of updating  $x_i$  is assigned to Agent  $i$  for all  $i \in \{1, \dots, n\}$  and the responsibility of updating  $z_i$  is assigned to Agent  $i$  for  $i \in \{2, \dots, n\}$ . Altogether, this gives rise to the protocol for distributed decentralised implementation of (8) described in Algorithm 1.



---

**Algorithm 1** Protocol for distributed decentralised implementation of (8).

---

**Require:** Let  $\lambda \in (0, \frac{2}{L})$  and  $\gamma \in (0, 1 - \frac{\lambda L}{2})$ .

- 1: For each  $i \in \llbracket 2, n \rrbracket$ , Agent  $i$  chooses  $z_{i-1}^0 \in \mathcal{H}$  and sends it to Agent  $i - 1$ .
- 2: **for**  $k = 0, 1, \dots$  **do**
- 3:     Agent 1 computes  $x_1^k = J_{\lambda A_1}(z_1^k)$  and sends it to Agents 2 and  $n$
- 4:     **for**  $i = 2, \dots, n - 1$  **do**
- 5:         Agent  $i$  computes

$$\begin{cases} x_i^k = J_{\lambda A_i}(z_i^k + x_{i-1}^k - z_{i-1}^k - \lambda B_{i-1}(x_{i-1}^k)) \\ z_{i-1}^{k+1} = z_{i-1}^k + \gamma(x_i^k - x_{i-1}^k), \end{cases}$$

      sends  $x_i^k$  to Agent  $i + 1$  and sends  $z_{i-1}^{k+1}$  to Agent  $i - 1$ ;

6:     **end for**

7:     Agent  $n$  computes

$$\begin{cases} x_n^k = J_{\lambda A_n}(x_1^k + x_{n-1}^k - z_{n-1}^k - \lambda B_{n-1}(x_{n-1}^k)) \\ z_{n-1}^{k+1} = z_{n-1}^k + \gamma(x_n^k - x_{n-1}^k), \end{cases}$$

      sends  $z_{n-1}^k$  to Agent  $n - 1$ ;

8: **end for**

---

**Remark 1** (Termination criterion for Algorithm 1) Let  $(\mathbf{z}^k)$  be the sequence generated by Algorithm 1. In order to detect termination, one could compute (possibly periodically) the residual given by

$$\|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2 = \sum_{i=1}^{n-1} \|z_i^{k+1} - z_i^k\|^2.$$

The structure of this residual is suitable for the distributed implementation within the protocol in the algorithm. Indeed, the  $i$ -th term in the sum, given by  $\|z_i^{k+1} - z_i^k\|^2$ , can already be computed by Agent  $i + 1$ , and therefore the full residual  $\|\mathbf{z}^{k+1} - \mathbf{z}^k\|^2$  can be computed by a global summation and broadcast operation (which is compatible with the existing communication pattern, with the addition of one extra dimension for carrying the sum). The same stopping criterion can also be applied to the algorithm presented in Sect. 4 generated by the iteration given in (32a) and (32b).

In order to analyse convergence of (8), we introduce the underlying fixed point operator  $T : \mathcal{H}^{n-1} \rightarrow \mathcal{H}^{n-1}$  given by

$$T(\mathbf{z}) := \mathbf{z} + \gamma \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix}, \quad (9)$$

where  $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$  depends on  $\mathbf{z} = (z_1, \dots, z_{n-1}) \in \mathcal{H}^{n-1}$  and is given by

$$\begin{cases} x_1 = J_{\lambda A_1}(z_1), \\ x_i = J_{\lambda A_i}(z_i + x_{i-1} - z_{i-1} - \lambda B_{i-1}(x_{i-1})) \quad \forall i \in \llbracket 2, n-1 \rrbracket, \\ x_n = J_{\lambda A_n}(x_1 + x_{n-1} - z_{n-1} - \lambda B_{n-1}(x_{n-1})). \end{cases} \quad (10)$$

In this way, the sequence  $(\mathbf{z}^k)$  given by (8a) satisfies  $\mathbf{z}^{k+1} = T(\mathbf{z}^k)$  for all  $k \in \mathbb{N}$ .

**Remark 2** Note that, although the sum of cocoercive operators is cocoercive (see, e.g., [16, Proposition 4.12]), considering the sum of  $n-1$  operators in (1) gives the freedom of either applying each operator as a forward step before the corresponding backward step, or to apply the sum of all of them before a particular backward step (by setting all the operators to be equal to zero except for one of them, which would be equal to the sum).

**Remark 3** (Special cases) If  $n = 2$ , then  $x_1 = x_{n-1}$  and  $T$  in (9) recovers the operator corresponding to *Davis–Yin splitting* [4–6] for finding a zero of  $A_1 + A_2 + B_1$ . In turn, this includes the *forward-backward algorithm* and *Douglas–Rachford splitting* as special cases by further taking  $A_1 = 0$  or  $B_1 = 0$ , respectively.

If  $B_1 = \dots = B_{n-1} = 0$ , then  $T$  in (9) reduces to the resolvent splitting algorithms proposed by the authors in [14]. This has been further studied in [19] for the particular case in which the operators  $A_i$  are normal cones of closed linear subspaces.

Although the number of set-valued and single-valued monotone operators in (7) differ by one, it is straightforward to derive a scheme where this is not the case by setting  $A_1 = 0$ . In this case,  $x_1 = J_{\lambda A_1}(z_1) = z_1$  can be used to eliminate  $x_1$  so that (9) and (10) respectively become

$$T(\mathbf{z}) := \mathbf{z} + \gamma \begin{pmatrix} x_2 - z_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix},$$

where

$$\begin{cases} x_2 = J_{\lambda A_2}(z_2 - \lambda B_1(z_1)), \\ x_i = J_{\lambda A_i}(z_i + x_{i-1} - z_{i-1} - \lambda B_{i-1}(x_{i-1})) \quad \forall i \in \llbracket 3, n-1 \rrbracket, \\ x_n = J_{\lambda A_n}(z_1 + x_{n-1} - z_{n-1} - \lambda B_{n-1}(x_{n-1})). \end{cases}$$

While at first it may seem unusual that the number of set-valued and single-valued monotone operators in (7) are not the same, we note that this same situation arises in Davis–Yin splitting as described above.

**Remark 4** The algorithm given by (8) appears to be new even in the special case with  $A_i = 0$  and  $B_i = \nabla f_i$  for convex smooth functions  $f_i$ . In this case, one of the most popular algorithms for solving  $\min_x \sum_i f_i(x)$  in a decentralised way is EXTRA, proposed in [20]. They are similar in spirit, but also have quite different properties. In particular, the main update of EXTRA is

$$\mathbf{x}^{k+1} = (\text{Id} + W)\mathbf{x}^k - \tilde{W}\mathbf{x}^{k-1} - \lambda[\nabla f(\mathbf{x}^k) - \nabla f(\mathbf{x}^{k-1})],$$

where  $W$  and  $\tilde{W}$  are certain mixing matrices and  $\mathbf{x}^1 = W\mathbf{x}^0 - \lambda\nabla f(\mathbf{x}^0)$ . Undoubtedly, an advantage of EXTRA is the ability to use a wider range of mixing matrices which, in terms of communication, generalises better for network topology.

In what follows, we first describe the relationship between the solutions of the monotone inclusion (7) and the fixed point set of the operator  $T$  in (9).

**Lemma 1** *Let  $n \geq 2$  and  $\gamma, \lambda > 0$ . The following assertions hold.*

- (i) *If  $\bar{x} \in \text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right)$ , then there exists  $\bar{z} \in \text{Fix } T$ .*
- (ii) *If  $(\bar{z}_1, \dots, \bar{z}_{n-1}) \in \text{Fix } T$ , then  $\bar{x} := J_{\lambda A_1}(\bar{z}_1) \in \text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right)$ . Moreover,*

$$\bar{x} = J_{\lambda A_i}(\bar{z}_i - \bar{z}_{i-1} + \bar{x} - \lambda B_{i-1}(\bar{x})) = J_{\lambda A_n}(2\bar{x} - \bar{z}_{n-1} - \lambda B_{n-1}(\bar{x})), \quad (11)$$

for all  $i \in \llbracket 2, n-1 \rrbracket$ .

Consequently,

$$\text{Fix } T \neq \emptyset \iff \text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right) \neq \emptyset.$$

**Proof** (i): Let  $\bar{x} \in \text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right)$ . Then there exists  $\mathbf{v} = (v_1, \dots, v_n) \in \mathcal{H}^n$  such that  $v_i \in A_i(\bar{x})$  and  $\sum_{i=1}^n v_i + \sum_{i=1}^{n-1} B_i(\bar{x}) = 0$ . Define the vector  $\bar{z} = (\bar{z}_1, \dots, \bar{z}_{n-1}) \in \mathcal{H}^{n-1}$  according to

$$\begin{cases} \bar{z}_1 := \bar{x} + \lambda v_1 \in (\text{Id} + \lambda A_1)\bar{x}, \\ \bar{z}_i := \lambda v_i + \bar{z}_{i-1} + \lambda B_{i-1}(\bar{x}) \in (\text{Id} + \lambda A_i)(\bar{x}) - \bar{x} + \bar{z}_{i-1} + \lambda B_{i-1}(\bar{x}), \end{cases}$$

for  $i \in \llbracket 2, n-1 \rrbracket$ . Then  $\bar{x} = J_{\lambda A_1}(\bar{z}_1)$  and  $\bar{x} = J_{\lambda A_i}(\bar{z}_i - \bar{z}_{i-1} + \bar{x} - \lambda B_{i-1}(\bar{x}))$  for  $i \in \llbracket 2, n-1 \rrbracket$ . Furthermore, we have

$$\begin{aligned}
 (\text{Id} + \lambda A_n)(\bar{x}) \ni \bar{x} + \lambda v_n &= \bar{x} - \lambda v_1 - \lambda \sum_{i=2}^{n-1} (v_i + B_{i-1}(\bar{x})) - \lambda B_{n-1}(\bar{x}) \\
 &= \bar{x} - (z_1 - \bar{x}) - \sum_{i=2}^{n-1} (\bar{z}_i - \bar{z}_{i-1}) - \lambda B_{n-1}(\bar{x}) \\
 &= 2\bar{x} - \bar{z}_{n-1} - \lambda B_{n-1}(\bar{x}),
 \end{aligned}$$

which implies that  $\bar{x} = J_{\lambda A_n}(2\bar{x} - \bar{z}_{n-1} - \lambda B_{n-1}(\bar{x}))$ . Altogether, it follows that  $\bar{\mathbf{z}} \in \text{Fix } T$ .

(ii): Let  $\bar{\mathbf{z}} \in \text{Fix } T$  and set  $\bar{x} := J_{\lambda A_1}(\bar{z}_1)$ . Then (11) holds thanks to the definition of  $T$ . The definition of the resolvent therefore implies

$$\begin{cases} \lambda A_1(\bar{x}) \ni \bar{z}_1 - \bar{x}, \\ \lambda A_i(\bar{x}) \ni \bar{z}_i - \bar{z}_{i-1} - \lambda B_{i-1}(\bar{x}) \quad \forall i \in \llbracket 2, n-1 \rrbracket, \\ \lambda A_n(\bar{x}) \ni \bar{x} - z_{n-1} - \lambda B_{n-1}(\bar{x}). \end{cases}$$

Summing together the above inclusions gives  $\bar{x} \in \text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right)$ , as claimed.  $\square$

Next, we study the nonexpansivity properties of the operator  $T$  in (9).

**Lemma 2** For all  $\mathbf{z} = (z_1, \dots, z_n) \in \mathcal{H}^{n-1}$  and  $\bar{\mathbf{z}} = (\bar{z}_1, \dots, \bar{z}_n) \in \mathcal{H}^{n-1}$ , we have

$$\begin{aligned}
 \|T(\mathbf{z}) - T(\bar{\mathbf{z}})\|^2 &+ \left( \frac{1-\gamma}{\gamma} - \frac{\lambda L}{2\gamma} \right) \|(\text{Id} - T)(\mathbf{z}) - (\text{Id} - T)(\bar{\mathbf{z}})\|^2 \\
 &+ \frac{1}{\gamma} \left\| \sum_{i=1}^{n-1} (\text{Id} - T)(\mathbf{z})_i - \sum_{i=1}^{n-1} (\text{Id} - T)(\bar{\mathbf{z}})_i \right\|^2 \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2.
 \end{aligned} \tag{12}$$

In particular, if  $\lambda \in (0, \frac{2}{L})$  and  $\gamma \in (0, 1 - \frac{\lambda L}{2})$ , then  $T$  is  $\alpha$ -averaged for  $\alpha = \frac{2\gamma}{2-\lambda L} \in (0, 1)$ .

**Proof** This proof mainly uses the monotonicity property of the operators  $A_1, \dots, A_n$  together with the cocoercivity property of the operators  $B_1, \dots, B_{n-1}$  to obtain some bounds which yield (12), from where the averagedness of operator  $T$  can be directly deduced. For convenience, denote  $\mathbf{z}^+ := T(\mathbf{z})$  and  $\bar{\mathbf{z}}^+ := T(\bar{\mathbf{z}})$ . Further, let  $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$  be given by (10) and let  $\bar{\mathbf{x}} = (\bar{x}_1, \dots, \bar{x}_n) \in \mathcal{H}^n$  be given analogously. Since  $z_1 - x_1 \in \lambda A_1(x_1)$  and  $\bar{z}_1 - \bar{x}_1 \in \lambda A_1(\bar{x}_1)$ , monotonicity of  $\lambda A_1$  implies

$$\begin{aligned}
 0 &\leq \langle x_1 - \bar{x}_1, (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1) \rangle \\
 &= \langle x_2 - \bar{x}_1, (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1) \rangle + \langle x_1 - x_2, (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1) \rangle.
 \end{aligned} \tag{13}$$

For  $i \in \llbracket 2, n-1 \rrbracket$ ,  $z_i - z_{i-1} + x_{i-1} - x_i - \lambda B_{i-1}(x_{i-1}) \in \lambda A_i(x_i)$  and  $\bar{z}_i - \bar{z}_{i-1} + \bar{x}_{i-1} - \bar{x}_i - \lambda B_{i-1}(\bar{x}_{i-1}) \in \lambda A_i(\bar{x}_i)$ . Thus, monotonicity of  $\lambda A_i$  yields

$$\begin{aligned}
 0 &\leq \langle x_i - \bar{x}_i, z_i - z_{i-1} + x_{i-1} - x_i - \lambda B_{i-1}(x_{i-1}) \rangle \\
 &\quad - \langle x_i - \bar{x}_i, \bar{z}_i - \bar{z}_{i-1} + \bar{x}_{i-1} - \bar{x}_i - \lambda B_{i-1}(\bar{x}_{i-1}) \rangle \\
 &= \langle x_i - \bar{x}_i, (z_i - z_{i-1} + x_{i-1} - x_i) - (\bar{z}_i - \bar{z}_{i-1} + \bar{x}_{i-1} - \bar{x}_i) \rangle \\
 &\quad - \lambda \langle x_i - \bar{x}_i, B_{i-1}(x_{i-1}) - B_{i-1}(\bar{x}_{i-1}) \rangle \\
 &= \langle x_{i+1} - \bar{x}_i, (z_i - x_i) - (\bar{z}_i - \bar{x}_i) \rangle + \langle x_i - x_{i+1}, (z_i - x_i) - (\bar{z}_i - \bar{x}_i) \rangle \\
 &\quad - \langle x_i - \bar{x}_{i-1}, (z_{i-1} - x_{i-1}) - (\bar{z}_{i-1} - \bar{x}_{i-1}) \rangle \\
 &\quad - \langle \bar{x}_{i-1} - \bar{x}_i, (z_{i-1} - x_{i-1}) - (\bar{z}_{i-1} - \bar{x}_{i-1}) \rangle \\
 &\quad - \lambda \langle x_i - \bar{x}_i, B_{i-1}(x_{i-1}) - B_{i-1}(\bar{x}_{i-1}) \rangle.
 \end{aligned}$$

Summing this inequality for  $i \in \llbracket 2, n-1 \rrbracket$  and simplifying gives

$$\begin{aligned}
 0 &\leq \langle x_n - \bar{x}_{n-1}, (z_{n-1} - x_{n-1}) - (\bar{z}_{n-1} - \bar{x}_{n-1}) \rangle \\
 &\quad - \langle x_2 - \bar{x}_1, (z_1 - x_1) - (\bar{z}_1 - \bar{x}_1) \rangle + \sum_{i=2}^{n-1} \langle x_i - x_{i+1}, (z_i - x_i) - (\bar{z}_i - \bar{x}_i) \rangle \\
 &\quad - \sum_{i=1}^{n-2} \langle \bar{x}_i - \bar{x}_{i+1}, (z_i - x_i) - (\bar{z}_i - \bar{x}_i) \rangle - \lambda \sum_{i=2}^{n-1} \langle x_i - \bar{x}_i, B_{i-1}(x_{i-1}) - B_{i-1}(\bar{x}_{i-1}) \rangle.
 \end{aligned} \tag{14}$$

Since  $x_1 + x_{n-1} - x_n - z_{n-1} - \lambda B_{n-1}(x_{n-1}) \in \lambda A_n(x_n)$  and  $\bar{x}_1 + \bar{x}_{n-1} - \bar{x}_n - \bar{z}_{n-1} - \lambda B_{n-1}(\bar{x}_{n-1}) \in \lambda A_n(\bar{x}_n)$ , monotonicity of  $\lambda A_n$  gives

$$\begin{aligned}
 0 &\leq \langle x_n - \bar{x}_n, x_1 + x_{n-1} - x_n - z_{n-1} - \lambda B_{n-1}(x_{n-1}) \rangle \\
 &\quad - \langle x_n - \bar{x}_n, \bar{x}_1 + \bar{x}_{n-1} - \bar{x}_n - \bar{z}_{n-1} - \lambda B_{n-1}(\bar{x}_{n-1}) \rangle \\
 &= \langle x_n - \bar{x}_n, (x_1 - x_n) - (\bar{x}_1 - \bar{x}_n) \rangle \\
 &\quad + \langle x_n - \bar{x}_n, (x_{n-1} - z_{n-1}) - (\bar{x}_{n-1} - \bar{z}_{n-1}) \rangle \\
 &\quad - \lambda \langle x_n - \bar{x}_n, B_{n-1}(x_{n-1}) - B_{n-1}(\bar{x}_{n-1}) \rangle \\
 &= -\langle x_n - \bar{x}_{n-1}, (z_{n-1} - x_{n-1}) - (\bar{z}_{n-1} - \bar{x}_{n-1}) \rangle \\
 &\quad + \langle \bar{x}_n - \bar{x}_{n-1}, (z_{n-1} - x_{n-1}) - (\bar{z}_{n-1} - \bar{x}_{n-1}) \rangle \\
 &\quad + \frac{1}{2} (\|x_1 - \bar{x}_1\|^2 - \|x_n - \bar{x}_n\|^2 - \|(x_1 - x_n) - (\bar{x}_1 - \bar{x}_n)\|^2) \\
 &\quad - \lambda \langle x_n - \bar{x}_n, B_{n-1}(x_{n-1}) - B_{n-1}(\bar{x}_{n-1}) \rangle.
 \end{aligned} \tag{15}$$

Adding (13), (14) and (15) and rearranging gives

$$\begin{aligned}
0 \leq & \sum_{i=1}^{n-1} \langle (x_i - \bar{x}_i) - (x_{i+1} - \bar{x}_{i+1}), \bar{x}_i - x_i \rangle \\
& + \sum_{i=1}^{n-1} \langle (x_i - \bar{x}_i) - (x_{i+1} - \bar{x}_{i+1}), z_i - \bar{z}_i \rangle \\
& + \frac{1}{2} (\|x_1 - \bar{x}_1\|^2 - \|x_n - \bar{x}_n\|^2 - \|(x_1 - x_n) - (\bar{x}_1 - \bar{x}_n)\|^2) \\
& - \lambda \sum_{i=1}^{n-1} \langle x_{i+1} - \bar{x}_{i+1}, B_i(x_i) - B_i(\bar{x}_i) \rangle.
\end{aligned} \tag{16}$$

The first term in (16) can be expressed as

$$\begin{aligned}
& \sum_{i=1}^{n-1} \langle (x_i - \bar{x}_i) - (x_{i+1} - \bar{x}_{i+1}), \bar{x}_i - x_i \rangle \\
& = \frac{1}{2} \sum_{i=1}^{n-1} (\|x_{i+1} - \bar{x}_{i+1}\|^2 - \|x_i - \bar{x}_i\|^2 - \|(x_i - x_{i+1}) - (\bar{x}_i - \bar{x}_{i+1})\|^2) \\
& = \frac{1}{2} \left( \|x_n - \bar{x}_n\|^2 - \|x_1 - \bar{x}_1\|^2 - \frac{1}{\gamma^2} \|(\mathbf{z} - \mathbf{z}^+) - (\bar{\mathbf{z}} - \bar{\mathbf{z}}^+)\|^2 \right),
\end{aligned} \tag{17}$$

and the second term in (16) can be written as

$$\begin{aligned}
& \sum_{i=1}^{n-1} \langle (x_i - x_{i+1}) - (\bar{x}_i - \bar{x}_{i+1}), z_i - \bar{z}_i \rangle \\
& = \frac{1}{\gamma} \sum_{i=1}^{n-1} \langle (z_i - z_i^+) - (\bar{z}_i - \bar{z}_i^+), z_i - \bar{z}_i \rangle \\
& = \frac{1}{\gamma} \langle (\mathbf{z} - \mathbf{z}^+) - (\bar{\mathbf{z}} - \bar{\mathbf{z}}^+), \mathbf{z} - \bar{\mathbf{z}} \rangle \\
& = \frac{1}{2\gamma} (\|(\mathbf{z} - \mathbf{z}^+) - (\bar{\mathbf{z}} - \bar{\mathbf{z}}^+)\|^2 + \|\mathbf{z} - \bar{\mathbf{z}}\|^2 - \|\mathbf{z}^+ - \bar{\mathbf{z}}^+\|^2).
\end{aligned} \tag{18}$$

To estimate the last term, Young's inequality and  $\frac{1}{L}$ -cocoercivity of  $B_1, \dots, B_{n-1}$  gives

$$\begin{aligned}
 & - \sum_{i=1}^{n-1} \langle x_{i+1} - \bar{x}_{i+1}, B_i(x_i) - B_i(\bar{x}_i) \rangle \\
 & = \sum_{i=1}^{n-1} \langle (\bar{x}_{i+1} - \bar{x}_i) - (x_{i+1} - x_i), B_i(x_i) - B_i(\bar{x}_i) \rangle \\
 & \quad + \sum_{i=1}^{n-1} \langle \bar{x}_i - x_i, B_i(x_i) - B_i(\bar{x}_i) \rangle \\
 & \leq \frac{L}{4} \sum_{i=1}^{n-1} \|(\bar{x}_{i+1} - \bar{x}_i) - (x_{i+1} - x_i)\|^2 + \frac{1}{L} \sum_{i=1}^{n-1} \|B_i(x_i) - B_i(\bar{x}_i)\|^2 \quad (19) \\
 & \quad - \frac{1}{L} \sum_{i=1}^{n-1} \|B_i(x_i) - B_i(\bar{x}_i)\|^2 \\
 & = \frac{L}{4} \sum_{i=1}^{n-1} \|(\bar{x}_{i+1} - \bar{x}_i) - (x_{i+1} - x_i)\|^2 \\
 & = \frac{L}{4\gamma^2} \|(\mathbf{z} - \mathbf{z}^+) - (\bar{\mathbf{z}} - \bar{\mathbf{z}}^+)\|^2.
 \end{aligned}$$

Thus, substituting (17) and (18) into (16), using (19) and simplifying gives the claimed inequality (12). Finally, to show that (12) implies  $T$  is  $\alpha$ -averaged with  $\alpha := \frac{2\gamma}{2-\lambda L}$ , note that  $\alpha \in (0, 1)$  and satisfies  $\frac{1-\alpha}{\alpha} = \frac{1-\gamma}{\gamma} - \frac{\lambda L}{2\gamma}$ . This completes the proof.  $\square$

The following theorem is our main convergence result regarding the algorithm given by (8).

**Theorem 3** *Let  $n \geq 2$ , let  $A_1, \dots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$  be maximally monotone and let  $B_1, \dots, B_{n-1} : \mathcal{H} \rightarrow \mathcal{H}$  be  $\frac{1}{L}$ -cocoercive with  $\text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right) \neq \emptyset$ . Further, let  $\lambda \in (0, \frac{2}{L})$  and  $\gamma \in (0, 1 - \frac{\lambda L}{2})$ . Given  $\mathbf{z}^0 \in \mathcal{H}^{n-1}$ , let  $(\mathbf{z}^k) \subseteq \mathcal{H}^{n-1}$  and  $(\mathbf{x}^k) \subseteq \mathcal{H}^n$  be the sequences given by (8). Then the following assertions hold.*

- (i) *The sequence  $(\mathbf{z}^k)$  converges weakly to a point  $\mathbf{z} \in \text{Fix } T$ .*
- (ii) *The sequence  $(\mathbf{x}^k)$  converges weakly to a point  $(x, \dots, x) \in \mathcal{H}^n$  with  $x \in \text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right)$ .*
- (iii) *The sequence  $(B_i(x_i^k))$  converges strongly to  $B_i(x)$  for all  $i \in \llbracket 1, n-1 \rrbracket$ .*

**Proof** (a): Since  $\text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right) \neq \emptyset$ , Lemma 1(i) implies  $\text{Fix } T \neq \emptyset$ . Since  $\lambda \in (0, \frac{2}{L})$  and  $\gamma \in (0, 1 - \frac{\lambda L}{2})$ , Lemma 2 implies  $T$  is averaged nonexpansive.

By applying [16, Theorem 5.15], we deduce that  $(\mathbf{z}^k)$  converges weakly to a point  $\mathbf{z} \in \text{Fix } T$  and that  $\lim_{k \rightarrow \infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$ .

(ii): By nonexpansivity of resolvents,  $L$ -Lipschitz continuity of  $B_1, \dots, B_{n-1}$ , and boundedness of  $(\mathbf{z}^k)$ , it follows that  $(\mathbf{x}^k)$  is also bounded. Further, (9) and the fact that  $\lim_{k \rightarrow \infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$  implies that

$$\lim_{k \rightarrow \infty} \|x_i^k - x_{i-1}^k\| = 0 \quad \forall i = 2, \dots, n. \quad (20)$$

Next, using the definition of the resolvent together with (8b), we have

$$S \begin{pmatrix} z_1^k - x_1^k \\ (z_2^k - x_2^k) - (z_1^k - x_1^k) + \lambda b_2^k \\ \vdots \\ (z_{n-1}^k - x_{n-1}^k) - (z_{n-2}^k - x_{n-2}^k) + \lambda b_{n-1}^k \\ x_n^k \end{pmatrix} \ni \begin{pmatrix} x_1^k - x_n^k \\ x_2^k - x_n^k \\ \vdots \\ x_{n-1}^k - x_{n-1}^k \\ x_1^k - x_n^k + \lambda \sum_{i=1}^n b_{i+1}^k \end{pmatrix}, \quad (21)$$

where  $b_i^k := B_{i-1}(x_i^k) - B_{i-1}(x_{i-1}^k)$  and the operator  $S : \mathcal{H}^n \rightrightarrows \mathcal{H}^n$  is given by

$$S := \begin{pmatrix} (\lambda A_1)^{-1} \\ (\lambda(A_2 + B_1))^{-1} \\ \vdots \\ (\lambda(A_{n-1} + B_{n-2}))^{-1} \\ \lambda(A_n + B_{n-1}) \end{pmatrix} + \begin{pmatrix} 0 & 0 & \dots & 0 & -\text{Id} \\ 0 & 0 & \dots & 0 & -\text{Id} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & -\text{Id} \\ \text{Id} & \text{Id} & \dots & \text{Id} & 0 \end{pmatrix}. \quad (22)$$

As the sum of two maximally monotone operators is again maximally monotone provided that one of the operators has full domain [16, Corollary 24.4(i)], it follows that  $S$  is maximally monotone. Consequently, it is demiclosed [16, Proposition 20.38]. That is, its graph is sequentially closed in the weak-strong topology.

Let  $\mathbf{w} \in \mathcal{H}^n$  be an arbitrary weak cluster point of the sequence  $(\mathbf{x}^k)$ . As a consequence of (20),  $\mathbf{w} = (x, \dots, x)$  for some  $x \in \mathcal{H}$ . Taking the limit along a subsequence of  $(\mathbf{x}^k)$  which converges weakly to  $\mathbf{w}$  in (21), using demiclosedness of  $S$  together with  $L$ -Lipschitz continuity of  $B_1, \dots, B_{n-1}$ , and unravelling the resulting expression gives

$$\begin{cases} \lambda A_1(x) \ni z_1 - x, \\ \lambda(A_i + B_{i-1})(x) \ni z_i - z_{i-1} \quad \forall i \in \llbracket 2, n-1 \rrbracket, \\ \lambda(A_n + B_{n-1})(x) \ni x - z_{n-1}, \end{cases}$$

which implies  $\mathbf{z} \in \text{Fix } T$  and  $x = J_{A_1}(z_1) \in \text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right)$ .

In other words,  $\mathbf{w} = (x, \dots, x) \in \mathcal{H}^n$  with  $x := J_{A_1}(z_1)$  is the unique weak sequential cluster point of the bounded sequence  $(\mathbf{x}^k)$ . We therefore deduce that  $(\mathbf{x}^k)$  converges weakly to  $\mathbf{w}$ , which completes this part of the proof.

(iii): For convenience, denote  $\mathbf{y}^k = (y_1^k, \dots, y_n^k)$  where



$$\begin{cases} y_1^k := z_1^k, \\ y_i^k := z_i^k + x_{i-1}^k - z_{i-1}^k - \lambda B_{i-1}(x_{i-1}^k) \quad \forall i \in \llbracket 2, n-1 \rrbracket, \\ y_n^k := x_1^k + x_{n-1}^k - z_{n-1}^k - \lambda B_{n-1}(x_{n-1}^k), \end{cases}$$

so that  $x_i^k = J_{\lambda A_i}(y_i^k)$  for all  $i \in \llbracket 1, n \rrbracket$ . Define  $\mathbf{y} = (y_1, \dots, y_n)$  in an analogous way with  $\mathbf{z}$  in place of  $\mathbf{z}^k$  and  $(x, \dots, x)$  in place of  $\mathbf{x}^k$ , so that  $x = J_{\lambda A_i}(y_i)$  for all  $i \in \llbracket 1, n \rrbracket$ .

Using firm nonexpansivity of resolvents yields

$$\begin{aligned} 0 &\leq \sum_{i=1}^n \langle J_{\lambda A_i}(y_i^k) - J_{\lambda A_i}(y_i), (\text{Id} - J_{\lambda A_i})(y_i^k) - (\text{Id} - J_{\lambda A_i})(y_i) \rangle \\ &= \langle x_1^k - x, (z_1^k - x_1^k) - (z_1 - x) \rangle \\ &\quad + \sum_{i=2}^{n-1} \langle x_i^k - x, (z_i^k - x_i^k) - (z_{i-1}^k - x_{i-1}^k) - \lambda B_{i-1}(x_{i-1}^k) \rangle \\ &\quad - \sum_{i=2}^{n-1} \langle x_i^k - x, z_i - z_{i-1} - \lambda B_{i-1}(x) \rangle \\ &\quad + \langle x_n^k - x, x_1^k - x_n^k - (z_{n-1}^k - x_{n-1}^k) \\ &\quad - \lambda B_{n-1}(x_{n-1}^k) \rangle - \langle x_n^k - x, x - z_{n-1} - \lambda B_{n-1}(x) \rangle \\ &= \langle x_1^k - x_n^k, (z_1^k - x_1^k) - (z_1 - x) \rangle + \langle x_n^k - x, (z_1^k - x_1^k) - (z_1 - x) \rangle \\ &\quad + \sum_{i=2}^{n-1} \langle x_i^k - x_n^k, (z_i^k - x_i^k) - (z_{i-1}^k - x_{i-1}^k) - (z_i - z_{i-1}) \rangle \\ &\quad + \langle x_n^k - x, (z_{n-1}^k - x_{n-1}^k) - (z_1^k - x_1^k) - (z_{n-1} - z_1) \rangle \\ &\quad - \lambda \sum_{i=1}^{n-1} \langle x_{i+1}^k - x_i^k, B_i(x_i^k) - B_i(x) \rangle - \lambda \sum_{i=1}^{n-1} \langle x_i^k - x, B_i(x_i^k) - B_i(x) \rangle \\ &\quad + \langle x_n^k - x, x_1^k - x_n^k \rangle - \langle x_n^k - x, (z_{n-1}^k - x_{n-1}^k) + (x - z_{n-1}) \rangle. \end{aligned} \quad (23)$$

Rearranging (23) followed by applying  $\frac{1}{L}$ -cocoercivity of  $B_1, \dots, B_{n-1}$  gives

$$\begin{aligned} &\langle x_n^k - x, x_1^k - x_n^k \rangle + \langle x_1^k - x_n^k, (z_1^k - x_1^k) - (z_1 - x) \rangle \\ &\quad - \lambda \sum_{i=1}^{n-1} \langle x_{i+1}^k - x_i^k, B_i(x_i^k) - B_i(x) \rangle \\ &\quad + \sum_{i=2}^{n-1} \langle x_i^k - x_n^k, ((z_i^k - x_i^k) - (z_{i-1}^k - x_{i-1}^k)) - (z_i - z_{i-1}) \rangle \\ &\geq \lambda \sum_{i=1}^{n-1} \langle x_i^k - x, B_i(x_i^k) - B_i(x) \rangle \geq \frac{\lambda}{L} \sum_{i=1}^{n-1} \|B_i(x_i^k) - B_i(x)\|^2. \end{aligned} \quad (24)$$

Note that the left-hand side of (24) converges to zero due to (20) and the boundedness of sequences  $(\mathbf{z}^k)$ ,  $(\mathbf{x}^k)$  and  $(B_i(x_i^k))$  for  $i \in \llbracket 1, n-1 \rrbracket$ . It then follows that  $B_i(x_i^k) \rightarrow B_i(x)$  for all  $i \in \llbracket 1, n-1 \rrbracket$ , as claimed.  $\square$

**Remark 5** (Attouch–Théra duality) Let  $I \subseteq \{1, \dots, n-1\}$  be a non-empty index set with cardinality denoted by  $|I|$ . Express the monotone inclusion (1) as

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i \in I} B_i(x) + \left( \sum_{i=1}^n A_i + \sum_{i \notin I} B_i \right)(x), \quad (25)$$

and note that the first operator  $\sum_{i \in I} B_i$  is  $\frac{1}{|I|L}$ -cocoercive (see, e.g., [16, Proposition 4.12]). The *Attouch–Théra dual* [21] associated with (25) takes the form

$$\text{find } u \in \mathcal{H} \text{ such that } 0 \in \left( \sum_{i \in I} B_i \right)^{-1}(u) - \left( \sum_{i=1}^n A_i + \sum_{i \notin I} B_i \right)^{-1}(-u), \quad (26)$$

where we note that the first operator  $\left( \sum_{i \in I} B_i \right)^{-1}$  is  $\frac{1}{|I|L}$ -strongly monotone. Hence, as a strongly monotone inclusion, (26) has a unique solution  $\bar{u} \in \mathcal{H}$ . Moreover, for any solution  $\bar{x} \in \mathcal{H}$  of (25), [21, Theorem 3.1] implies  $\bar{u} = \left( \sum_{i \in I} B_i \right)(\bar{x})$ . In the context of the previous result, Theorem 3(c) implies  $\sum_{i \in I} B_i(x_i^k) \rightarrow \bar{u}$  as  $k \rightarrow \infty$ . In other words, the algorithm in (8) also produces a sequence which converges strongly to the unique solution of the dual inclusion (26).

**Remark 6** (i) When  $B_1 = \dots = B_{n-1} = 0$ , Theorem 3 recovers [14, Theorem 4.5].

(ii) In the special case when  $n = 2$ , (12) from Lemma 2 simplifies to give the stronger inequality

$$\|T(\mathbf{z}) - T(\bar{\mathbf{z}})\|^2 + \left( \frac{2-\gamma}{\gamma} - \frac{\lambda L}{2\gamma} \right) \|(\text{Id} - T)(\mathbf{z}) - (\text{Id} - T)(\bar{\mathbf{z}})\|^2 \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2. \quad (27)$$

This assures averagedness of  $T$  provided that  $\gamma \in (0, 2 - \frac{\lambda L}{2})$ , which is larger than the range of permissible values for  $\gamma$  in the statement of Theorem 3. However, by using (27), a proof similar to that of Theorem 3 guarantees the convergence for a larger range of parameter values, namely, when  $\lambda \in (0, \frac{4}{L})$  and  $\gamma \in (0, 2 - \frac{\lambda L}{2})$ . For details, see [5, 6].

## 4 A distributed forward-reflected-backward method

Let  $n \geq 3$  and consider the problem

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-2} B_i \right)(x), \quad (28)$$

where  $A_1, \dots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$  are maximally monotone and  $B_1, \dots, B_{n-2} : \mathcal{H} \rightarrow \mathcal{H}$  are monotone and  $L$ -Lipschitz continuous.

Developing splitting algorithms which use forward evaluations of Lipschitz continuous monotone operators is generally more intricate than those exploiting cocoercivity, such as the one in the previous section. For concreteness, consider the special case of (28) with two operators given by

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in (A_1 + B_1)(x). \quad (29)$$

It is well known that the *forward-backward method* for (29) given by

$$x^{k+1} = J_{\lambda A_1}(x^k - \lambda B_1(x^k)) \quad (30)$$

fails to converge for any  $\lambda > 0$ . Indeed, consider the particular instance of (29) given by  $\mathcal{H} = \mathbb{R}^2$ ,  $A_1 := 0$  and  $B_1 := \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ , whose unique solution is  $(0, 0)^T$ . Then,  $B_1$  is skew-symmetric and thus monotone (but not cocoercive), but the sequence generated by (30) will diverge for any non-zero starting point, since the eigenvalues of  $\text{Id} - \lambda B_1$  are  $1 \pm \lambda i$ . However, a small modification of (30) gives rise to

$$x^{k+1} = J_{\lambda A_1}(x^k - 2\lambda B_1(x^k) + \lambda B_1(x^{k-1})), \quad (31)$$

which is known as the *forward-reflected-backward method* [13]. Unlike (30), it converges for any  $\lambda < \frac{1}{2L}$ . While (31) is not the only constant stepsize scheme for solving (29), as there are a few which are fundamentally different [3, 22], it is arguably one of the simplest. In this section, we develop a modification of the method from the previous section which converges for Lipschitz continuous operators by drawing inspiration from the differences between (31) and (30).

Given  $\lambda \in (0, \frac{1}{2L})$  and  $\gamma \in (0, 1 - 2\lambda L)$  and an initial point  $\mathbf{z}^0 = (z_1^0, \dots, z_{n-1}^0) \in \mathcal{H}^{n-1}$ , our proposed algorithm for (28) generates two sequences,  $(\mathbf{z}^k) \subseteq \mathcal{H}^{n-1}$  and  $(\mathbf{x}^k) \subseteq \mathcal{H}^n$ , according to

$$\mathbf{z}^{k+1} = \mathbf{z}^k + \gamma \begin{pmatrix} x_2^k - x_1^k \\ x_3^k - x_2^k \\ \vdots \\ x_n^k - x_{n-1}^k \end{pmatrix} \quad (32a)$$

and

$$\begin{cases} x_1^k = J_{\lambda A_1}(z_1^k), \\ x_2^k = J_{\lambda A_2}(z_2^k + x_1^k - z_1^k - \lambda B_1(x_1^k)), \\ x_i^k = J_{\lambda A_i}(z_i^k + x_{i-1}^k - z_{i-1}^k - \lambda B_{i-1}(x_{i-1}^k) - \lambda(B_{i-2}(x_{i-1}^k) - B_{i-2}(x_{i-2}^k))), \\ x_n^k = J_{\lambda A_n}(x_1^k + x_{n-1}^k - z_{n-1}^k - \lambda(B_{n-2}(x_{n-1}^k) - B_{n-2}(x_{n-2}^k))). \end{cases} \quad (32b)$$

for  $i \in \llbracket 3, n-1 \rrbracket$ .

Compared to the algorithm proposed in the previous section, the only major change here is that some expressions for  $x_i^k$  in (32b) incorporate a “reflection-type” term involving the operator  $B_{i-2}$ . This precise form seems important for our subsequence convergence analysis and it seems not easy to incorporate “reflection-type” terms involving the operator  $B_{i-1}$ . The structure of (32) allows for a similar protocol to the one described in Algorithm 1 to be used for a distributed decentralised implementation. The only change to the protocol (in terms of communication) is that Agent  $i$  must also now send  $\lambda(B_{i-1}(x_i^k) - B_{i-1}(x_{i-1}^k))$  to Agent  $i + 1$  for all  $i \in \llbracket 2, n - 1 \rrbracket$ .

**Remark 7** To the best of our knowledge, the scheme given by (32) does not directly recover any existing forward-backward-type scheme as special case (although it is clearly related to (31)). For example, if we take  $n = 3$  and  $A_1 = A_3 = 0$ . Then  $x_1^k$  and  $x_3^k$  can be eliminated from (32) to give

$$\begin{cases} x_2^k = J_{\lambda A_2}(z_2^k - \lambda B_1(z_1^k)) \\ z_1^{k+1} = z_1^k + \gamma(x_2^k - z_1^k) \\ z_2^{k+1} = z_2^k + \gamma(z_1^k - z_2^k - \lambda(B_1(x_2^k) - B_1(z_1^k))). \end{cases}$$

To better understand the relationship between this and (31), it is instructive to consider the limiting case with  $\gamma = 1$ . Indeed, when  $\gamma = 1$ ,  $x_2^k$  and  $z_2^k$  can be eliminated to give

$$z_1^{k+1} = J_{\lambda A_2}(z_1^{k-1} - 2\lambda B_1(z_1^k) + \lambda B_1(z_1^{k-1})).$$

Although this closely resembles (31) for finding zero of  $A_2 + B_1$ , it is not exactly the same due to the index of the first term inside the resolvent.

In order to analyse (32), we introduce the underlying fixed point operator  $\tilde{T} : \mathcal{H}^{n-1} \rightarrow \mathcal{H}^{n-1}$  given by

$$\tilde{T}(\mathbf{z}) := \mathbf{z} + \gamma \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix}, \quad (33)$$

where  $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$  depends on  $\mathbf{z} = (z_1, \dots, z_n) \in \mathcal{H}$  and is given by

$$\begin{cases} x_1 = J_{\lambda A_1}(z_1), \\ x_2 = J_{\lambda A_2}(z_2 + x_1 - z_1 - \lambda B_1(x_1)), \\ x_i = J_{\lambda A_i}(z_i + x_{i-1} - z_{i-1} - \lambda B_{i-1}(x_{i-1}) - \lambda(B_{i-2}(x_{i-1}) - B_{i-2}(x_{i-2}))), \\ x_n = J_{\lambda A_n}(x_1 + x_{n-1} - z_{n-1} - \lambda(B_{n-2}(x_{n-1}) - B_{n-2}(x_{n-2}))), \end{cases} \quad (34)$$

for  $i \in \llbracket 3, n-1 \rrbracket$ . In this way, the sequence  $(\mathbf{z}^k)$  given by (32) satisfies  $\mathbf{z}^{k+1} = \tilde{T}(\mathbf{z}^k)$  for all  $k \in \mathbb{N}$ .

Next, we analyse the nonexpansivity properties of the operator  $\tilde{T}$ . The proof of the following result is similar to that of Lemma 2, but using the Lipschitzian properties of the operators  $B_1, \dots, B_{n-2}$  instead of cocoercivity.

**Lemma 3** *Let  $\bar{\mathbf{z}} = (\bar{z}_1, \dots, \bar{z}_{n-1}) \in \text{Fix } \tilde{T}$ . Then, for all  $\mathbf{z} = (z_1, \dots, z_{n-1}) \in \mathcal{H}^{n-1}$ , we have*

$$\begin{aligned} \|\tilde{T}(\mathbf{z}) - \bar{\mathbf{z}}\|^2 + \left( \frac{1-\gamma}{\gamma} - \frac{2\lambda L}{\gamma} \right) \|(\text{Id} - \tilde{T})(\mathbf{z})\|^2 + \frac{1}{\gamma} \left\| \sum_{i=1}^{n-1} (\text{Id} - \tilde{T})(\mathbf{z})_i \right\|^2 \\ + \gamma \lambda L \|(\text{Id} - \tilde{T})(\mathbf{z})_1\|^2 + \gamma \lambda L \|(\text{Id} - \tilde{T})(\mathbf{z})_{n-1}\|^2 \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2. \end{aligned} \quad (35)$$

In particular, if  $\lambda \in (0, \frac{1}{2L})$  and  $\gamma \in (0, 1 - 2\lambda L)$ , then  $\tilde{T}$  is  $\sigma$ -strongly quasi-nonexpansive for  $\sigma = \frac{1-\gamma}{\gamma} - \frac{2\lambda L}{\gamma} > 0$ .

**Proof** For convenience, denote  $\mathbf{z}^+ = \tilde{T}(\mathbf{z})$ . Further, let  $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$  be given by (34) and let  $\bar{\mathbf{x}} = (\bar{x}, \dots, \bar{x}) \in \mathcal{H}^{n-1}$  be given analogously. Note that the expression of  $\bar{\mathbf{x}}$  is justified as  $\bar{\mathbf{z}} = \tilde{T}(\bar{\mathbf{z}})$ . Monotonicity of  $\lambda A_1$  implies

$$0 \leq \langle x_2 - \bar{x}, (z_1 - x_1) - (\bar{z}_1 - \bar{x}) \rangle + \langle x_1 - x_2, (z_1 - x_1) - (\bar{z}_1 - \bar{x}) \rangle. \quad (36)$$

In order to simplify the case study, we introduce the zero operator  $B_0 := 0$ . By monotonicity of  $\lambda A_i$ , we deduce

$$\begin{aligned} 0 \leq & \langle x_{i+1} - \bar{x}, (z_i - x_i) - (\bar{z}_i - \bar{x}) \rangle + \langle x_i - x_{i+1}, (z_i - x_i) - (\bar{z}_i - \bar{x}) \rangle \\ & - \langle x_i - \bar{x}, (z_{i-1} - x_{i-1}) - (\bar{z}_{i-1} - \bar{x}) \rangle \\ & - \lambda \langle x_i - \bar{x}, B_{i-1}(x_{i-1}) - B_{i-1}(\bar{x}) \rangle \\ & - \lambda \langle x_i - \bar{x}, B_{i-2}(x_{i-1}) - B_{i-2}(x_{i-1}) \rangle, \end{aligned} \quad (37)$$

and monotonicity of  $\lambda A_n$  yields

$$\begin{aligned} 0 \leq & -\langle x_n - \bar{x}, (z_{n-1} - x_{n-1}) - (\bar{z}_{n-1} - \bar{x}) \rangle \\ & - \lambda \langle x_n - \bar{x}, B_{n-2}(x_{n-1}) - B_{n-2}(x_{n-2}) \rangle \\ & + \frac{1}{2} (\|x_1 - \bar{x}\|^2 - \|x_n - \bar{x}\|^2 + \|x_1 - x_n\|^2). \end{aligned} \quad (38)$$

Summing together (36)–(38), we obtain the inequality

$$\begin{aligned}
 0 \leq & \sum_{i=1}^{n-1} \langle (\bar{z}_i - \bar{x}) - (z_i - x_i), x_{i+1} - x_i \rangle \\
 & + \frac{1}{2} (\|x_1 - \bar{x}\|^2 - \|x_n - \bar{x}\|^2 + \|x_1 - x_n\|^2) \\
 & - \lambda \sum_{i=2}^{n-1} \langle x_i - \bar{x}, B_{i-1}(x_{i-1}) - B_{i-1}(\bar{x}) \rangle \\
 & - \lambda \sum_{i=3}^n \langle x_i - \bar{x}, B_{i-2}(x_{i-1}) - B_{i-2}(x_{i-2}) \rangle,
 \end{aligned} \tag{39}$$

where we have omitted the index  $i = 2$  in the last sum, since  $B_0 := 0$ . The first term in (39) multiplied by  $2\gamma$  can be written as

$$\begin{aligned}
 2\gamma \sum_{i=1}^{n-1} \langle (\bar{z}_i - \bar{x}) - (z_i - x_i), x_{i+1} - x_i \rangle \\
 = \sum_{i=1}^{n-1} (\|\bar{z}_i - z_i\|^2 + \|z_i^+ - z_i\|^2 - \|z_i^+ - \bar{z}_i\|^2) \\
 - \frac{1}{\gamma} \sum_{i=1}^{n-1} \|z_i^+ - z_i\|^2 + \gamma (\|x_n - \bar{x}\|^2 - \|x_1 - \bar{x}\|^2).
 \end{aligned} \tag{40}$$

Therefore, multiplying (39) by  $2\gamma$  and substituting (40), we reach the inequality

$$\begin{aligned}
 \|\tilde{T}(\mathbf{z}) - \bar{\mathbf{z}}\|^2 + \frac{1-\gamma}{\gamma} \|(\text{Id} - \tilde{T})(\mathbf{z})\|^2 + \frac{1}{\gamma} \left\| \sum_{i=1}^{n-1} (\text{Id} - \tilde{T})(\mathbf{z})_i \right\|^2 \\
 \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2 - 2\gamma\lambda \sum_{i=2}^{n-1} \langle x_i - \bar{x}, B_{i-1}(x_{i-1}) - B_{i-1}(\bar{x}) \rangle \\
 - 2\gamma\lambda \sum_{i=3}^n \langle x_i - \bar{x}, B_{i-2}(x_{i-1}) - B_{i-2}(x_{i-2}) \rangle.
 \end{aligned} \tag{41}$$

Using monotonicity of  $B_1, \dots, B_{n-2}$ , the second last term can be estimated as

$$- \sum_{i=2}^{n-1} \langle x_i - \bar{x}, B_{i-1}(x_{i-1}) - B_{i-1}(\bar{x}) \rangle \leq \sum_{i=2}^{n-1} \langle x_i - \bar{x}, B_{i-1}(x_i) - B_{i-1}(x_{i-1}) \rangle \tag{42}$$

and, using  $L$ -Lipschitz continuity of  $B_1, \dots, B_{n-2}$ , the last term can be estimated as

$$\begin{aligned}
 & - \sum_{i=3}^n \langle x_i - \bar{x}, B_{i-2}(x_{i-1}) - B_{i-2}(x_{i-2}) \rangle \\
 & = - \sum_{i=3}^n \langle x_{i-1} - \bar{x}, B_{i-2}(x_{i-1}) - B_{i-2}(x_{i-2}) \rangle \\
 & \quad + \sum_{i=3}^n \langle x_{i-1} - x_i, B_{i-2}(x_{i-1}) - B_{i-2}(x_{i-2}) \rangle \\
 & \leq - \sum_{i=3}^n \langle x_{i-1} - \bar{x}, B_{i-2}(x_{i-1}) - B_{i-2}(x_{i-2}) \rangle \\
 & \quad + \frac{L}{2} \sum_{i=3}^n (\|x_{i-1} - x_i\|^2 + \|x_{i-1} - x_{i-2}\|^2) \\
 & = - \sum_{i=2}^{n-1} \langle x_i - \bar{x}, B_{i-1}(x_i) - B_{i-1}(x_{i-1}) \rangle + L \sum_{i=2}^n \|x_i - x_{i-1}\|^2 \\
 & \quad - \frac{L}{2} \|x_2 - x_1\|^2 - \frac{L}{2} \|x_n - x_{n-1}\|^2 \\
 & = - \sum_{i=2}^{n-1} \langle x_i - \bar{x}, B_{i-1}(x_i) - B_{i-1}(x_{i-1}) \rangle + \frac{L}{\gamma^2} \|(\text{Id} - \tilde{T})(\mathbf{z})\|^2 \\
 & \quad - \frac{L}{2} \|(\text{Id} - \tilde{T})(\mathbf{z})_1\|^2 - \frac{L}{2} \|(\text{Id} - \tilde{T})(\mathbf{z})_{n-1}\|^2.
 \end{aligned} \tag{43}$$

Thus, substituting (42) and (43) into (41) gives (35), which completes the proof.  $\square$

**Remark 8** Compared to Lemma 2 from the previous section, the conclusions of Lemma 3 are weaker in two ways. Firstly, the permissible stepsize range of  $\lambda \in (0, \frac{1}{2L})$  is smaller than in Lemma 2, which allowed  $\lambda \in (0, \frac{2}{L})$ . And, secondly, the operator  $\tilde{T}$  is only shown to be strongly quasi-nonexpansive in Lemma 3 whereas  $T$  is known to be averaged nonexpansive.

The following theorem is our main result regarding convergence of (32).

**Theorem 4** Let  $n \geq 3$ , let  $A_1, \dots, A_n : \mathcal{H} \rightrightarrows \mathcal{H}$  be maximally monotone and let  $B_1, \dots, B_{n-2} : \mathcal{H} \rightarrow \mathcal{H}$  be monotone and  $L$ -Lipschitz continuous with  $\text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-2} B_i \right) \neq \emptyset$ . Further, let  $\lambda \in (0, \frac{1}{2L})$  and  $\gamma \in (0, 1 - 2\lambda L)$ . Given  $\mathbf{z}^0 \in \mathcal{H}^{n-1}$ , let  $(\mathbf{z}^k) \subseteq \mathcal{H}^{n-1}$  and  $(\mathbf{x}^k) \subseteq \mathcal{H}^n$  be the sequences given by (32). Then the following assertions hold.

- (i) The sequence  $(\mathbf{z}^k)$  converges weakly to a point  $\mathbf{z} \in \text{Fix } \tilde{T}$ .
- (ii) The sequence  $(\mathbf{x}^k)$  converges weakly to a point  $(x, \dots, x) \in \mathcal{H}^n$  with  $x \in \text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-2} B_i \right)$ .

**Proof** (a): Since  $\text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-2} B_i \right) \neq \emptyset$ , Lemma 1(i) implies that the set of fixed points of operator  $T$  in (9, 10) (with  $B_{n-1} = 0$ ) is nonempty. The latter set coincides with the set of fixed points of operator  $\tilde{T}$  in (33, 34), so  $\text{Fix } \tilde{T} \neq \emptyset$ . Since  $\lambda \in (0, \frac{1}{2L})$  and  $\gamma \in (0, 1 - 2\lambda L)$ , Lemma 3 implies that  $(\mathbf{z}^k)$  is Fejér monotone with respect to  $\text{Fix } \tilde{T}$  and that  $\lim_{k \rightarrow +\infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$ . By nonexpansivity of resolvents,  $L$ -Lipschitz continuity of  $B_2, \dots, B_{n-1}$ , and boundedness of  $(\mathbf{z}^k)$ , it follows that  $(\mathbf{x}^k)$  is also bounded. Further, (33) and the fact that  $\lim_{k \rightarrow \infty} \|\mathbf{z}^{k+1} - \mathbf{z}^k\| = 0$  implies that

$$\lim_{k \rightarrow \infty} \|x_i^k - x_{i-1}^k\| = 0 \quad \forall i = 2, \dots, n. \quad (44)$$

Let  $\mathbf{u} = (u_1, \dots, u_{n-1}) \in \mathcal{H}^{n-1}$  be an arbitrary weak cluster point of  $(\mathbf{z}^k)$ . Then, due to (44), there exists a point  $x \in \mathcal{H}$  such that  $(\mathbf{u}, \mathbf{w})$  is a weak cluster point of  $(\mathbf{z}^k, \mathbf{x}^k)$ , where  $\mathbf{w} = (x, \dots, x) \in \mathcal{H}^n$ . Let  $S$  denote the maximally monotone operator defined by (22) when  $B_{n-1} = 0$ . Then (32b) implies

$$S \begin{pmatrix} z_1^k - x_1^k \\ (z_2^k - x_2^k) - (z_1^k - x_1^k) + \lambda b_2^k \\ (z_3^k - x_3^k) - (z_2^k - x_2^k) + \lambda b_3^k - \lambda b_2^k \\ \vdots \\ (z_{n-1}^k - x_{n-1}^k) - (z_{n-2}^k - x_{n-2}^k) + \lambda b_{n-1}^k - \lambda b_{n-2}^k \\ x_n^k \end{pmatrix} \ni \begin{pmatrix} x_1^k - x_n^k \\ x_2^k - x_n^k \\ x_3^k - x_n^k \\ \vdots \\ x_{n-1}^k - x_n^k \\ x_1^k - x_n^k \end{pmatrix}, \quad (45)$$

where  $b_i^k := B_{i-1}(x_i^k) - B_{i-1}(x_{i-1}^k)$ . Taking the limit along a subsequence of  $(\mathbf{z}^k, \mathbf{x}^k)$  which converges weakly to  $(\mathbf{u}, \mathbf{w})$  in (45), using demiclosedness of  $S$  together with  $L$ -Lipschitz continuity of  $B_2, \dots, B_{n-1}$ , and unravelling the resulting expression gives that  $\mathbf{u} \in \text{Fix } \tilde{T}$  and  $x = J_{\lambda A_1}(u_1) \in \text{zer} \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-2} B_i \right)$ . Thus, by [16, Theorem 5.5], it follows that  $(\mathbf{z}^k)$  converges weakly to a point  $\mathbf{z} \in \text{Fix } \tilde{T}$ .

(b): Follows by using an argument analogous to the one in Theorem 3(b).  $\square$

**Remark 9** (Exploiting cocoercivity) If a Lipschitz continuous operator  $B_i$  in (28) is actually cocoercive, then it is possible to reduce the number evaluations of  $B_i$  per iteration by combining the ideas in Sects. 3 and 4. In fact, we can consider the problem

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \left( \sum_{i=1}^n A_i + \sum_{i=1}^{n-1} B_i \right)(x),$$

where  $B_1, \dots, B_{n-2}$  are each either monotone and Lipschitz continuous or cocoercive, and  $B_{n-1}$  is cocoercive. For this problem, we can replace (34) in the definition of  $\tilde{T}$  with



$$\begin{cases} x_1 = J_{\lambda A_1}(z_1), \\ x_2 = J_{\lambda A_2}(z_2 + x_1 - z_1 - \lambda B_1(x_1)), \\ x_i = J_{\lambda A_i}(z_i + x_{i-1} - z_{i-1} - \lambda B_{i-1}(x_{i-1}) - \lambda b_{i-1}) \quad \forall i \in \llbracket 3, n-1 \rrbracket, \\ x_n = J_{\lambda A_n}(x_1 + x_{n-1} - z_{n-1} - \lambda B_{n-1}(x_{n-1}) - \lambda b_{n-1}), \end{cases}$$

where  $b_2, \dots, b_{n-1} \in \mathcal{H}$  are given by

$$b_i = \begin{cases} 0 & \text{if } B_{i-1} \text{ is cocoercive,} \\ B_{i-1}(x_i) - B_{i-1}(x_{i-1}) & \text{if } B_{i-1} \text{ is monotone and Lipschitz.} \end{cases}$$

This modification can be shown to converge using a proof similar to Theorem 4 for  $\lambda \in (0, \frac{1}{2L})$ . However, it is not straightforward to recover Theorem 3 as a special case of such a result because the stepsizes range of  $\lambda \in (0, \frac{2}{L})$  in the cocoercive only case (i.e., Theorem 3) are larger than the range in the mixed case. Moreover, Theorem 3(c) (strong convergence to dual solutions) does not have an analogue in the statement of Theorem 4. In addition, keeping the two cases separate allows the analysis to be as transparent as possible.

**Acknowledgements** FJAA will always be indebted to his mentor, Professor Asen L. Dontchev, for valuable career (and life) advice as well as for sparking his interest in the analysis of optimisation algorithms under Lipschitzian properties. The authors are thankful to the anonymous referees for their careful reading and for providing very helpful comments.

**Funding** Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. FJAA and DTB were partially supported by the Ministry of Science, Innovation and Universities of Spain and the European Regional Development Fund (ERDF) of the European Commission, grant PGC2018-097960-B-C22. FJAA was partially supported by the Generalitat Valenciana (AICO/2021/165). YM was supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The project number is 305286. MKT was supported in part by Australian Research Council grant DE200100063. DTB was supported by MINECO and European Social Fund (PRE2019-090751) under the program “Ayudas para contratos predoctorales para la formación de doctores” 2019.

**Data availability** We do not analyse or generate any datasets, because our work proceeds within a theoretical and mathematical approach.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Baillon, J.-B., Haddad, G.: Quelques propriétés des opérateurs angle-bornés et  $n$ -cycliquement monotones. *Isr. J. Math.* **26**(2), 137–150 (1977)
2. Rockafellar, R.T.: Monotone operators associated with saddle-functions and minimax problems. In: Browder, F.E. (ed.) *Nonlinear Functional Analysis Part Proceedings of Symposia in Pure Mathematics*, vol. 18, pp. 241–250. American Mathematical Soc, Providence (1970)
3. Tseng, P.: A modified forward-backward splitting method for maximal monotone mappings. *SIAM J. Control Optim.* **38**(2), 431–446 (2000). <https://doi.org/10.1137/S0363012998338806>
4. Davis, D., Yin, W.: A three-operator splitting scheme and its optimization applications. *Set-Valued Var. Anal.* **25**(4), 829–858 (2017). <https://doi.org/10.1007/s11228-017-0421-z>
5. Dao, M.N., Phan, H.M.: An adaptive splitting algorithm for the sum of three operators. *Fixed Point Theory Algorithms Sci. Eng.* (2021). <https://doi.org/10.1186/s13663-021-00701-8>
6. Aragón-Artacho, F.J., Torregrosa-Belén, D.: A direct proof of convergence of Davis–Yin splitting algorithm allowing larger stepsizes. *Set-Valued Var. Anal.* **30**, 1011–1029 (2022). <https://doi.org/10.1007/s11228-022-00631-6>
7. Rieger, J., Tam, M.K.: Backward-forward-reflected-backward splitting for three operator monotone inclusions. *Appl. Math. Comput.* **381**, 125248 (2020). <https://doi.org/10.1016/j.amc.2020.125248>
8. Raguet, H., Fadili, J., Peyré, G.: A generalized forward-backward splitting. *SIAM J. Imaging Sci.* **6**(3), 1199–1226 (2013). <https://doi.org/10.1137/120872802>
9. Johnstone, P.R., Eckstein, J.: Projective splitting with forward steps. *Math. Program.* (2020). <https://doi.org/10.1007/s10107-020-01565-3>
10. Johnstone, P.R., Eckstein, J.: Single-forward-step projective splitting: exploiting cocoercivity. *Comput. Optim. Appl.* **78**(1), 125–166 (2021). <https://doi.org/10.1007/s10589-020-00238-3>
11. Bertsekas, D., Tsitsiklis, J.: *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs (1989)
12. Condat, L., Malinovsky, G., Richtárik, P.: Distributed proximal splitting algorithms with rates and acceleration. *Front. Signal Process.* **1**, 776825 (2022). <https://doi.org/10.3389/frsip.2021.776825>
13. Malitsky, Y., Tam, M.K.: A forward-backward splitting method for monotone inclusions without cocoercivity. *SIAM J. Optim.* **26**(3), 1451–1472 (2020). <https://doi.org/10.1137/18M1207260>
14. Malitsky, Y., Tam, M.K.: Resolvent splitting for sums of monotone operators with minimal lifting (2021). <http://arxiv.org/abs/2108.02897>
15. Ryu, E.K.: Uniqueness of DRS as the 2 operator resolvent-splitting and impossibility of 3 operator resolvent-splitting. *Math. Program.* **182**(1), 233–273 (2020). <https://doi.org/10.1007/s10107-019-01403-1>
16. Bauschke, H.H., Combettes, P.L.: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd edn. Springer, Berlin (2017). <https://doi.org/10.1007/978-3-319-48311-5>
17. Minty, G.J.: Monotone (nonlinear) operators in Hilbert space. *Duke Math. J.* **29**, 341–346 (1962)
18. Aragón-Artacho, F.J., Boj, R.I., Torregrosa-Belén, D.: A primal-dual splitting algorithm for composite monotone inclusions with minimal lifting (2022). <http://arxiv.org/abs/2202.09665>
19. Bauschke, H.H., Singh, S., Wang, X.: The splitting algorithms by Ryu, by Malitsky-Tam, and by Campoy applied to normal cones of linear subspaces converge strongly to the projection onto the intersection (2022). <http://arxiv.org/abs/2203.03832>
20. Shi, W., Ling, Q., Wu, G., Yin, W.: EXTRA: an exact first-order algorithm for decentralized consensus optimization. *SIAM J. Optim.* **25**(2), 944–966 (2015). <https://doi.org/10.1137/14096668X>
21. Attouch, H., Théra, M.: A general duality principle for the sum of two operators. *J. Convex Anal.* **3**, 1–24 (1996)
22. Csetnek, E.R., Malitsky, Y., Tam, M.K.: Shadow Douglas-Rachford splitting for monotone inclusions. *Appl. Math. Comput.* **80**(3), 665–678 (2019). <https://doi.org/10.1007/s00245-019-09597-8>