Almost-Surely Convergent Randomly Activated Monotone Operator Splitting Methods*

Patrick L. Combettes[1] and Javier I. Madariaga[2]

[1]North Carolina State University
Department of Mathematics
Raleigh, NC 27695, USA
plc@math.ncsu.edu

[2]North Carolina State University
Department of Mathematics
Raleigh, NC 27695, USA
jimadari@ncsu.edu

Abstract. We propose stochastic splitting algorithms for solving large-scale composite inclusion problems involving monotone and linear operators. They activate at each iteration blocks of randomly selected resolvents of monotone operators and, unlike existing methods, achieve almost sure convergence of the iterates to a solution without any regularity assumptions or knowledge of the norms of the linear operators. Applications to image recovery and machine learning are provided.

Keywords. Convex optimization, data analysis, machine learning, monotone inclusion, signal recovery, splitting algorithm, stochastic algorithm.

*Contact author: P. L. Combettes. Email: plc@math.ncsu.edu. This work was supported by the National Science Foundation under grant CCF-2211123.

§1. Introduction

The problem of extracting information from data is at the core of many tasks in signal processing, inverse problems, and machine learning. A prevalent methodology to seek meaningful solutions is to build a mathematical model that incorporates the prior knowledge about the object of interest $\overline{x}$ and the data, which consist of observations mathematically or physically related to $\overline{x}$ (see Figure 1). Since the first mathematical formalizations of Euler [23] and Mayer [31] in the late 1740s, which contained the embryo of least-squares data fitting techniques, convex minimization formulations have been a tool of choice. The following problem encapsulates a broad range of minimization models found in data analysis problems [2, 4, 5, 7, 10, 12, 16, 25, 28, 39] (see Section 2.1 for notation).

Problem 1.1. H is a separable real Hilbert space and $f \in \Gamma_0(H)$. For every $k \in \{1, \ldots, p\}$, G_k is a separable real Hilbert space, $g_k \in \Gamma_0(G_k)$, and $0 \neq L_k \colon H \to G_k$ is linear and bounded. It is assumed that $\operatorname{zer}(\partial f + \sum_{k=1}^{p} L_k^* \circ (\partial g_k) \circ L_k) \neq \varnothing$. The task is to

$$\underset{x \in H}{\text{minimize}} \;\; f(x) + \sum_{k=1}^{p} g_k(L_k x). \tag{1.1}$$

In recent years, an increasing number of problem formulations have emerged, which cannot be naturally reduced to tractable minimization problems and which are best captured by more general notions of equilibria provided by inclusion problems [14, 15, 17, 18, 24, 27, 35, 41, 42]. A formulation covering such models, as well as Problem 1.1, is the following composite monotone inclusion formulation.

Problem 1.2. H is a separable real Hilbert space and $A \colon H \to 2^H$ is maximally monotone. For every $k \in \{1, \ldots, p\}$, G_k is a separable real Hilbert space, $B_k \colon G_k \to 2^{G_k}$ is maximally monotone, and $0 \neq L_k \colon H \to G_k$ is linear and bounded. It is assumed that $Z = \operatorname{zer}(A + \sum_{k=1}^{p} L_k^* \circ B_k \circ L_k) \neq \varnothing$. The task is to

$$\text{find} \;\; x \in H \;\; \text{such that} \;\; 0 \in Ax + \sum_{k=1}^{p} L_k^*\big(B_k(L_k x)\big). \tag{1.2}$$

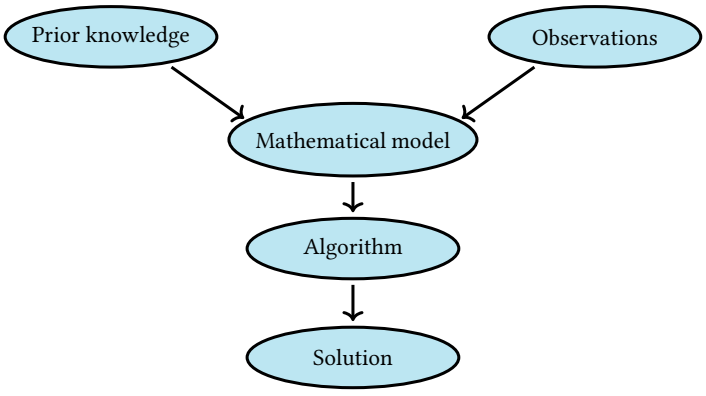


Figure 1: Data processing flowchart.

Splitting algorithms for solving Problem 1.2 operate on the principle that each nonlinear and linear operator is used separately over the course of the iterations. Since the nonlinear operators are general set-valued monotone operators, they must be activated through their resolvent. Various deterministic operator splitting methods are available to solve Problem 1.2, most of which require the activation of the resolvents of the $p + 1$ operators A and $(B_k)_{1\leqslant k\leqslant p}$ at each iteration [11]. Our specific focus is on solving Problem 1.2 in instances when p is large, as is often the case in data analysis problems. In such scenarios, memory and computing power limitations make the execution of standard monotone operator splitting algorithms inefficient if not simply impossible. We aim at designing monotone splitting algorithms which are stochastic in the sense that they activate a randomly selected block of operators at each iteration and, in addition, allow for random errors in the implementation of these resolvent steps. Furthermore, the proposed algorithms satisfy the following requirements:

- **R1:** They guarantee the almost sure convergence of the sequence of iterates to a solution to Problem 1.2 (respectively Problem 1.1) without any additional assumptions on the nonlinear operators (respectively the functions), the linear operators, or the underlying Hilbert spaces.
- **R2:** At each iteration, more than one randomly selected resolvent of the operators $(A, B_1, \ldots, B_p)$ can be activated.
- **R3:** Knowledge of bounds on the norms of the linear operators is not required.
- **R4:** The operators are available only through a stochastic approximation.

Requirement **R1** imposes actual iterate convergence to a solution and not a weaker form of convergence such as ergodic convergence, vanishing step sizes, or, in the context of Problem 1.1, convergence of the values of the objective function. It also asks that Problems 1.2 and 1.1 be addressed in their generality, without restricting their scope by introducing additional assumptions. Requirement **R2** makes it possible to activate more than one operator, hence opening the way to matching efficiently the computational load of an iteration to the possibly parallel architecture at hand. Requirement **R3** broadens the scope of the methods by not assuming any knowledge of the norms of the linear operators present in the model. For instance, in domain decomposition methods, it is quite difficult to obtain tight upper bounds on the norms of the trace operators [3]. Finally, in the spirit of the classical stochastic iteration models of [8, 22, 37], **R4** addresses the robustness of the algorithm to stochastic errors affecting the implementation of the operators.

As will be seen in the literature review of Section 2.2, there does not seem to exist methods that satisfy simultaneously **R1–R4**. Our main contribution is presented in Section 3, where we propose three algorithmic frameworks that comply with **R1–R4**. Section 4 is devoted to the minimization setting of Problem 1.1. The last section of the paper is Section 5, where the proposed algorithms are applied to signal restoration, support vector machine, classification, and image reconstruction problems.

§2. Notation and existing algorithms

2.1. Notation

Throughout, $\mathcal{H}$ is a separable real Hilbert space with power set $2^{\mathcal{H}}$, identity operator Id, scalar product $\langle \cdot \,|\, \cdot \rangle$, and associated norm $\|\cdot\|$.

Let $A\colon H \to 2^H$. The graph of A is $\operatorname{gra} A = \{(x, x^*) \in H \times H \mid x^* \in Ax\}$ and the set of zeros of A is $\operatorname{zer} A = \{x \in H \mid 0 \in Ax\}$. The inverse of A is the operator $A^{-1}\colon H \to 2^H$ with graph $\operatorname{gra} A^{-1} = \{(x^*, x) \in H \times H \mid x^* \in Ax\}$ and the resolvent of A is $J_A = (\mathrm{Id} + A)^{-1}$. Further, A is maximally monotone if

$$(\forall (x, x^*) \in H \times H) \quad \big[\, (x, x^*) \in \operatorname{gra} A \;\Leftrightarrow\; (\forall (y, y^*) \in \operatorname{gra} A)\ \langle x - y \mid x^* - y^* \rangle \geqslant 0 \,\big]. \tag{2.1}$$

An operator $F\colon H \to H$ is firmly nonexpansive if

$$(\forall x \in H)(\forall y \in H) \quad \langle x - y \mid Fx - Fy \rangle \geqslant \|Fx - Fy\|^2. \tag{2.2}$$

Lemma 2.1. *Let $F\colon H \to H$ be firmly nonexpansive and let $\gamma \in \left]0, +\infty\right[$. Then there exists a maximally monotone operator $A\colon H \to 2^H$ such that the following hold:*

(i) $F = J_A$.

(ii) $J_{\gamma F} = \mathrm{Id} - \gamma J_{(1+\gamma)^{-1}A} \circ (1+\gamma)^{-1}\mathrm{Id}$.

Proof. (i): See [6, Corollary 23.9].

(ii): This follows from (i) and [6, Proposition 23.29]. □

$\Gamma_0(H)$ denotes the class of lower semicontinuous convex functions $f\colon H \to \left]-\infty, +\infty\right]$ such that $\operatorname{dom} f = \{x \in H \mid f(x) < +\infty\} \neq \varnothing$. Let $f \in \Gamma_0(H)$. The subdifferential of f is the maximally monotone operator

$$\partial f\colon H \to 2^H\colon x \mapsto \{x^* \in H \mid (\forall z \in H)\ \langle z - x \mid x^* \rangle + f(x) \leqslant f(z)\} \tag{2.3}$$

and the proximity operator of f is

$$\operatorname{prox}_f = J_{\partial f}\colon H \to H\colon x \mapsto \operatorname*{argmin}_{z \in H} \Big(f(z) + \frac{1}{2}\|x - z\|^2 \Big). \tag{2.4}$$

Let C be a nonempty closed convex subset of H. Then ι_C denotes the indicator function of C, d_C the distance function to C,

$$N_C = \partial \iota_C\colon x \mapsto \begin{cases} \{x^* \in H \mid (\forall y \in C)\ \langle y - x \mid x^* \rangle \leqslant 0\}, & \text{if } x \in C; \\ \varnothing, & \text{otherwise} \end{cases} \tag{2.5}$$

the normal cone operator of C, and $\operatorname{proj}_C = \operatorname{prox}_{\iota_C} = J_{N_C}$ the projection operator onto C. In particular, if V is a closed vector subspace of H, then

$$N_V\colon H \to 2^H\colon x \mapsto \begin{cases} V^\perp, & \text{if } x \in V; \\ \varnothing, & \text{otherwise.} \end{cases} \tag{2.6}$$

The underlying probability space $(\Omega, \mathcal{F}, P)$ is assumed to be complete and $\mathcal{B}_H$ denotes the Borel σ-algebra of H. An H-valued random variable is a measurable mapping $x\colon (\Omega, \mathcal{F}) \to (H, \mathcal{B}_H)$. The σ-algebra generated by a family Φ of random variables is denoted by $\sigma(\Phi)$. Given $x\colon \Omega \to H$ and $S \subset H$, we set $[x \in S] = \{\omega \in \Omega \mid x(\omega) \in S\}$. The reader is referred to [6] for background on monotone operators and convex analysis, and to [29] for background on probability in Hilbert spaces.

We use sans-serif letters to denote deterministic variables and italicized serif letters to denote random variables. Finally, in connection with Problem 1.2, we define the Hilbert direct sum

$$\mathbf{G} = \mathsf{G}_1 \oplus \cdots \oplus \mathsf{G}_p, \tag{2.7}$$

as well as the subspace

$$\mathbf{W} = \left\{ \mathbf{x} \in \mathsf{H} \oplus \mathbf{G} \;\middle|\; (\forall k \in \{1, \dots, p\})\ \mathsf{x}_{k+1} = \mathsf{L}_k \mathsf{x}_1 \right\}, \tag{2.8}$$

and note that

$$\mathbf{W}^\perp = \left\{ \mathbf{x}^* \in \mathsf{H} \oplus \mathbf{G} \;\middle|\; \mathsf{x}_1^* = -\sum_{k=1}^{p} \mathsf{L}_k^* \mathsf{x}_{k+1}^* \right\}. \tag{2.9}$$

2.2. Existing algorithms

It seems that no algorithm satisfying requirements **R1–R4** has been explicitly proposed to solve Problem 1.2 — or even Problem 1.1 — in the literature. There is a vast body of papers on random activation algorithms in the special case of Problem 1.1 that consists in minimizing a sum of smooth functions $\sum_{k=1}^{p} g_k$ in $\mathsf{H} = \mathbb{R}^N$ via so-called stochastic gradient descent methods. Their principle is to activate a randomly selected gradient in $(\nabla g_k)_{1 \leqslant k \leqslant p}$ at each iteration; see [21] and its bibliography and [19, 40] for related work with random proximal activations for this type of problem. These methods focus on a very specific instance of Problem 1.1 and they do not satisfy **R1**–**R2**. The only random activation algorithm tailored to Problem 1.1 which guarantees almost sure convergence of the iterates without additional assumptions such as strong convexity is the following (see also [1] for a nonadaptive version).

Proposition 2.2 ([9, Theorem 2.1 and Algorithm 3.1]). *Consider the setting of Problem 1.1 and suppose that* $\mathsf{H} = \mathbb{R}^N$ *and that, for every* $k \in \{1, \dots, p\}$, $\mathsf{G}_k = \mathbb{R}^{M_k}$, *all considered as standard Euclidean spaces. Let* $(\pi_k)_{1 \leqslant k \leqslant p}$ *be real numbers in* $]0, 1]$ *such that* $\sum_{k=1}^{p} \pi_k = 1$, *and let* $(k_n)_{n \in \mathbb{N}}$ *be identically distributed* $\{1, \dots, p\}$*-valued random variables such that, for every* $k \in \{1, \dots, p\}$, $\mathsf{P}[k_0 = k] = \pi_k$. *For every* $k \in \{1, \dots, p\}$ *and every* $n \in \mathbb{N}$, *set* $\varepsilon_{k,n} = 1_{[k_n = k]}$. *Let* $\tau_0 \in]0, +\infty[$ *and* $\sigma_0 \in]0, +\infty[$ *be such that*

$$\tau_0 \sigma_0 \max_{1 \leqslant k \leqslant p} \frac{\|\mathsf{L}_k\|^2}{\pi_k} < 1. \tag{2.10}$$

Further, let $\chi_0 \in [0, 1[$, $\eta \in]0, 1[$, *and* $\delta \in]1, +\infty[$, *set* $\rho_0 = 0$ *and* $v_0 = 0$, *let* $x_{1,0}$ *be a* H*-valued random variable, and let* $\boldsymbol{y}_0$ *be a* $\mathbf{G}$*-valued random variable. Set* $\boldsymbol{z}_0 = \boldsymbol{y}_0$ *and* $\mathbf{L} \colon \mathsf{H} \to \mathbf{G} \colon \mathsf{x} \mapsto (\mathsf{L}_k \mathsf{x})_{1 \leqslant k \leqslant p}$,

and iterate

$$
\begin{array}{l}
\textit{for } n = 0, 1, \dots \\
\left|\begin{array}{l}
(\tau_{n+1}, \sigma_{n+1}, \chi_{n+1}) = \begin{cases} \Big(\dfrac{\tau_n}{1-\chi_n}, \sigma_n(1-\chi_n), \chi_n\eta\Big), & \textit{if } \rho_n > \|\mathbf{L}\|\nu_n\delta; \\ \Big(\tau_n(1-\chi_n), \dfrac{\sigma_n}{1-\chi_n}, \chi_n\eta\Big), & \textit{if } \rho_n < \|\mathbf{L}\|\nu_n\delta; \\ (\tau_n, \sigma_n, \chi_n), & \textit{if } \dfrac{\|\mathbf{L}\|\nu_n}{\delta} \leqslant \rho_n \leqslant \|\mathbf{L}\|\nu_n\delta \end{cases} \\
x_{1,n+1} = x_{1,n} + \operatorname{prox}_{\tau_{n+1}f}\big(x_{1,n} - \tau_{n+1}\sum_{k=1}^{p} L_k^* z_{k,n}\big) \\
\textit{for } k = 1, \dots, p \\
\left|\begin{array}{l}
y_{k,n+1} = y_{k,n+1} + \varepsilon_{k,n}\Big(\operatorname{prox}_{\sigma_{n+1}g_k^*}\big(y_{k,n} + \sigma_{n+1}L_k x_{1,n+1}\big) - y_{k,n}\Big) \\
z_{k,n+1} = y_{k,n} + \varepsilon_{k,n}\Big(y_{k,n+1} + \dfrac{1}{\pi_k}\big(y_{k,n+1} - y_{k,n}\big) - y_{k,n}\Big) \\
\rho_{n+1} = \Big\|\dfrac{1}{\tau_{n+1}}\big(x_n - x_{n+1}\big) - \dfrac{1}{\pi_{k_n}}L_{k_n}^*\big(y_{k_n,n} - y_{k_n,n+1}\big)\Big\|_1 \\
\nu_{n+1} = \dfrac{1}{\pi_{k_n}}\Big\|L_{k_n}\big(x_n - x_{n+1}\big) - \dfrac{1}{\sigma_{n+1}}\big(y_{k_n,n} - y_{k_n,n+1}\big)\Big\|_1,
\end{array}\right.
\end{array}\right.
\end{array}
\tag{2.11}
$$

where $\|\cdot\|_1$ *denotes the* ℓ^1*-norm. Then* $(x_{1,n})_{n\in\mathbb{N}}$ *converges* P-a.s. *to an* $\operatorname{Argmin}(f+\sum_{k=1}^{p} g_k\circ L_k)$*-valued random variable.*

Algorithm (2.11) is of interest because it guarantees **R1** in a finite-dimensional setting. However, it does not satisfy **R2** since, at each iteration, f must be activated together with one of the functions $(g_k)_{1\leqslant k\leqslant p}$. It does not satisfy **R3** either since it requires the knowledge of the norms of linear operators in (2.10). We also note that it does not tolerate errors in the evaluation of the proximity operators, which means that **R4** is not satisfied.

Let us now turn to the general Problem 1.2. The only algorithm that satisfies **R1** is that of [34], which corresponds to an implementation of the random block-coordinate forward-backward algorithm of [13, Section 5.2] suggested in [13, Remark 5.10(iv)].

Proposition 2.3 ([34, Proposition 4.6]). *Consider the setting of Problem 1.2. Let* $W\colon H \to H$ *and, for every* $k \in \{1, \dots, p\}$*, let* $U_k\colon G_k \to G_k$ *be bounded linear strongly positive self-adjoint operators such that*

$$
\sum_{k=1}^{p}\big\|U_k^{1/2}L_k W^{1/2}\big\|^2 < \frac{1}{2}. \tag{2.12}
$$

Let $(\lambda_n)_{n\in\mathbb{N}}$ *be a sequence in* $]0, 1]$ *such that* $\inf_{n\in\mathbb{N}}\lambda_n > 0$*, let* $x_{1,0}$ *and* $(a_{1,n})_{n\in\mathbb{N}}$ *be* H*-valued random variables, let* v_0 *and* $(b_n)_{n\in\mathbb{N}}$ *be* $\mathbf{G}$*-valued random variables, and let* $(\varepsilon_n)_{n\in\mathbb{N}}$ *be identically distributed* $\{0, 1\}^p \smallsetminus \{\mathbf{0}\}$*-valued random variables. Iterate*

$$
\begin{array}{l}
\textit{for } n = 0, 1, \dots \\
\left|\begin{array}{l}
y_{1,n} = J_{WA}\Big(x_{1,n} - W\big(\sum_{k=1}^{p} L_k^* v_{k,n}\big)\Big) + a_{1,n} \\
x_{1,n+1} = x_{1,n} + \lambda_n\big(y_{1,n} - x_{1,n}\big) \\
\textit{for } k = 1, \dots, p \\
\left|\begin{array}{l}
u_{k,n} = \varepsilon_{k,n}\Big(J_{U_k B_k^{-1}}\Big(v_{k,n} + U_k\big(L_k(2y_{1,n} - x_{1,n})\big)\Big) + b_{k,n}\Big) \\
v_{k,n+1} = v_{k,n} + \varepsilon_{k,n}\lambda_n\big(u_{k,n} - v_{k,n}\big),
\end{array}\right.
\end{array}\right.
\end{array}
\tag{2.13}
$$

and set $(\forall n \in \mathbb{N})\ \mathcal{E}_n = \sigma(\boldsymbol{\varepsilon}_n)$ *and* $\mathcal{X}_n = \sigma(x_{1,l}, \boldsymbol{v}_l)_{0\leqslant l\leqslant n}$. *In addition, assume that the following hold:*

(i) $\sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}(\|a_{1,n}\|^2 \,|\, \mathcal{X}_n)} < +\infty$ *and* $\sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}(\|\boldsymbol{b}_n\|^2 \,|\, \mathcal{X}_n)} < +\infty$.

(ii) *For every* $n \in \mathbb{N}$, $\mathcal{E}_n$ *and* $\mathcal{X}_n$ *are independent.*

(iii) *For every* $l \in \{1, \ldots, p\}$, $\mathsf{P}[\varepsilon_{l,0} = 1] > 0$.

Then $(x_{1,n})_{n\in\mathbb{N}}$ *converges weakly* P-a.s. *to a* Z*-valued random variable.*

Let us note that algorithm (2.13) satisfies **R1** but not **R2** since it must activate A at each iteration, nor does it satisfy **R3** since it requires the knowledge of the norms of linear operators to implement (2.12). Another framework related to Problem 1.2 is that of [20], which allows for random activations in Problem 1.2 in a finite-dimensional setting when no linear operator is present and under the assumption that the operators $(B_k)_{1\leqslant k\leqslant p}$ are cocoercive. It therefore does not satisfy several requirements of **R1** and activates only one operator per iteration, which violates **R2**. On the other hand, the recent work [38] solves Problem 1.2 in a finite-dimensional setting when no linear operator is present and under strong monotonicity of the nonlinear operators. Hence, it does not satisfy **R1** and, since it does not allow for multiple activations at each iteration, it does not satisfy **R2** either.

§3. Proposed algorithms

3.1. Multivariate framework

Our strategy consists in embedding Problem 1.2 into multivariate problems that have the following general form studied in [13] and involve m agents $(x_1, \ldots, x_m)$.

Problem 3.1. Let $(\mathbf{X}_i)_{1\leqslant i\leqslant m}$ and $(\mathbf{Y}_j)_{1\leqslant j\leqslant r}$ be families of separable real Hilbert spaces with Hilbert direct sums $\mathbf{X} = \mathbf{X}_1 \oplus \cdots \oplus \mathbf{X}_m$ and $\mathbf{Y} = \mathbf{Y}_1 \oplus \cdots \oplus \mathbf{Y}_r$. For every $i \in \{1, \ldots, m\}$ and every $j \in \{1, \ldots, r\}$, let $C_i \colon \mathbf{X}_i \to 2^{\mathbf{X}_i}$ and $D_j \colon \mathbf{Y}_j \to 2^{\mathbf{Y}_j}$ be maximally monotone, and let $M_{ji} \colon \mathbf{X}_i \to \mathbf{Y}_j$ be linear and bounded. Set

$$\begin{cases} \boldsymbol{M} \colon \mathbf{X} \to \mathbf{Y} \colon \mathbf{x} \mapsto \big(\sum_{i=1}^m M_{1i}x_i, \ldots, \sum_{i=1}^m M_{ri}x_i\big) \\ \boldsymbol{C} \colon \mathbf{X} \to 2^{\mathbf{X}} \colon \mathbf{x} \mapsto C_1x_1 \times \cdots \times C_mx_m \\ \boldsymbol{D} \colon \mathbf{Y} \to 2^{\mathbf{Y}} \colon \mathbf{y} \mapsto D_1y_1 \times \cdots \times D_ry_r. \end{cases} \tag{3.1}$$

The task is to

$$\text{find}\ \ \mathbf{x} \in \mathbf{X}\ \ \text{such that}\ \ \mathbf{0} \in \boldsymbol{C}\mathbf{x} + \boldsymbol{M}^*\big(\boldsymbol{D}(\boldsymbol{M}\mathbf{x})\big). \tag{3.2}$$

The set of solutions to (3.2) is denoted by $\mathbf{Z}$ and assumed to be nonempty. Further, the projection operator onto the subspace

$$\mathbf{V} = \big\{(\mathbf{x}, \mathbf{y}) \in \mathbf{X} \oplus \mathbf{Y} \mid \mathbf{y} = \boldsymbol{M}\mathbf{x}\big\} \tag{3.3}$$

is decomposed as

$$\mathrm{proj}_{\mathbf{V}} \colon (\mathbf{x}, \mathbf{y}) \mapsto \big(Q_l(\mathbf{x}, \mathbf{y})\big)_{1\leqslant l\leqslant m+r}, \quad \text{where} \quad \begin{cases} (\forall i \in \{1, \ldots, m\})\ \ Q_i \colon \mathbf{X} \oplus \mathbf{Y} \to \mathbf{X}_i \\ (\forall j \in \{1, \ldots, r\})\ \ Q_{m+j} \colon \mathbf{X} \oplus \mathbf{Y} \to \mathbf{Y}_j. \end{cases} \tag{3.4}$$

Our approach is ultimately based on the Douglas–Rachford algorithm implemented in $\mathbf{X} \oplus \mathbf{Y}$. Define

$$\mathbf{A} \colon \mathbf{X} \oplus \mathbf{Y} \to 2^{\mathbf{X} \oplus \mathbf{Y}} \colon (\mathbf{x}, \mathbf{y}) \mapsto \mathbf{C}\mathbf{x} \times \mathbf{D}\mathbf{y} \quad \text{and} \quad \mathbf{B} = \mathbf{N}_{\mathbf{V}}. \tag{3.5}$$

Then it follows from [13, Eq. (5.23)] that $(\mathbf{x}, \mathbf{y}) \in \operatorname{zer}(\mathbf{A}+\mathbf{B})$ if and only if $\mathbf{x} \in \operatorname{zer}(\mathbf{C}+\mathbf{M}^* \circ \mathbf{D} \circ \mathbf{M})$ and $\mathbf{y} = \mathbf{M}\mathbf{x}$. We can construct a point in $\operatorname{zer}(\mathbf{A} + \mathbf{B})$ iteratively by the Douglas–Rachford algorithm [6, Section 26.3], which requires the resolvents of $\mathbf{A}$ and $\mathbf{B}$. By [6, Proposition 23.18], $\mathbf{J}_{\mathbf{A}}$ can be decomposed in terms of $(\mathrm{J}_{\mathrm{C}_1}, \ldots, \mathrm{J}_{\mathrm{C}_m}, \mathrm{J}_{\mathrm{D}_1}, \ldots, \mathrm{J}_{\mathrm{D}_r})$. On the other hand, $\mathbf{J}_{\mathbf{B}} = \operatorname{proj}_{\mathbf{V}}$ and it follows from (3.3) and [6, Example 29.19(ii)] that

$$(\forall \mathbf{x} \in \mathbf{X})(\forall \mathbf{y} \in \mathbf{Y}) \quad \operatorname{proj}_{\mathbf{V}}(\mathbf{x}, \mathbf{y}) = (\mathbf{p}, \mathbf{M}\mathbf{p}), \quad \text{where} \quad \mathbf{p} = (\mathbf{Id} + \mathbf{M}^* \circ \mathbf{M})^{-1}\big(\mathbf{x} + \mathbf{M}^*\mathbf{y}\big). \tag{3.6}$$

This operator is decomposed in terms of the operators $(\mathrm{Q}_{\mathrm{l}})_{1 \leqslant \mathrm{l} \leqslant \mathrm{m}+\mathrm{r}}$ in (3.4). The following result provides a randomly block-activated implementation of this product space version of the Douglas–Rachford algorithm.

Theorem 3.2 ([13, Corollary 5.3]). *Consider the setting of Problem 3.1. Set* $\mathrm{O} = \{0, 1\}^{\mathrm{m}+\mathrm{r}} \smallsetminus \{\mathbf{0}\}$, *let* $\gamma \in \left]0, +\infty\right[$, *let* $(\lambda_{\mathrm{n}})_{\mathrm{n}\in\mathbb{N}}$ *be a sequence in* $\left]0, 2\right[$ *such that* $\inf_{\mathrm{n}\in\mathbb{N}} \lambda_{\mathrm{n}} > 0$ *and* $\sup_{\mathrm{n}\in\mathbb{N}} \lambda_{\mathrm{n}} < 2$, *let* $\boldsymbol{x}_0$, $\boldsymbol{z}_0$, $(\boldsymbol{a}_{\mathrm{n}})_{\mathrm{n}\in\mathbb{N}}$, *and* $(\boldsymbol{c}_{\mathrm{n}})_{\mathrm{n}\in\mathbb{N}}$ *be* $\mathbf{X}$*-valued random variables, let* $\boldsymbol{y}_0$, $\boldsymbol{w}_0$, $(\boldsymbol{b}_{\mathrm{n}})_{\mathrm{n}\in\mathbb{N}}$, *and* $(\boldsymbol{d}_{\mathrm{n}})_{\mathrm{n}\in\mathbb{N}}$ *be* $\mathbf{Y}$*-valued random variables, and let* $(\boldsymbol{\varepsilon}_{\mathrm{n}})_{\mathrm{n}\in\mathbb{N}}$ *be identically distributed* O*-valued random variables. Iterate*

$$\begin{array}{l}
\textit{for } \mathrm{n} = 0, 1, \ldots \\
\quad \textit{for } \mathrm{i} = 1, \ldots, \mathrm{m} \\
\quad\quad x_{\mathrm{i},\mathrm{n}+1} = x_{\mathrm{i},\mathrm{n}} + \varepsilon_{\mathrm{i},\mathrm{n}}\big(\mathrm{Q}_{\mathrm{i}}(\boldsymbol{z}_{\mathrm{n}}, \boldsymbol{w}_{\mathrm{n}}) + a_{\mathrm{i},\mathrm{n}} - x_{\mathrm{i},\mathrm{n}}\big) \\
\quad\quad z_{\mathrm{i},\mathrm{n}+1} = z_{\mathrm{i},\mathrm{n}} + \varepsilon_{\mathrm{i},\mathrm{n}}\lambda_{\mathrm{n}}\big(\mathrm{J}_{\gamma \mathrm{C}_{\mathrm{i}}}(2x_{\mathrm{i},\mathrm{n}+1} - z_{\mathrm{i},\mathrm{n}}) + c_{\mathrm{i},\mathrm{n}} - x_{\mathrm{i},\mathrm{n}+1}\big) \\
\quad \textit{for } \mathrm{j} = 1, \ldots, \mathrm{r} \\
\quad\quad y_{\mathrm{j},\mathrm{n}+1} = y_{\mathrm{j},\mathrm{n}} + \varepsilon_{\mathrm{m}+\mathrm{j},\mathrm{n}}\big(\mathrm{Q}_{\mathrm{m}+\mathrm{j}}(\boldsymbol{z}_{\mathrm{n}}, \boldsymbol{w}_{\mathrm{n}}) + b_{\mathrm{j},\mathrm{n}} - y_{\mathrm{j},\mathrm{n}}\big) \\
\quad\quad w_{\mathrm{j},\mathrm{n}+1} = w_{\mathrm{j},\mathrm{n}} + \varepsilon_{\mathrm{m}+\mathrm{j},\mathrm{n}}\lambda_{\mathrm{n}}\big(\mathrm{J}_{\gamma \mathrm{D}_{\mathrm{j}}}(2y_{\mathrm{j},\mathrm{n}+1} - w_{\mathrm{j},\mathrm{n}}) + d_{\mathrm{j},\mathrm{n}} - y_{\mathrm{j},\mathrm{n}+1}\big),
\end{array} \tag{3.7}$$

and set $(\forall \mathrm{n} \in \mathbb{N})$ $\mathcal{E}_{\mathrm{n}} = \sigma(\boldsymbol{\varepsilon}_{\mathrm{n}})$ *and* $\mathcal{S}_{\mathrm{n}} = \sigma(\boldsymbol{z}_{\mathrm{l}}, \boldsymbol{w}_{\mathrm{l}})_{0 \leqslant \mathrm{l} \leqslant \mathrm{n}}$. *In addition, assume that the following are satisfied:*

(i) $\sum_{\mathrm{n}\in\mathbb{N}} \sqrt{\mathrm{E}(\|\boldsymbol{a}_{\mathrm{n}}\|^2 \,|\, \mathcal{S}_{\mathrm{n}})} < +\infty$, $\sum_{\mathrm{n}\in\mathbb{N}} \sqrt{\mathrm{E}(\|\boldsymbol{b}_{\mathrm{n}}\|^2 \,|\, \mathcal{S}_{\mathrm{n}})} < +\infty$, $\sum_{\mathrm{n}\in\mathbb{N}} \sqrt{\mathrm{E}(\|\boldsymbol{c}_{\mathrm{n}}\|^2 \,|\, \mathcal{S}_{\mathrm{n}})} < +\infty$, $\sum_{\mathrm{n}\in\mathbb{N}} \sqrt{\mathrm{E}(\|\boldsymbol{d}_{\mathrm{n}}\|^2 \,|\, \mathcal{S}_{\mathrm{n}})} < +\infty$, $\boldsymbol{a}_{\mathrm{n}} \rightharpoonup \mathbf{0}$ P-a.s., *and* $\boldsymbol{b}_{\mathrm{n}} \rightharpoonup \mathbf{0}$ P-a.s.

(ii) *For every* $\mathrm{n} \in \mathbb{N}$, $\mathcal{E}_{\mathrm{n}}$ *and* $\mathcal{S}_{\mathrm{n}}$ *are independent.*

(iii) *For every* $\mathrm{l} \in \{1, \ldots, \mathrm{m} + \mathrm{r}\}$, $\mathrm{P}[\varepsilon_{\mathrm{l},0} = 1] > 0$.

Then $(\boldsymbol{x}_{\mathrm{n}})_{\mathrm{n}\in\mathbb{N}}$ *converges weakly* P-a.s. *to a* $\mathbf{Z}$*-valued random variable.*

Remark 3.3. The measurability of the weak limit in [13, Corollary 5.3] relies on [13, Proposition 2.3], which involves Pettis' theorem [36, Corollary 1.13]. The applicability of the latter follows from the separability of H and the fact that $(\Omega, \mathcal{F}, \mathrm{P})$ is a complete probability space; see [26, Sections 1.1a–b] for details.

Remark 3.4. At iteration n, the random variables $(\varepsilon_{\mathrm{i},\mathrm{n}})_{1 \leqslant \mathrm{i} \leqslant \mathrm{m}}$ and $(\varepsilon_{\mathrm{m}+\mathrm{j},\mathrm{n}})_{1 \leqslant \mathrm{j} \leqslant \mathrm{r}}$ act as switches which control which components are updated, while the random variables $a_{\mathrm{i},\mathrm{n}}$, $b_{\mathrm{j},\mathrm{n}}$, $c_{\mathrm{i},\mathrm{n}}$, and $d_{\mathrm{j},\mathrm{n}}$ model approximations in the implementation of the operators Q_{i}, Q_{j}, $\mathrm{J}_{\gamma \mathrm{C}_{\mathrm{i}}}$, and $\mathrm{J}_{\gamma \mathrm{D}_{\mathrm{j}}}$, respectively.

We now present three frameworks for solving Problem 1.2 which are based on specializations of Theorem 3.2.

3.2. Framework 1

The first approach stems from the observation that Problem 3.1 reduces to Problem 1.2 when $m = 1$, $r = p$, $X_1 = H$, $C_1 = A$, and $(\forall k \in \{1, \dots, p\})$ $Y_k = G_k$, $M_{k,1} = L_k$, and $D_k = B_k$. Surprisingly, this basic observation does not seem to have been exploited in attempts to design random block activation algorithms for solving Problem 1.1 or Problem 1.2 (or special cases thereof) using the stochastic quasi-Fejér framework of [13]; see for instance [9, 30, 32, 33].

We derive from Theorem 3.2 the following convergence result.

Proposition 3.5. *Consider the setting of Problem 1.2. Set* $O = \{0,1\}^{1+p} \smallsetminus \{\mathbf{0}\}$, *let* $\gamma \in]0, +\infty[$, *let* $(\lambda_n)_{n\in\mathbb{N}}$ *be a sequence in* $]0,2[$ *such that* $\inf_{n\in\mathbb{N}} \lambda_n > 0$ *and* $\sup_{n\in\mathbb{N}} \lambda_n < 2$, *let* $x_{1,0}$, $z_{1,0}$, $(c_{1,n})_{n\in\mathbb{N}}$, *and* $(e_n)_{n\in\mathbb{N}}$ *be* H*-valued random variables, let* $\boldsymbol{y}_0$, $\boldsymbol{w}_0$, *and* $(\boldsymbol{d}_n)_{n\in\mathbb{N}}$ *be* $\mathbf{G}$*-valued random variables, and let* $(\boldsymbol{\varepsilon}_n)_{n\in\mathbb{N}}$ *be identically distributed* O*-valued random variables. Set* $Q = (\mathrm{Id} + \sum_{k=1}^{p} L_k^* \circ L_k)^{-1}$ *and iterate*

$$
\begin{array}{l}
\text{for } n = 0, 1, \dots \\
\left\lfloor \begin{array}{l}
s_n = Q\big(z_{1,n} + \sum_{k=1}^{p} L_k^* w_{k,n}\big) + e_n \\
x_{1,n+1} = x_{1,n} + \varepsilon_{1,n}(s_n - x_{1,n}) \\
z_{1,n+1} = z_{1,n} + \varepsilon_{1,n}\lambda_n\big(J_{\gamma A}(2x_{1,n+1} - z_{1,n}) + c_{1,n} - x_{1,n+1}\big) \\
\text{for } k = 1, \dots, p \\
\left\lfloor \begin{array}{l}
y_{k,n+1} = y_{k,n} + \varepsilon_{1+k,n}(L_k s_n - y_{k,n}) \\
w_{k,n+1} = w_{k,n} + \varepsilon_{1+k,n}\lambda_n\big(J_{\gamma B_k}(2y_{k,n+1} - w_{k,n}) + d_{k,n} - y_{k,n+1}\big).
\end{array} \right.
\end{array} \right.
\end{array}
\tag{3.8}
$$

In addition, assume that the following are satisfied:

(i) $\sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}(\|c_{1,n}\|^2 \,|\, \sigma(z_{1,l}, \boldsymbol{v}_l)_{0\leqslant l\leqslant n})} < +\infty$, $\sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}(\|\boldsymbol{d}_n\|^2 \,|\, \sigma(z_{1,l}, \boldsymbol{v}_l)_{0\leqslant l\leqslant n})} < +\infty$, $\sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}(\|e_n\|^2 \,|\, \sigma(z_{1,l}, \boldsymbol{v}_l)_{0\leqslant l\leqslant n})} < +\infty$, *and* $e_n \rightharpoonup 0$.

(ii) *For every* $n \in \mathbb{N}$, $\sigma(\boldsymbol{\varepsilon}_n)$ *and* $\sigma(z_{1,l}, \boldsymbol{w}_l)_{0\leqslant l\leqslant n}$ *are independent.*

(iii) *For every* $l \in \{1, \dots, p+1\}$, $\mathsf{P}[\varepsilon_{l,0} = 1] > 0$.

Then $(x_{1,n})_{n\in\mathbb{N}}$ *converges weakly* P-a.s. *to a* Z*-valued random variable.*

Proof. In Problem 3.1, set $m = 1$, $r = p$, $X_1 = H$, $C_1 = A$, and, for every $k \in \{1, \dots, p\}$, $Y_k = G_k$, $M_{k,1} = L_k$, and $D_k = B_k$. Further, for every $n \in \mathbb{N}$, set $a_{1,n} = e_n$ and, for every $k \in \{1, \dots, p\}$, set $b_{k,n} = L_k e_n$. Then it follows from (i) that $a_{1,n} \rightharpoonup 0$ P-a.s., $\boldsymbol{b}_n \rightharpoonup \mathbf{0}$ P-a.s., and

$$
\begin{aligned}
\sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}(\|\boldsymbol{b}_n\|^2 \,|\, \sigma(z_l, \boldsymbol{v}_l)_{0\leqslant l\leqslant n})} &\leqslant \sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}\left(\left(\sum_{k=1}^{p} \|L_k\|^2\right)\|e_n\|^2 \,\middle|\, \sigma(z_l, \boldsymbol{v}_l)_{0\leqslant l\leqslant n}\right)} \\
&= \sqrt{\sum_{k=1}^{p} \|L_k\|^2} \sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}\big(\|e_n\|^2 \,\big|\, \sigma(z_l, \boldsymbol{v}_l)_{0\leqslant l\leqslant n}\big)} \\
&< +\infty.
\end{aligned}
\tag{3.9}
$$

The assertion therefore results from Theorem 3.2. □

3.3. Framework 2

In Framework 1, Problem 3.1 collapses to Problem 1.2 by reducing the number of agents to $m = 1$. Here, we use $m = p + 1$ agents in Problem 3.1 and capture Problem 1.2 by forcing these agents $(x_1, \dots, x_{p+1})$ to lie in the subspace $\mathbf{W}$ of (2.8).

Proposition 3.6. *Consider the setting of Problem 1.2. Set* $O = \{0,1\}^{p+2} \smallsetminus \{\mathbf{0}\}$, *let* $\gamma \in \left]0, +\infty\right[$, *let* $(\lambda_n)_{n\in\mathbb{N}}$ *be a sequence in* $\left]0,2\right[$ *such that* $\inf_{n\in\mathbb{N}} \lambda_n > 0$ *and* $\sup_{n\in\mathbb{N}} \lambda_n < 2$, *let* $\boldsymbol{x}_0$, $\boldsymbol{z}_0$, $\boldsymbol{u}_0$, $\boldsymbol{v}_0$, *and* $(\boldsymbol{c}_n)_{n\in\mathbb{N}}$ *be* $H \oplus \mathbf{G}$*-valued random variables, let* $(e_n)_{n\in\mathbb{N}}$ *be* H*-valued random variables, and let* $(\boldsymbol{\varepsilon}_n)_{n\in\mathbb{N}}$ *be identically distributed* O*-valued random variables. Set* $Q = (\mathrm{Id} + \sum_{k=1}^{p} L_k^* \circ L_k)^{-1}$. *Iterate*

$$
\begin{array}{l}
\text{for } n = 0, 1, \dots \\
\quad \text{for } i = 1, \dots, p+1 \\
\quad\quad x_{i,n+1} = x_{i,n} + \varepsilon_{i,n}\Big(\dfrac{z_{i,n} + v_{i,n}}{2} - x_{i,n}\Big) \\
\quad z_{1,n+1} = z_{1,n} + \varepsilon_{1,n}\lambda_n\big(J_{\gamma A}(2x_{1,n+1} - z_{1,n}) + c_{1,n} - x_{1,n+1}\big) \\
\quad \text{for } k = 1, \dots, p \\
\quad\quad z_{k+1,n+1} = z_{k+1,n} + \varepsilon_{k+1,n}\lambda_n\big(J_{\gamma B_k}(2x_{k+1,n+1} - z_{k+1,n}) + c_{k+1,n} - x_{k+1,n+1}\big) \\
\quad \text{for } k = 1, \dots, p+1 \\
\quad\quad u_{k,n+1} = u_{k,n} + \varepsilon_{p+2,n}\Big(\dfrac{z_{k,n} + v_{k,n}}{2} - u_{k,n}\Big) \\
\quad s_n = \varepsilon_{p+2,n}\Big(Q\big(2u_{1,n+1} - v_{1,n} + \sum_{k=1}^{p} L_k^*(2u_{k+1,n+1} - v_{k+1,n})\big) + e_n\Big) \\
\quad v_{1,n+1} = v_{1,n} + \varepsilon_{p+2,n}\lambda_n(s_n - u_{1,n+1}) \\
\quad \text{for } k = 1, \dots, p \\
\quad\quad v_{k+1,n+1} = v_{k+1,n} + \varepsilon_{p+2,n}\lambda_n(L_k s_n - u_{k+1,n+1}).
\end{array}
\tag{3.10}
$$

In addition, assume that the following are satisfied:

(i) $\sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}(\|\boldsymbol{c}_n\|^2 \,|\, \sigma(\boldsymbol{z}_l, \boldsymbol{v}_l)_{0\leqslant l\leqslant n})} < +\infty$ *and* $\sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}(\|e_n\|^2 \,|\, \sigma(\boldsymbol{z}_l, \boldsymbol{v}_l)_{0\leqslant l\leqslant n})} < +\infty$.

(ii) *For every* $n \in \mathbb{N}$, $\sigma(\boldsymbol{\varepsilon}_n)$ *and* $\sigma(\boldsymbol{z}_l, \boldsymbol{v}_l)_{0\leqslant l\leqslant n}$ *are independent.*

(iii) *For every* $l \in \{1, \dots, p+2\}$, $\mathsf{P}[\varepsilon_{l,0} = 1] > 0$.

Then $(x_{1,n})_{n\in\mathbb{N}}$ *converges weakly* P-a.s. *to a* Z*-valued random variable.*

Proof. In Problem 3.1, set $m = p+1$, $r = 1$, $X_1 = H$, $(X_i)_{2\leqslant i\leqslant m} = (G_{i-1})_{2\leqslant i\leqslant m}$, and $Y_1 = H \oplus \mathbf{G}$. Thus, $\mathbf{Y} = Y_1 = H \oplus \mathbf{G} = \mathbf{X}$. Moreover, for every $i \in \{1, \dots, p+1\}$, set

$$
M_{1i} \colon X_i \to H \oplus \mathbf{G} \colon x_i \mapsto (0, \dots, 0, \underbrace{x_i}_{\text{ith position}}, 0, \dots, 0),
\tag{3.11}
$$

which yields

$$
M_{1i}^* \colon H \oplus \mathbf{G} \to X_i \colon (x_1^*, \dots, x_{p+1}^*) \mapsto x_i^*.
\tag{3.12}
$$

Further, denote by $\mathbf{x} = (x_1, \dots, x_{p+1})$ a generic element in $H \oplus \mathbf{G}$ and define $D_1 = N_{\mathbf{W}}$, where $\mathbf{W}$ is the subspace of (2.8). In this configuration, (3.2) reduces to

$$
\text{find } \ \mathbf{x} \in H \oplus \mathbf{G} \ \text{ such that } \ \mathbf{0} \in \bigtimes_{i=1}^{p+1} C_i x_i + N_{\mathbf{W}}\mathbf{x}.
\tag{3.13}
$$

We observe that

$$(\forall i \in \{1,\dots,p+1\})(\forall l \in \{1,\dots,p+1\}) \quad M_{1i}^* \circ M_{1l} = \begin{cases} \mathrm{Id}, & \text{if } i = l; \\ 0, & \text{if } i \neq l. \end{cases} \tag{3.14}$$

As a result, $(\mathbf{Id} + \mathbf{M}^* \circ \mathbf{M})^{-1} = (1/2)\mathbf{Id}$ and we derive from (3.4), (3.6), and (3.12) that

$$Q_{p+2} \colon (z, v) \mapsto \frac{z+v}{2} \quad \text{and} \quad (\forall l \in \{1,\dots,p+1\}) \quad Q_l \colon (z, v) \mapsto \frac{z_l + v_l}{2}. \tag{3.15}$$

Altogether, (3.7) with variables $y_{1,n} = \boldsymbol{u}_n \in H \oplus \mathbf{G}$ and $w_{1,n} = \boldsymbol{v}_n \in H \oplus \mathbf{G}$ becomes

$$\begin{array}{l} \text{for } n = 0, 1, \dots \\ \quad \text{for } i = 1, \dots, p+1 \\ \quad\quad \left\lfloor \begin{array}{l} x_{i,n+1} = x_{i,n} + \varepsilon_{i,n}\Big(\dfrac{z_{i,n} + v_{i,n}}{2} - x_{i,n}\Big) \\ z_{i,n+1} = z_{i,n} + \varepsilon_{i,n}\lambda_n\big(J_{\gamma C_i}(2x_{i,n+1} - z_{i,n}) + c_{i,n} - x_{i,n+1}\big) \end{array} \right. \\ \quad \boldsymbol{u}_{n+1} = \boldsymbol{u}_n + \varepsilon_{p+2,n}\Big(\dfrac{\boldsymbol{z}_n + \boldsymbol{v}_n}{2} - \boldsymbol{u}_n\Big) \\ \quad \boldsymbol{v}_{n+1} = \boldsymbol{v}_n + \varepsilon_{p+2,n}\lambda_n\big(\mathrm{proj}_{\mathbf{W}}(2\boldsymbol{u}_{n+1} - \boldsymbol{v}_n) + \boldsymbol{d}_{1,n} - \boldsymbol{u}_{n+1}\big), \end{array} \tag{3.16}$$

where $\boldsymbol{d}_{1,n}$ is the error incurred when projecting onto $\mathbf{W}$ at iteration n. We derive from (2.8) and [6, Example 29.19(ii)] that

$$\mathrm{proj}_{\mathbf{W}} \colon (x, y_1, \dots, y_p) \mapsto (s, L_1 s, \dots, L_p s), \quad \text{where} \quad s = \Big(\mathrm{Id} + \sum_{k=1}^{p} L_k^* \circ L_k\Big)^{-1}\Big(x + \sum_{k=1}^{p} L_k^* y_k\Big). \tag{3.17}$$

Set $(\forall n \in \mathbb{N})$ $\boldsymbol{d}_{1,n} = (e_n, L_1 e_n, \dots, L_p e_n)$. Then we infer from (i) that

$$\begin{aligned} \sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}\big(\|\boldsymbol{d}_{1,n}\|^2 \,\big|\, \sigma(z_l, v_l)_{0\leqslant l\leqslant n}\big)} &\leqslant \sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}\bigg(\Big(1 + \sum_{k=1}^{p} \|L_k\|^2\Big)\|e_n\|^2 \,\bigg|\, \sigma(z_l, v_l)_{0\leqslant l\leqslant n}\bigg)} \\ &= \sqrt{1 + \sum_{k=1}^{p} \|L_k\|^2} \sum_{n\in\mathbb{N}} \sqrt{\mathsf{E}\big(\|e_n\|^2 \,\big|\, \sigma(z_l, v_l)_{0\leqslant l\leqslant n}\big)} \\ &< +\infty. \end{aligned} \tag{3.18}$$

Thus, it follows from Theorem 3.2 that, with $\mathbf{Z}$ denoting the set of solutions to (3.13),

$$\text{the sequence } \big(x_{1,n}, x_{2,n}, \dots, x_{p+1,n}\big)_{n\in\mathbb{N}} \text{ in (3.16) converges weakly P-a.s.} \\ \text{to a } \mathbf{Z}\text{-valued random variable } \overline{\boldsymbol{x}} = \big(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_{p+1}\big) \text{ if } \mathbf{Z} \neq \varnothing. \tag{3.19}$$

Next, we specialize (3.13) to

$$C_1 = A \quad \text{and} \quad (\forall i \in \{2,\dots,p+1\}) \quad C_i = B_{i-1}. \tag{3.20}$$

In this context, (3.16) reduces to (3.10). Recalling that Z denotes the set of solutions to Problem 1.2, in view of (3.19), it remains to show that

$$\mathbf{Z} = \big\{(x_1, L_1 x_1, \dots, L_p x_1) \mid x_1 \in Z\big\}. \tag{3.21}$$

Let $\mathbf{x} \in \mathrm{H} \oplus \mathbf{G}$. We have

$$
\begin{aligned}
\mathbf{x} \in \mathbf{Z} &\Leftrightarrow \mathbf{x} \text{ solves (3.13)} \\
&\Leftrightarrow \begin{cases} \mathbf{x} \in \mathbf{W} \\ (\exists\, \mathbf{x}^* \in \mathbf{W}^\perp) \quad \mathbf{0} \in \bigtimes_{i=1}^{p+1} C_i x_i + \mathbf{x}^* \end{cases} \\
&\Leftrightarrow \begin{cases} (\exists\, x_1 \in \mathrm{H}) \quad \mathbf{x} = (x_1, L_1 x_1, \ldots, L_p x_1) \\ (\exists\, \mathbf{x}^* \in \mathbf{W}^\perp) \quad \mathbf{0} \in Ax_1 \times B_1(L_1 x_1) \times \cdots \times B_p(L_p x_1) + \mathbf{x}^* \end{cases} \\
&\Leftrightarrow \begin{cases} (\exists\, x_1 \in \mathrm{H}) \quad \mathbf{x} = (x_1, L_1 x_1, \ldots, L_p x_1) \\ (\exists\, (y_1^*, \ldots, y_p^*) \in \mathbf{G}) \quad (0, 0, \ldots, 0) \in \\ \qquad\qquad Ax_1 \times B_1(L_1 x_1) \times \cdots \times B_p(L_p x_1) + \Big(\sum_{k=1}^{p} L_k^* y_k^*, -y_1^*, \ldots, -y_p^* \Big) \end{cases} \\
&\Leftrightarrow \begin{cases} (\exists\, x_1 \in \mathrm{H}) \quad \mathbf{x} = (x_1, L_1 x_1, \ldots, L_p x_1) \\ (\exists\, (y_1^*, \ldots, y_p^*) \in \mathbf{G}) \quad \begin{cases} 0 \in Ax_1 + \sum_{k=1}^{p} L_k^* y_k^* \\ (\forall k \in \{1, \ldots, p\}) \quad y_k^* \in B_k(L_k x_1). \end{cases} \end{cases} \\
&\Leftrightarrow \begin{cases} (\exists\, x_1 \in \mathrm{H}) \quad \mathbf{x} = (x_1, L_1 x_1, \ldots, L_p x_1) \\ 0 \in Ax_1 + \sum_{k=1}^{p} L_k^* \big(B_k(L_k x_1)\big) \end{cases} \\
&\Leftrightarrow (\exists\, x_1 \in Z) \quad \mathbf{x} = (x_1, L_1 x_1, \ldots, L_p x_1), \qquad (3.22)
\end{aligned}
$$

which completes the proof. □

3.4. Framework 3

Our last algorithm connects Problem 1.2 to Problem 3.1 by means of a coupling operator $\mathbf{E}$ mapping to an auxiliary space $\mathbf{K}$ and such that $\ker \mathbf{E}$ coincides with the space $\mathbf{W}$ of (2.8).

Proposition 3.7. *Consider the setting of Problem 1.2, let $(\mathrm{K}_j)_{1 \leqslant j \leqslant r}$ be separable real Hilbert spaces, set*

$$
\mathbf{K} = \bigoplus_{j=1}^{r} \mathrm{K}_j, \qquad (3.23)
$$

and let

$$
\mathbf{E} \colon \mathrm{H} \oplus \mathbf{G} \to \mathbf{K} \colon \mathbf{x} \mapsto \Bigg(\sum_{i=1}^{p+1} E_{ji} x_i \Bigg)_{1 \leqslant j \leqslant r} \qquad (3.24)
$$

be linear, bounded, and such that $\ker \mathbf{E} = \mathbf{W}$. Define $\mathbf{V}$ as in (3.3), where $\mathbf{X}$ is replaced with $\mathrm{H} \oplus \mathbf{G}$, $\mathbf{Y}$ with $\mathbf{K}$, and $\mathbf{M}$ with $\mathbf{E}$, and decompose its projection operator as $\mathrm{proj}_{\mathbf{V}} \colon \mathbf{x} \mapsto (\mathrm{R}_j \mathbf{x})_{1 \leqslant j \leqslant p+1+r}$, where $\mathrm{R}_1 \colon \mathrm{H} \oplus \mathbf{G} \oplus \mathbf{K} \to \mathrm{H}$, $(\forall i \in \{1, \ldots, p\})$ $\mathrm{R}_{1+i} \colon \mathrm{H} \oplus \mathbf{G} \oplus \mathbf{K} \to \mathrm{G}_i$, and $(\forall k \in \{1, \ldots, r\})$ $\mathrm{R}_{p+1+k} \colon \mathrm{H} \oplus \mathbf{G} \oplus \mathbf{K} \to \mathrm{K}_k$. Set $\mathrm{O} = \{0, 1\}^{p+1+r} \smallsetminus \{\mathbf{0}\}$, let $\gamma \in \left]0, +\infty\right[$, let $(\lambda_n)_{n \in \mathbb{N}}$ be a sequence in $\left]0, 2\right[$ such that $\inf_{n \in \mathbb{N}} \lambda_n > 0$ and $\sup_{n \in \mathbb{N}} \lambda_n < 2$, let $\boldsymbol{x}_0$, $\boldsymbol{z}_0$, $(\boldsymbol{a}_n)_{n \in \mathbb{N}}$, and $(\boldsymbol{c}_n)_{n \in \mathbb{N}}$ be $\mathrm{H} \oplus \mathbf{G}$-valued random variables,

let $\boldsymbol{y}_0$, $\boldsymbol{w}_0$, *and* $(\boldsymbol{b}_n)_{n\in\mathbb{N}}$ *be* $\mathbf{K}$*-valued random variables, and let* $(\boldsymbol{\varepsilon}_n)_{n\in\mathbb{N}}$ *be identically distributed* O*-valued random variables. Iterate*

$$
\begin{array}{l}
\textit{for } n = 0, 1, \dots \\
\left|\begin{array}{l}
x_{1,n+1} = x_{1,n} + \varepsilon_{1,n}\big(R_1(\boldsymbol{z}_n, \boldsymbol{w}_n) + a_{1,n} - x_{1,n}\big) \\
z_{1,n+1} = z_{1,n} + \varepsilon_{1,n}\lambda_n\big(J_{\gamma A}\left(2x_{1,n+1} - z_{1,n}\right) + c_{1,n} - x_{1,n+1}\big) \\
\textit{for } k = 1, \dots, p \\
\left|\begin{array}{l}
x_{k+1,n+1} = x_{k+1,n} + \varepsilon_{k+1,n}\big(R_{k+1}(\boldsymbol{z}_n, \boldsymbol{w}_n) + a_{k+1,n} - x_{k+1,n}\big) \\
z_{k+1,n+1} = z_{k+1,n} + \varepsilon_{k+1,n}\lambda_n\big(J_{\gamma B_k}\left(2x_{k+1,n+1} - z_{k+1,n}\right) + c_{k+1,n} - x_{k+1,n+1}\big)
\end{array}\right. \\
\textit{for } j = 1, \dots, r \\
\left|\begin{array}{l}
y_{j,n+1} = y_{j,n} + \varepsilon_{p+1+j,n}\big(R_{p+1+j}(\boldsymbol{z}_n, \boldsymbol{w}_n) + b_{j,n} - y_{j,n}\big) \\
w_{j,n+1} = w_{j,n} - \varepsilon_{p+1+j,n}\lambda_n y_{j,n+1}.
\end{array}\right.
\end{array}\right.
\end{array}
\tag{3.25}
$$

In addition, assume that the following are satisfied:

(i) $\sum_{n\in\mathbb{N}} \sqrt{\mathrm{E}(\|\boldsymbol{a}_n\|^2 \mid \sigma(\boldsymbol{z}_l, \boldsymbol{w}_l)_{0\leqslant l\leqslant n})} < +\infty$, $\sum_{n\in\mathbb{N}} \sqrt{\mathrm{E}(\|\boldsymbol{b}_n\|^2 \mid \sigma(\boldsymbol{z}_l, \boldsymbol{w}_l)_{0\leqslant l\leqslant n})} < +\infty$, $\sum_{n\in\mathbb{N}} \sqrt{\mathrm{E}(\|\boldsymbol{c}_n\|^2 \mid \sigma(\boldsymbol{z}_l, \boldsymbol{w}_l)_{0\leqslant l\leqslant n})} < +\infty$, $\boldsymbol{a}_n \rightharpoonup \mathbf{0}$ P-a.s., *and* $\boldsymbol{b}_n \rightharpoonup \mathbf{0}$ P-a.s.

(ii) *For every* $n \in \mathbb{N}$, $\sigma(\boldsymbol{\varepsilon}_n)$ *and* $\sigma(\boldsymbol{z}_l, \boldsymbol{w}_l)_{0\leqslant l\leqslant n}$ *are independent.*

(iii) *For every* $l \in \{1, \dots, p+1+r\}$, $\mathrm{P}[\varepsilon_{l,0} = 1] > 0$.

Then $(x_{1,n})_{n\in\mathbb{N}}$ *converges weakly* P-a.s. *to a* $\mathbf{Z}$*-valued random variable.*

Proof. In Problem 3.1, set $m = p+1$, $X_1 = H$, $(X_i)_{2\leqslant i\leqslant m} = (G_{i-1})_{2\leqslant i\leqslant m}$, $\mathbf{Y} = \mathbf{K}$, for every $j \in \{1, \dots, r\}$, $D_j = N_{\{0\}}$, and, for every $i \in \{1, \dots, m\}$, $M_{ji} = E_{ji}$. Thus, the subspace $\mathbf{V}$ of (3.3) becomes

$$
\mathbf{V} = \left\{(\mathbf{x}, \mathbf{y}) \in \mathbf{X} \oplus \mathbf{Y} \;\middle|\; (\forall j \in \{1, \dots, r\})\ y_j = \sum_{i=1}^{p+1} E_{ji}x_i\right\},
\tag{3.26}
$$

Further, denote by $\mathbf{x} = (x_1, \dots, x_{p+1})$ a generic element in $H \oplus \mathbf{G}$. In this configuration, (3.2) reduces to

$$
\text{find}\ \ \mathbf{x} \in H \oplus \mathbf{G}\ \ \text{such that}\ \ \mathbf{0} \in \bigtimes_{i=1}^{p+1} C_i x_i + \mathbf{E}^*\big(N_{\{\mathbf{0}\}}(\mathbf{E}\mathbf{x})\big).
\tag{3.27}
$$

We note that Proposition 3.7 is the application of Theorem 3.2 to (3.27) when

$$
C_1 = A \quad \text{and} \quad (\forall i \in \{2, \dots, p+1\}) \quad C_i = B_{i-1}.
\tag{3.28}
$$

Let $\mathbf{Z}$ be the set of solutions to (3.27) in the context of (3.28). Recalling that Z denotes the set of solutions to Problem 1.2, it remains to show that

$$
\mathbf{Z} = \big\{(x_1, L_1x_1, \dots, L_px_1) \mid x_1 \in Z\big\}.
\tag{3.29}
$$

Let $\mathbf{x} \in H \oplus \mathbf{G}$. It follows at once from (3.24) that

$$
\iota_{\mathbf{W}}(\mathbf{x}) = \iota_{\{\mathbf{0}\}}(\mathbf{E}\mathbf{x}).
\tag{3.30}
$$

Hence, we deduce from [6, Corollary 16.53] that

$$
N_{\mathbf{W}}\mathbf{x} = \mathbf{E}^*\big(N_{\{\mathbf{0}\}}(\mathbf{E}\mathbf{x})\big).
\tag{3.31}
$$

Note that the set in (3.31) is nonempty if and only if $\mathbf{x} \in \mathbf{W}$. Consequently,

$$\mathbf{x} \in \mathbf{Z} \Leftrightarrow \mathbf{x} \text{ solves (3.27)} \Leftrightarrow \mathbf{x} \text{ solves (3.13)} \tag{3.32}$$

and the claim follows from (3.22). □

Let us provide some examples of implementations of Proposition 3.7.

Example 3.8. In Proposition 3.7, set $r = p$, $\mathbf{K} = \mathbf{G}$, and, for every $k \in \{1, \dots, p\}$ and every $i \in \{1, \dots, p+1\}$,

$$E_{ki} = \begin{cases} L_k, & \text{if } i = 1; \\ -\mathrm{Id}, & \text{if } i = k+1; \\ 0, & \text{otherwise.} \end{cases} \tag{3.33}$$

Let $\mathbf{x} \in H \oplus \mathbf{G}$, let $\mathbf{y} \in \mathbf{G}$, and set $q = (2\mathrm{Id} + \sum_{k=1}^{p} L_k^* \circ L_k)^{-1}(2x_1 + \sum_{k=1}^{p} L_k^*(x_{k+1} + y_k))$. Then, for every $i \in \{1, \dots, p+1\}$,

$$R_i(\mathbf{x}, \mathbf{y}) = \begin{cases} q, & \text{if } i = 1; \\ \dfrac{1}{2}\big(L_{i-1}q + x_i - y_{i-1}\big), & \text{if } 2 \leqslant i \leqslant p+1; \\ \dfrac{1}{2}\big(L_{i-p-1}q - x_{i-p} + y_{i-p-1}\big), & \text{if } p+2 \leqslant i \leqslant 2p+1. \end{cases} \tag{3.34}$$

Let $(e_n)_{n\in\mathbb{N}}$ be H-valued random variables such that $\sum_{n\in\mathbb{N}} \sqrt{E(\|e_n\|^2 \mid \sigma(z_l, w_l)_{0\leqslant l\leqslant n})} < +\infty$ and $e_n \rightharpoonup 0$ P-a.s., and set

$$Q = \left(2\mathrm{Id} + \sum_{k=1}^{p} L_k^* \circ L_k\right)^{-1}. \tag{3.35}$$

Then (3.25) becomes

$$\begin{array}{l} \text{for } n = 0, 1, \dots \\ \quad q_n = Q\Big(2z_{1,n} + \displaystyle\sum_{k=1}^{p} L_k^*(z_{k+1,n} + w_{k,n})\Big) + e_n \\ \quad x_{1,n+1} = x_{1,n} + \varepsilon_{1,n}(q_n - x_{1,n}) \\ \quad z_{1,n+1} = z_{1,n} + \varepsilon_{1,n}\lambda_n\big(J_{\gamma A}\left(2x_{1,n+1} - z_{1,n}\right) + c_{1,n} - x_{1,n+1}\big) \\ \quad \text{for } k = 1, \dots, p \\ \quad\quad x_{k+1,n+1} = x_{k+1,n} + \varepsilon_{k+1,n}\Big(\dfrac{L_k q_n + z_{k+1,n} - w_{k,n}}{2} - x_{k+1,n}\Big) \\ \quad\quad z_{k+1,n+1} = z_{k+1,n} + \varepsilon_{k+1,n}\lambda_n\big(J_{\gamma B_k}\left(2x_{k+1,n+1} - z_{k+1,n}\right) + c_{k+1,n} - x_{k+1,n+1}\big) \\ \quad \text{for } k = 1, \dots, p \\ \quad\quad y_{k,n+1} = y_{k,n} + \varepsilon_{p+1+k,n}\Big(\dfrac{L_k q_n - z_{k+1,n} + w_{k,n}}{2} - y_{k,n}\Big) \\ \quad\quad w_{k,n+1} = w_{k,n} - \varepsilon_{p+1+k,n}\lambda_n y_{k,n+1} \end{array} \tag{3.36}$$

and Proposition 3.7 asserts that $(x_{1,n})_{n\in\mathbb{N}}$ converges weakly P-a.s. to a solution to Problem 1.2.

The next examples focus on the special case of Problem 1.2 in which, for every $k \in \{1, \dots, p\}$, $G_k = H$ and $L_k = \mathrm{Id}$, that is,

$$\text{find } x \in H \text{ such that } 0 \in Ax + \sum_{k=1}^{p} B_k x. \tag{3.37}$$

Example 3.9. Consider the setting of Example 3.8 where, for every $k \in \{1, \dots, p\}$, $G_k = H$ and $L_k = \mathrm{Id}$. Then, in view of (3.33), the operator $\mathbf{E}$ is defined by setting, for every $k \in \{1, \dots, p\}$ and every $i \in \{1, \dots, p+1\}$,

$$E_{ki} = \begin{cases} \mathrm{Id}, & \text{if } i = 1; \\ -\mathrm{Id}, & \text{if } i = k+1; \\ 0, & \text{otherwise.} \end{cases} \tag{3.38}$$

Further, the operator Q of (3.35) is just $(p+2)^{-1}\mathrm{Id}$. Thus, (3.25) becomes

$$\begin{array}{l} \text{for } n = 0, 1, \dots \\ \left\lfloor \begin{array}{l} q_n = \dfrac{1}{p+2}\Big(2z_{1,n} + \displaystyle\sum_{k=1}^{p}(z_{k+1,n} + w_{k,n})\Big) \\ x_{1,n+1} = x_{1,n} + \varepsilon_{1,n}(q_n - x_{1,n}) \\ z_{1,n+1} = z_{1,n} + \varepsilon_{1,n}\lambda_n\big(J_{\gamma A}\left(2x_{1,n+1} - z_{1,n}\right) + c_{1,n} - x_{1,n+1}\big) \\ \text{for } k = 1, \dots, p \\ \left\lfloor \begin{array}{l} x_{k+1,n+1} = x_{k+1,n} + \varepsilon_{k+1,n}\Big(\dfrac{q_n + z_{k+1,n} - w_{k,n}}{2} - x_{k+1,n}\Big) \\ z_{k+1,n+1} = z_{k+1,n} + \varepsilon_{k+1,n}\lambda_n\big(J_{\gamma B_k}\left(2x_{k+1,n+1} - z_{k+1,n}\right) + c_{k+1,n} - x_{k+1,n+1}\big) \end{array} \right. \\ \text{for } k = 1, \dots, p \\ \left\lfloor \begin{array}{l} y_{k,n+1} = y_{k,n} + \varepsilon_{p+1+k,n}\Big(\dfrac{q_n - z_{k+1,n} + w_{k,n}}{2} - y_{k,n}\Big) \\ w_{k,n+1} = w_{k,n} - \varepsilon_{p+1+k,n}\lambda_n y_{k,n+1} \end{array} \right. \end{array} \right. \end{array} \tag{3.39}$$

and Proposition 3.7 asserts that $(x_{1,n})_{n\in\mathbb{N}}$ converges weakly P-a.s. to a solution to (3.37).

Example 3.10. In Proposition 3.7, set $r = p+1$, $\mathbf{K} = H^{p+1}$, and, for every $k \in \{1, \dots, p+1\}$ and every $i \in \{1, \dots, p+1\}$,

$$E_{ki} = \begin{cases} \dfrac{p}{p+1}\mathrm{Id}, & \text{if } k = i; \\ -\dfrac{1}{p+1}\mathrm{Id}, & \text{if } k \neq i. \end{cases} \tag{3.40}$$

Then $\ker \mathbf{E}$ is the subspace of all the vectors $\mathbf{x} \in H^{p+1}$ such that, for every $i \in \{1, \dots, p+1\}$, $x_i = (p+1)^{-1}\sum_{j=1}^{p+1} x_j$. Hence, for every $i \in \{1, \dots, 2p+2\}$, every $\mathbf{x} \in H^{p+1}$, and every $\mathbf{y} \in H^{p+1}$,

$$R_i(\mathbf{x}, \mathbf{y}) = \begin{cases} \dfrac{1}{2}(x_i + y_i) + \dfrac{1}{2(p+1)}\displaystyle\sum_{j=1}^{p+1}(x_j - y_j), & \text{if } i \leqslant p+1; \\ \dfrac{1}{2}(x_i + y_i) - \dfrac{1}{2(p+1)}\displaystyle\sum_{j=1}^{p+1}(x_j + y_j), & \text{if } p+2 \leqslant i \leqslant 2p+2. \end{cases} \tag{3.41}$$

Then (3.25) becomes

$$
\begin{aligned}
&\text{for } n = 0, 1, \dots \\
&\left|\begin{aligned}
&x_{1,n+1} = x_{1,n} + \varepsilon_{1,n}\Big(\frac{z_{1,n} + w_{1,n}}{2} + \frac{1}{2(p+1)}\sum_{l=1}^{p+1}(z_{l,n} - w_{l,n}) - x_{1,n}\Big) \\
&z_{1,n+1} = z_{1,n} + \varepsilon_{1,n}\lambda_n\big(J_{\gamma A}\left(2x_{1,n+1} - z_{1,n}\right) + c_{1,n} - x_{1,n+1}\big) \\
&\text{for } k = 1, \dots, p \\
&\left|\begin{aligned}
&x_{k+1,n+1} = x_{k+1,n} + \varepsilon_{k+1,n}\Big(\frac{z_{k+1,n} + w_{k+1,n}}{2} + \frac{1}{2(p+1)}\sum_{l=1}^{p+1}(z_{l,n} - w_{l,n}) - x_{k+1,n}\Big) \\
&z_{k+1,n+1} = z_{k+1,n} + \varepsilon_{k+1,n}\lambda_n\big(J_{\gamma B_k}\left(2x_{k+1,n+1} - z_{k+1,n}\right) + c_{k+1,n} - x_{k+1,n+1}\big)
\end{aligned}\right. \\
&\text{for } j = 1, \dots, p+1 \\
&\left|\begin{aligned}
&y_{j,n+1} = y_{j,n} + \varepsilon_{p+1+j,n}\Big(\frac{z_{j,n} + w_{j,n}}{2} - \frac{1}{2(p+1)}\sum_{l=1}^{p+1}(z_{l,n} + w_{l,n}) - y_{j,n}\Big) \\
&w_{j,n+1} = w_{j,n} - \varepsilon_{p+1+j,n}\lambda_n y_{j,n+1}
\end{aligned}\right.
\end{aligned}
\tag{3.42}
$$

and Proposition 3.7 asserts that $(x_{1,n})_{n\in\mathbb{N}}$ converges weakly P-a.s. to a solution to (3.37).

Remark 3.11. In Example 3.9, the operator $\mathbf{E}$ applied to $\mathbf{x} \in H^{p+1}$ couples each agent in $(x_2, \dots, x_{p+1})$ with x_1. In Example 3.10 the operator $\mathbf{E}$ applied to $\mathbf{x} \in H^{p+1}$ couples each agent in $(x_1, \dots, x_{p+1})$ with the average of all the agents. Various alternative coupling operators $\mathbf{E}$ can be considered to enforce the condition $x_1 = \cdots = x_{p+1}$.

3.5. Computation of inverse operators

The existing algorithms presented in Section 2.2 require the computation of norms of arbitrary linear operators whereas the proposed algorithms of Sections 3.2–3.4 require the inversion of strongly positive Hermitian operators of the type $\mathbf{Id} + \mathbf{L}^* \circ \mathbf{L}$. Note that, because of the strongly positive Hermitian structure of $\mathbf{Id} + \mathbf{L}^* \circ \mathbf{L}$, the computation of the inverse is typically much cheaper than the computation of the norm of $\mathbf{L}$ in (2.10) or those of $(U_k^{1/2} L_k W^{1/2})_{1\leqslant k\leqslant p}$ in (2.12). In a finite-dimensional setting, in full generality, if $\mathbf{Id} + \mathbf{L}^* \circ \mathbf{L}$ has size N, then its inversion via the Cholesky decomposition method requires about $N^3/6$ multiplications. However, this complexity can be reduced in several standard scenarios. Here are two examples in $H = \mathbb{R}^N$ that will be used in Section 5.

Example 3.12.

(i) If, for every $k \in \{1, \dots, p\}$, $L_k = \mathrm{Id}$, then

$$
\begin{cases}
\left(\mathrm{Id} + \sum_{k=1}^{p} L_k^* \circ L_k\right)^{-1} = \dfrac{1}{1+p}\mathrm{Id} \\
\left(2\mathrm{Id} + \sum_{k=1}^{p} L_k^* \circ L_k\right)^{-1} = \dfrac{1}{2+p}\mathrm{Id}.
\end{cases}
\tag{3.43}
$$

The cost of the inversion is $O(1)$.

(ii) Suppose that, for every $k \in \{1, \dots, p\}$, L_k is a block-Toeplitz. Then, following a standard argument [2], each L_k can be approximated by a block-circulant matrix with convolution

kernel ℓ_k and

$$\begin{cases} \left(\mathrm{Id} + \sum_{k=1}^{p} L_k^* \circ L_k\right)^{-1} & : x \mapsto \mathfrak{F}^{-1}\left(\mathfrak{F}(x) \div \left(1 + \sum_{k=1}^{p} |\mathfrak{F}(\ell_k)|^2\right)\right) \\ \left(2\mathrm{Id} + \sum_{k=1}^{p} L_k^* \circ L_k\right)^{-1} & : x \mapsto \mathfrak{F}^{-1}\left(\mathfrak{F}(x) \div \left(2 + \sum_{k=1}^{p} |\mathfrak{F}(\ell_k)|^2\right)\right). \end{cases} \tag{3.44}$$

where $\mathfrak{F}$ denotes the discrete Fourier transform and $\div$ denotes pointwise division. The cost of the inversion using the fast Fourier transform is $O(N \log(N))$ [2].

(iii) The worst case is if the operators $(L_k)_{1\leqslant k\leqslant p}$ do not present a special structure. Even so, the composed operators $\mathrm{Id}+\sum_{k=1}^{p} L_k^*\circ L_k$ and $2\mathrm{Id}+\sum_{k=1}^{p} L_k^*\circ L_k$ are symmetric and positive-definite. Hence they admit a Cholesky decomposition. The cost of computing the Cholesky decomposition is $O(N^3)$ (one time) and the cost of solving the linear system using the Cholesky decomposition is $O(N^2)$. It will be shown in the numerical experiments that in the case when no special structure is present, Framework 2 is preferred since the application of the inverse operator does not occur at every iteration.

§4. Minimization problems

We dedicate this section to the minimization setting of Problem 1.1. Let us first formalize the connection between Problems 1.1 and 1.2.

Proposition 4.1. *In Problem 1.2, set* $A = \partial f$ *and* $(\forall k \in \{1, \dots, p\})\ B_k = \partial g_k$. *Then every solution to* (1.2) *solves Problem 1.1.*

Proof. Set $\mathbf{L}\colon H \to \mathbf{G}\colon x \mapsto (L_1 x, \dots, L_p x)$ and $\mathbf{g}\colon \mathbf{G} \to]-\infty, +\infty]\colon \mathbf{y} \mapsto \sum_{k=1}^{p} g_k(y_k)$. Then $\mathbf{L}^*\colon \mathbf{G} \to H\colon \mathbf{y} \mapsto \sum_{k=1}^{p} L_k^* y_k$. Hence, it follows from [6, Proposition 16.9] that

$$x \in \operatorname{zer}\left(\partial f + \sum_{k=1}^{p} L_k^* \circ (\partial g_k) \circ L_k\right) = \operatorname{zer}\big(\partial f + \mathbf{L}^* \circ (\partial \mathbf{g}) \circ \mathbf{L}\big). \tag{4.1}$$

However, [6, Proposition 27.5(i)] asserts that

$$\operatorname{zer}\left(\partial f + \sum_{k=1}^{p} \mathbf{L}^* \circ (\partial \mathbf{g}) \circ \mathbf{L}\right) \subset \operatorname{Argmin}\big(f + \mathbf{g} \circ \mathbf{L}\big) = \operatorname{Argmin}\left(f + \sum_{k=1}^{p} g_k \circ L_k\right), \tag{4.2}$$

which confirms the claim. □

Problem 1.1 relies on the assumption that $\operatorname{zer}(\partial f + \sum_{k=1}^{p} L_k^* \circ (\partial g_k) \circ L_k) \neq \varnothing$. Let us provide sufficient conditions that guarantee it.

Proposition 4.2. *Let* H *be a separable real Hilbert space and* $f \in \Gamma_0(H)$. *For every* $k \in \{1, \dots, p\}$, *let* G_k *be a separable real Hilbert space, let* $g_k \in \Gamma_0(G_k)$, *and let* $0 \neq L_k\colon H \to G_k$ *be linear and bounded. Set*

$$\mathbf{S} = \big\{(L_k x - y_k)_{1\leqslant k\leqslant p} \mid x \in \operatorname{dom} f \text{ and } (\forall k \in \{1, \dots, p\})\ \ y_k \in \operatorname{dom} g_k\big\}. \tag{4.3}$$

Then $\operatorname{zer}(\partial f + \sum_{k=1}^{p} L_k^* \circ (\partial g_k) \circ L_k) \neq \varnothing$ *if the following hold:*

(i) $f(x) + \sum_{k=1}^{p} g_k(L_k x) \to +\infty$ *as* $\|x\| \to +\infty$.

(ii) *Any of the following is satisfied:*

(a) *The cone generated by* $\mathbf{S}$ *is a closed vector subspace of* $\mathbf{G}$.

(b) *For every* $k \in \{1, \ldots, p\}$, g_k *is real-valued.*

(c) H *and* $(G_k)_{1 \leqslant k \leqslant p}$ *are finite-dimensional, and there exists* $x \in \operatorname{ri} \operatorname{dom} f$ *such that*

$$(\forall k \in \{1, \ldots, p\}) \quad L_k x \in \operatorname{ri} \operatorname{dom} g_k, \tag{4.4}$$

where ri *stands for the relative interior.*

Proof. Set $\mathbf{L} \colon H \to \mathbf{G} \colon x \mapsto (L_1 x, \ldots, L_p x)$ and $\mathbf{g} \colon \mathbf{G} \to]-\infty, +\infty] \colon \mathbf{y} \mapsto \sum_{k=1}^{p} g_k(y_k)$. Then $\mathbf{L}$ is linear and bounded, $\mathbf{g} \in \Gamma_0(\mathbf{G})$, $\mathbf{S} = \big\{\mathbf{L}x - \mathbf{y} \mid x \in \operatorname{dom} f \text{ and } \mathbf{y} \in \operatorname{dom} \mathbf{g}\big\}$, and $f + \mathbf{g} \circ \mathbf{L} = f + \sum_{k=1}^{p} g_k \circ L_k$. On the other hand, it follows from (ii) that $\mathbf{0} \in \mathbf{S}$, which implies that $\operatorname{dom}(f + \mathbf{g} \circ \mathbf{L}) \neq \varnothing$. Thus, because $f + \mathbf{g} \circ \mathbf{L}$ is also lower semicontinuous and convex, we have $f + \mathbf{g} \circ \mathbf{L} \in \Gamma_0(H)$. Hence, since (i) states that $f(x) + \mathbf{g}(\mathbf{L}x) \to +\infty$ as $\|x\| \to +\infty$, it follows from [6, Proposition 11.15(i)] that

$$\operatorname{Argmin}(f + \mathbf{g} \circ \mathbf{L}) \neq \varnothing. \tag{4.5}$$

However, (ii) and [6, Proposition 27.5(iii)] guarantee that

$$\operatorname{Argmin}(f + \mathbf{g} \circ \mathbf{L}) = \operatorname{zer}\big(\partial f + \mathbf{L}^* \circ (\partial \mathbf{g}) \circ \mathbf{L}\big), \tag{4.6}$$

which completes the proof. □

In view of Proposition 4.1 and (2.4), we obtain the following solution methods for Problem 1.1.

Corollary 4.3. *Consider the setting of Problem 1.1 and set* $F = \operatorname{Argmin}(f + \sum_{k=1}^{p} g_k \circ L_k)$. *In* (3.8), (3.10), *and* (3.25), *replace the resolvent operators* $(J_{\gamma A}, J_{\gamma B_1}, \ldots, J_{\gamma B_p})$ *by the proximity operators* $(\operatorname{prox}_{\gamma f}, \operatorname{prox}_{\gamma g_1}, \ldots, \operatorname{prox}_{\gamma g_p})$. *Then Propositions 3.5, 3.6, and 3.7 provide sequences* $(x_{1,n})_{n \in \mathbb{N}}$ *which converge weakly* P-a.s. *to an* F*-valued random variable.*

§5. Numerical experiments

5.1. Preamble

We present four experiments to illustrate the numerical behavior of the three algorithmic frameworks presented in Section 3. These algorithms are initialized by setting $\boldsymbol{x}_0$, $\boldsymbol{z}_0$, $\boldsymbol{y}_0$, and $\boldsymbol{w}_0$ to $\mathbf{0}$, and they use the proximal parameter $\gamma = 1.0$ and the relaxation strategy $(\forall n \in \mathbb{N})\ \lambda_n = 1.9$. The random variable $\boldsymbol{\varepsilon}_0$ activates operator indices in $\{1, \ldots, p+1\}$ (Framework 1), $\{1, \ldots, p+2\}$ (Framework 2), and $\{1, \ldots, 2p+1\}$ (Framework 3 using Example 3.8), with a uniform distribution.

We also provide comparisons with the existing methods of Section 2.2 when applicable, because they do provide almost sure iterate convergence to a solution, although they do not satisfy the requirements **R2**–**R3**:

- Algorithm (2.11) is initialized with $x_{1,0} = 0$ and $\boldsymbol{y}_0 = \mathbf{0}$. Further, for every $k \in \{1, \ldots, p\}$, $\pi_k = 1/p$ and, to enforce (2.10), we set $\tau_0 = 0.9/\sqrt{p}$ and $\sigma_0 = 1/\big(\sqrt{p} \max_{1 \leqslant k \leqslant p} \|L_k\|^2\big)$. In addition we set $\chi_0 = 0.5$, $\eta = 0.5$, and $\delta = 1.5$. We recall that algorithm (2.11) can activate only one operator at each iteration and does not satisfy **R2**–**R4**.

- Algorithm (2.13) is initialized with $x_{1,0} = 0$ and $v_0 = \mathbf{0}$. Further, $W = 0.9\tau\mathrm{Id}$ and, for every $k \in \{1, \ldots, p\}$, $U_k = (\tau/\|L_k\|^2)\mathrm{Id}$, where $\lambda_n \equiv 1$ and, to enforce (2.12), $\tau = 1/\sqrt{2p}$. We recall that algorithm (2.13) does not satisfy **R2**–**R3**.

These parameters were found to enhance the performance of these two algorithms. The first three experiments (Sections 5.2–5.4) correspond to minimization problems fitting the format of Problem 1.1. The last experiment (Section 5.5) is a nonminimization problem that fits the format of Problem 1.2, and algorithm (2.11) is therefore not applicable.

5.2. Signal restoration

The goal is to recover the original signal $\overline{x} \in H = \mathbb{R}^N$ ($N = 1000$) shown in Figure 2(a) from $M = 10$ noisy observations $(r_l)_{1\leqslant l\leqslant M}$ given by

$$(\forall l \in \{1, \ldots, M\}) \quad r_l = L_l\overline{x} + w_l \tag{5.1}$$

where, for every $l \in \{1, \ldots, M\}$, $L_l \colon \mathbb{R}^N \to \mathbb{R}^N$ is a known linear operator, $\eta_l \in \left]0, +\infty\right[$, and $w_l \in [-\eta_l, \eta_l]^N$ is the realization of a bounded random noise vector. The parameters $(\eta_l)_{1\leqslant l\leqslant M} \in \left]0, +\infty\right[^M$ are not known exactly and are underestimated by $(\xi_l)_{1\leqslant l\leqslant M} \in \left]0, +\infty\right[^M$. For every $l \in \{1, \ldots, M\}$, L_l is a Gaussian convolution filter with zero mean and standard deviation taken uniformly in $[20, 40]$, $\eta_l = 0.1$, w_l is taken uniformly in $[-\eta_l, \eta_l]^N$, and $\xi_l = 0.07$. For every $l \in \{1, \ldots, M\}$ and every $j \in \{1, \ldots, N\}$, set $Z_{l,j} = [\langle r_l \,|\, e_j\rangle - \xi_l, \langle r_l \,|\, e_j\rangle + \xi_l]$. Since the intersection of these sets is empty, we cannot recover the signal by solving the associated convex feasibility

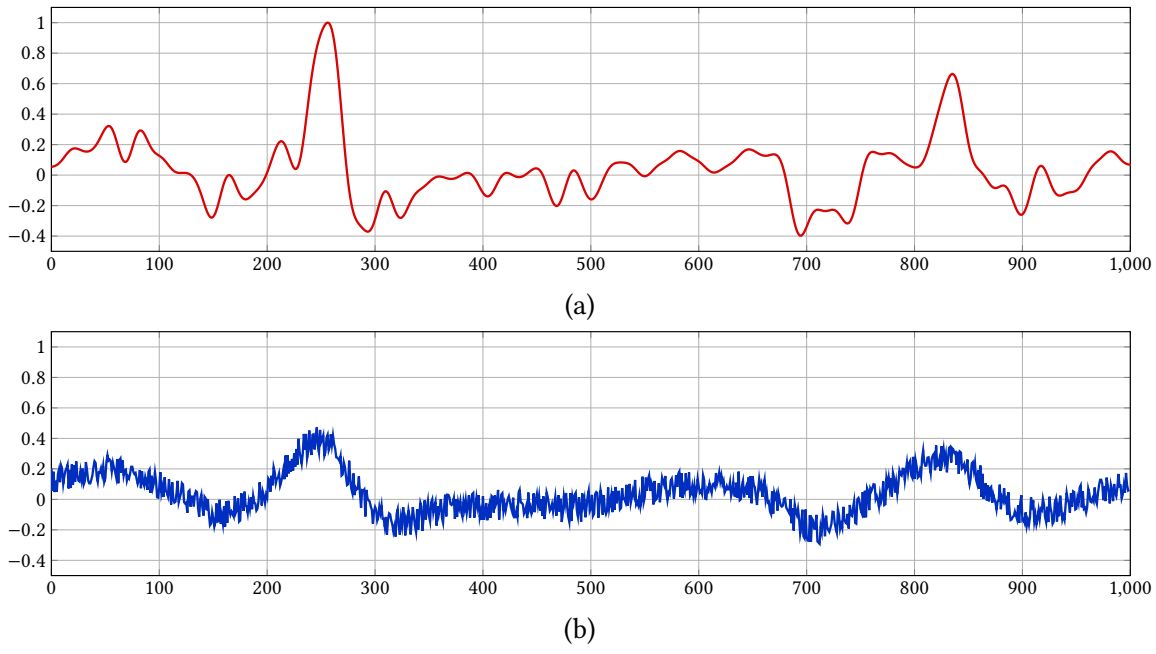


Figure 2: Experiment of Section 5.2. (a): Original signal $\overline{x}$. (b): Noisy observation r_1.

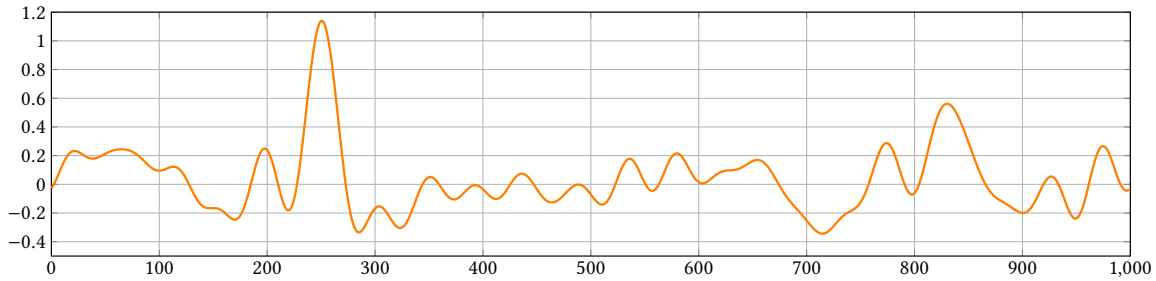


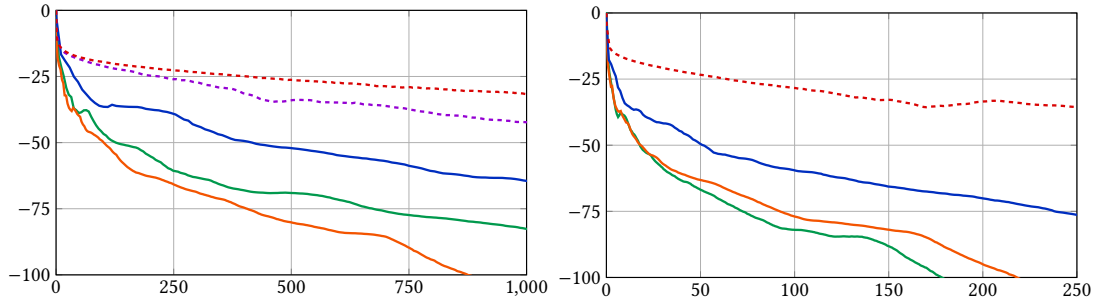
Figure 3: Experiment of Section 5.2. Solution produced by Framework 2.

problem. Instead, our objective is to solve an instantiation of Problem 1.1 with $p = MN$, to wit,

$$\underset{x\in\mathbb{R}^N}{\text{minimize}} \;\; \alpha\|x\| + \sum_{l=1}^{M}\sum_{j=1}^{N} d_{Z_{l,j}}\big(\langle L_l x \,|\, e_j\rangle\big), \tag{5.2}$$

where $\alpha = 0.05$. Since, for every $x \in \mathbb{R}^N$, $\alpha\|x\| \leqslant \alpha\|x\| + \sum_{l=1}^{M}\sum_{j=1}^{N} d_{Z_{l,j}}\big(\langle L_l x \,|\, e_j\rangle\big)$, condition (i) in Proposition 4.2 holds. In addition, for every $l \in \{1,\ldots,M\}$ and every $j \in \{1,\ldots,N\}$, $d_{Z_{l,j}}$ is real-valued. Hence, condition (ii)(b) in Proposition 4.2 holds as well, which confirms that (5.2) is an instance of Problem 1.1. We can thus invoke Corollary 4.3. The three frameworks of Sections 3.2–3.4 are used to solve (5.2), where the operator $\mathbf{E}$ in Proposition 3.7 is that of Example 3.8. Two experiments are conducted: the random variable $\boldsymbol{\varepsilon}_0$ produces (a) 1 activation with 1 core; and (b) 8 activations with 8 cores. Given $\gamma \in \left]0,+\infty\right[$, the operators $(\mathrm{prox}_{\gamma d_{Z_{l,j}}})_{1\leqslant l\leqslant M,1\leqslant j\leqslant N}$ are computed via [6, Example 24.28] and $\mathrm{prox}_{\gamma\|\cdot\|}$ via [6, Example 24.20]. Furthermore, the convolutions and the inversions of linear operators are implemented using the fast Fourier transform [2]; see Example 3.12(ii). As mentioned in Section 5.1, we also compare with:

- Algorithm (2.11), which can activate only one operator at each iteration.
- Algorithm (2.13), where the random variable $\boldsymbol{\varepsilon}_0$ activates (a) 1; and (b) 8 indices in $\{1,\ldots,p\}$ with a uniform distribution at each iteration.



(a) (b)

Figure 4: Experiment of Section 5.2. Normalized error $20\log_{10}(\|x_{1,n}-x_\infty\|/\|x_{1,0}-x_\infty\|)$ (dB) versus execution time (s). (a): Block size 1 with 1 core. (b): Block size 8 with 8 cores. Green: Framework 1. Orange: Framework 2. Blue: Framework 3 with Example 3.8. Dashed violet: Algorithm (2.11). Dashed red: Algorithm (2.13).

The solution produced by Framework 2 is shown in Figure 3. We display in Figure 4 the normalized error versus execution time.

5.3. Overlapping group lasso regression

We address the overlapping group lasso regression problem of [44]. Here $H = \mathbb{R}^N$ and q groups of indices $(I_k)_{1\leqslant k\leqslant q}$ in $\{1,\dots,N\}$ are present, with $\bigcup_{k=1}^{q} I_k = \{1,\dots,N\}$. In addition, for every $k \in \{1,\dots,q\}$,

$$S_k\colon \mathbb{R}^N \to \mathbb{R}^{\operatorname{card} I_k}\colon x = (\xi_j)_{1\leqslant j\leqslant N} \mapsto (\xi_j)_{j\in I_k}. \tag{5.3}$$

The goal is to

$$\underset{x\in\mathbb{R}^N}{\text{minimize}} \ \frac{\alpha}{2}\|Ax-b\|^2 + \frac{1}{q}\sum_{k=1}^{q}\|S_k x\|, \tag{5.4}$$

where $A \in \mathbb{R}^{M\times N}$, $b = (\beta_l)_{1\leqslant l\leqslant M} \in \mathbb{R}^M$, and $\alpha \in \left]0,+\infty\right[$. In the experiment, $M = 1200$, $N = 3610$, $q = 40$, and, as in [44], $\alpha = 5/q^2$. The entries of A are independent and identically distributed (i.i.d.) samples from a $\mathcal{N}(1,10)$ distribution. The entries of the reference vector $\bar{x} \in \mathbb{R}^N$ are i.i.d. samples from a uniform distribution on $[0,10]$, and $b = A\bar{x} + w$, where $w \in \mathbb{R}^M$ has entries that are i.i.d. samples from a $\mathcal{N}(0,0.1)$ distribution. We split the term $\|Ax-b\|^2$ into a sum of 30 blocks of 40 entries each. Finally, the groups are defined by

$$(\forall k \in \{1,\dots,q\}) \ \ I_k = \{90k-89,\dots,90k+10\}. \tag{5.5}$$

Let $(a_l)_{1\leqslant l\leqslant M}$ be the rows of A. Then (5.4) is equivalent to

$$\underset{x\in\mathbb{R}^N}{\text{minimize}} \ \sum_{k=1}^{p} g_k(L_k x), \tag{5.6}$$

where $p = 70$,

$$(\forall k \in \{1,\dots,30\}) \quad \begin{cases} L_k\colon \mathbb{R}^N \to \mathbb{R}^{40}\colon x \mapsto \big(\langle x \mid a_l\rangle\big)_{40(k-1)+1\leqslant l\leqslant 40k} \\ g_k\colon \mathbb{R}^{40} \to \mathbb{R}\colon y \mapsto \dfrac{\alpha}{2}\|y - (\beta_l)_{40(k-1)+1\leqslant l\leqslant 40k}\|^2, \end{cases} \tag{5.7}$$

and

$$(\forall k \in \{31,\dots,70\}) \quad \begin{cases} L_k\colon \mathbb{R}^N \to \mathbb{R}^{100}\colon x \mapsto S_{k-30}x \\ g_k\colon \mathbb{R}^{100} \to \mathbb{R}\colon y \mapsto \dfrac{1}{q}\|y\|. \end{cases} \tag{5.8}$$

Let $x = (\xi_l)_{1\leqslant l\leqslant N} \in \mathbb{R}^N$ and $j \in \{1,\dots,N\}$. Since $\bigcup_{k=1}^{q} I_k = \{1,\dots,N\}$,

$$\frac{1}{q}\sum_{k=1}^{q}\|S_k x\| = \frac{1}{q}\sum_{k=1}^{q}\|(\xi_l)_{l\in I_k}\| = \frac{1}{q}\sum_{k=1}^{q}\sqrt{\sum_{l\in I_k}|\xi_l|^2} \geqslant \frac{1}{q}|\xi_j|. \tag{5.9}$$

In turn,

$$\sum_{k=1}^{70} g_k(L_k x) = \|Ax-b\|^2 + \frac{1}{q}\sum_{k=1}^{q}\|S_k x\| \geqslant \frac{1}{qN}\sum_{j=1}^{N}|\xi_j| \geqslant \frac{1}{qN}\sqrt{\sum_{j=1}^{N}|\xi_j|^2} = \frac{1}{qN}\|x\|, \tag{5.10}$$

which ensures that condition (i) in Proposition 4.2 holds. In addition, for every $k \in \{1, \dots, 70\}$, g_k is real-valued. Hence, condition (ii)(b) in Proposition 4.2 holds as well. Therefore Proposition 4.2 guarantees that (5.6) is an instance of Problem 1.1 and we invoke Corollary 4.3 to justify the convergence of the algorithms. We employ the three frameworks of Sections 3.2–3.4 to solve (5.4), where the operator $\mathbf{E}$ in Proposition 3.7 is that defined in Example 3.8. Two experiments are conducted: the random variable $\boldsymbol{\varepsilon}_0$ produces (a) 1 activation with 1 core; and (b) 8 activations with 8 cores. Given $\gamma \in]0, +\infty[$ and $z \in \mathbb{R}^{40}$, we compute $\operatorname{prox}_{\gamma\|\cdot\|}$ via [6, Example 24.20], $\operatorname{prox}_{\gamma\|\cdot - z\|^2}$ via [6, Proposition 24.8(i)], and the inverse operators by solving the linear systems with Example 3.12(iii). We also compare with:

- Algorithm (2.11).
- Algorithm (2.13), where the random variable $\boldsymbol{\varepsilon}_0$ activates (a) 1; and (b) 8 indices in $\{1, \dots, p\}$ with a uniform distribution at each iteration.

We display in Figure 5 the normalized error versus execution time.

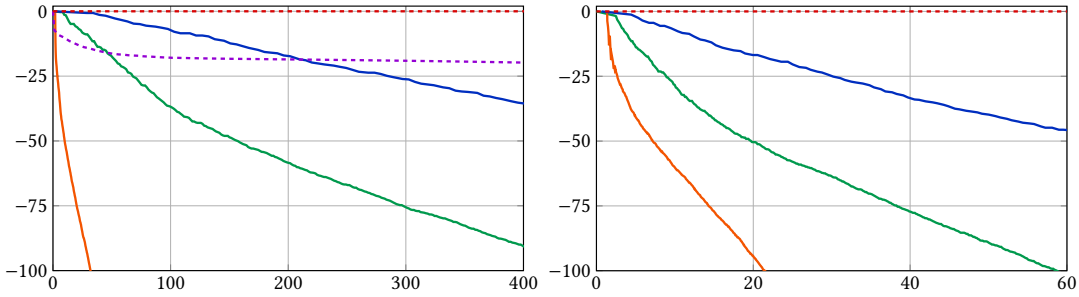


Figure 5: Experiment of Section 5.3. Normalized error $20 \log_{10}(\|x_{1,n} - x_\infty\| / \|x_{1,0} - x_\infty\|)$ (dB) versus execution time (s). (a): Block size 1 with 1 core. (b): Block size 8 with 8 cores. Green: Framework 1. Orange: Framework 2. Blue: Framework 3 with Example 3.8. Dashed violet: Algorithm (2.11). Dashed red: Algorithm (2.13).

5.4. Classification using the hinge loss

We address a binary classification problem. The training data samples $(u_k, \xi_k)_{1 \leqslant k \leqslant p}$ are in $\mathbb{R}^N \times \{-1, 1\}$ and the goal is to learn a linear classifier $x \in H = \mathbb{R}^N$. For this purpose, we solve the instance of Problem 1.1 corresponding to the support vector machine model

$$\operatorname*{minimize}_{x \in \mathbb{R}^N} \ \frac{\alpha}{2}\|x\|^2 + \frac{1}{p}\sum_{k=1}^{p} g_k(x), \tag{5.11}$$

where $\alpha \in]0, +\infty[$ and, for every $k \in \{1, \dots, p\}$, $g_k \colon x \mapsto \max\{0, 1 - \xi_k \langle x \mid u_k \rangle\}$. In the experiment, $N = 1500$, $\alpha = 1$, $p = 750$, and, for every $k \in \{1, \dots, p\}$, the entries of u_k are i.i.d. samples from a $\mathcal{N}(100, 10)$ distribution, and $(\xi_k)_{1 \leqslant k \leqslant p}$ are i.i.d. samples from a uniform distribution on $\{-1, 1\}$. Since, for every $x \in \mathbb{R}^N$, $(\alpha/2)\|x\|^2 \leqslant (\alpha/2)\|x\|^2 + \sum_{k=1}^{p} g_k(x)$, condition (i) in Proposition 4.2 holds. In addition, for every $k \in \{1, \dots, p\}$, g_k is real-valued, so that condition (ii)(b) in Proposition 4.2

holds as well. This guarantees that (5.11) is an instance of Problem 1.1 and we can therefore invoke Corollary 4.3. We employ four methods to solve this problem: Framework 1, Framework 2, and Framework 3 using the operators $\mathbf{E}$ defined in Examples 3.9 and 3.10. In the case of Example 3.10 in Framework 3, the random variable $\boldsymbol{\varepsilon}_0$ activates indices uniformly in $\{1, \ldots, 2p + 2\}$. Three experiments are conducted: the random variable $\boldsymbol{\varepsilon}_0$ produces (a) 1 activation with 1 core; (b) 8 activations with 8 cores, and (c) 32 activations with 32 cores. Given $\gamma \in]0, +\infty[$, the operators $(\mathrm{prox}_{\gamma g_k})_{1 \leqslant k \leqslant p}$ are computed via [6, Example 24.37]. The inverse operators are explicitly computed in Example 3.12(i).

We also compare with:

- Algorithm (2.11), which can activate only one operator at each iteration.
- Algorithm (2.13), where the random variable $\boldsymbol{\varepsilon}_0$ activates (a) 1; (b) 8; and (c) 32 indices in $\{1, \ldots, p\}$ with a uniform distribution at each iteration.

We display in Figure 6 the normalized error versus execution time for each instance. The execution time is evaluated based on the assumption that the computation corresponding to each selected index is assigned to a dedicated core and that all the cores are working in parallel.

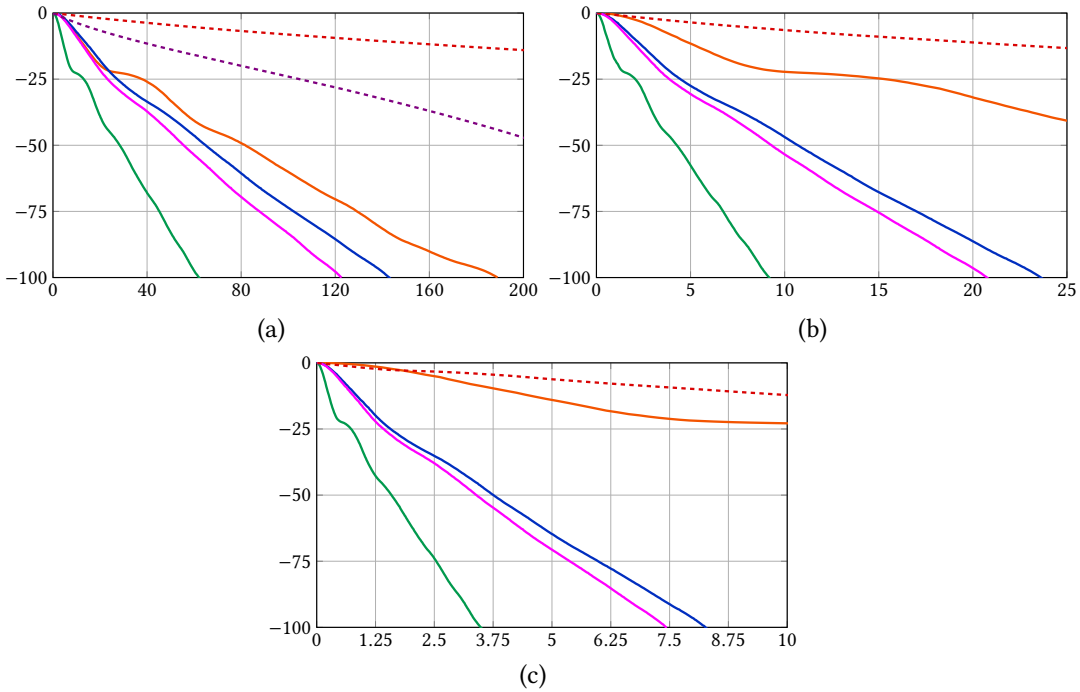


Figure 6: Experiment of Section 5.4. Normalized error $20 \log_{10}(\|x_{1,n} - x_\infty\| / \|x_{1,0} - x_\infty\|)$ (dB) versus execution time (s). (a): Block size 1 with 1 core. (b): Block size 8 with 8 cores. (c): Block size 32 with 32 cores. Green: Framework 1. Orange: Framework 2. Blue: Framework 3 with Example 3.9. Magenta: Framework 3 with Example 3.10. Dashed violet: Algorithm (2.11). Dashed red: Algorithm (2.13).

5.5. Image reconstruction from phase

In contrast with the previous examples, we consider a data analysis framework, first proposed in [18], which requires the monotone inclusion format of Problem 1.2 and is not reducible to the minimization setting of Problem 1.1. The goal is to recover an image in a nonempty closed convex subset C of H from p nonlinear observations $(r_k)_{1\leqslant k\leqslant p}$ produced by Wiener models, namely,

$$\text{find} \quad x \in C \quad \text{such that} \quad (\forall k \in \{1,\dots,p\}) \quad r_k = F_k(L_k x), \tag{5.12}$$

where each operator $F_k\colon G_k \to G_k$ is firmly nonexpansive and each operator $L_k\colon H \to G_k$ is linear and bounded. In many instances, the operators $(F_k)_{1\leqslant k\leqslant p}$ or $(L_k)_{1\leqslant k\leqslant p}$ may be imperfectly known or the model may be corrupted by perturbations and, as a result, (5.12) may not have solutions. A classical approach would be to relax it into a minimization problem such as the least-squares model

$$\underset{x\in C}{\text{minimize}} \quad \sum_{k=1}^{p} \|F_k(L_k x) - r_k\|^2. \tag{5.13}$$

However, because of the nonlinearity of the operators $(F_k)_{1\leqslant k\leqslant p}$, the resulting optimization problem is nonconvex and usually intractable. The strategy of [18] consists in relaxing (5.12) into the variational inequality problem

$$\text{find} \quad x \in C \quad \text{such that} \quad (\forall y \in C) \quad \sum_{k=1}^{p} \alpha_k \langle L_k(y-x) \mid F_k(L_k x) - r_k\rangle \geqslant 0, \tag{5.14}$$

where the weights $(\alpha_k)_{1\leqslant k\leqslant p}$ are in $]0,+\infty[$. As shown there, (5.14) is an exact relaxation of (5.12) in the sense that, if (5.12) happens to have solutions, they are the same as those of (5.14). Let us introduce the operators

$$(\forall k \in \{1,\dots,p\}) \quad B_k = \alpha_k(F_k - r_k), \tag{5.15}$$

which are maximally monotone by [6, Example 20.30]. Then, in terms of the normal cone operator of (2.5), (5.14) is equivalent to

$$\text{find} \quad x \in H \quad \text{such that} \quad 0 \in N_C x + \sum_{k=1}^{p} L_k^*\big(B_k(L_k x)\big). \tag{5.16}$$

This inclusion problem is now in the format of Problem 1.2 with $A = N_C$, which allows us to apply the algorithms proposed in Sections 3.2–3.4 to solve it with guaranteed almost sure convergence of the iterates to a solution.

The specific image recovery problem under consideration is similar to that of [18, Section 5.1]. The goal is to recover the original image $\overline{x} \in H = \mathbb{R}^N$ ($N = 256^2$) of Figure 7(a) from the following prior knowledge and $p = 62$ observations:

(i) Bounds on pixel values: $\overline{x} \in C = [0,255]^N$.

(ii) The degraded images $(r_k)_{1\leqslant k\leqslant 20}$ in $\mathbb{R}^N$ are obtained via a blurring process, addition of noise, and finally clipping. In terms of the model (5.12), for every $k \in \{1,\dots,20\}$, $G_k = \mathbb{R}^N$, $r_k = F_k(L_k\overline{x} + w_k)$, where L_k performs convolution with a Gaussian kernel with a standard

deviation of 3, $w_k \in \mathbb{R}^N$ is a noise vector with i.i.d. entries uniformly distributed in $[-50, 50]$, and

$$F_k \colon \mathbb{R}^N \to \mathbb{R}^N \colon y \mapsto \operatorname{proj}_{C_1} y, \quad \text{where} \quad C_1 = [0, 60]^N \tag{5.17}$$

models a hard clipping process. This nonlinear measurement process models a low-quality image acquired by a device which saturates at photon counts beyond a certain threshold. As an example, the first degraded image r_1 is shown in Figure 7(b).

(iii) The degraded images $(r_k)_{21 \leqslant k \leqslant 40}$ in $\mathbb{R}^N$ are obtained by a process similar to (ii). Here, for every $k \in \{21, \dots, 40\}$, the blurring operator L_k performs a convolution in the vertical direction with a uniform kernel of length 20, the entries of the noise vector $w_k \in \mathbb{R}^N$ are i.i.d. and uniformly distributed in $[-70, 70]$, and pixel values beyond 90 are soft-clipped by

$$F_k \colon \mathbb{R}^N \to \mathbb{R}^N \colon (\eta_j)_{1 \leqslant j \leqslant N} \mapsto \left(\frac{90 \max\{0, \eta_j\}}{90 + |\eta_j|} \right)_{1 \leqslant j \leqslant N}. \tag{5.18}$$

As an example, the degraded image r_{21} is shown in Figure 7(c).

(iv) The degraded images $(r_k)_{41 \leqslant k \leqslant 60}$ in $\mathbb{R}^N$ are obtained through an image formation process similar to that of (iii). For every $k \in \{41, \dots, 60\}$, the blurring operator L_k now performs a convolution in the horizontal direction with a uniform kernel of length 24, and the entries of the noise vector $w_k \in \mathbb{R}^N$ are i.i.d. and uniformly distributed in $[-90, 90]$. For every $k \in \{41, \dots, 60\}$, pixel values beyond 90 are soft-clipped by the same operator F_k as in (5.18).

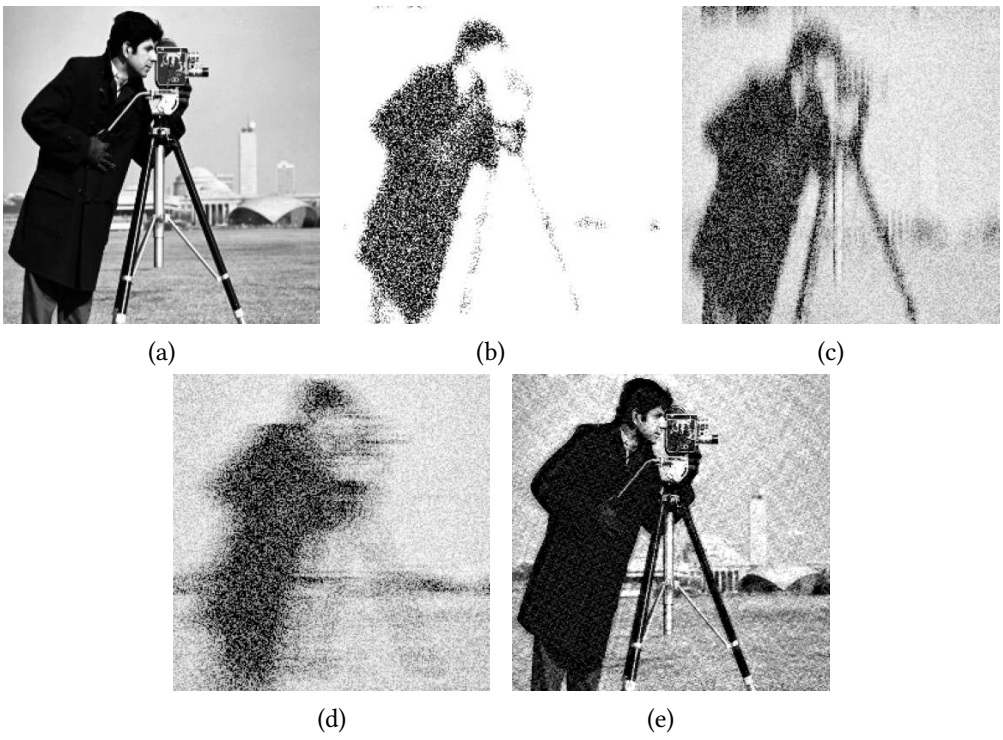


Figure 7: Experiment of Section 5.5: (a): Original image $\overline{x}$. (b): Degraded image r_1. (c): Degraded image r_{21}. (d): Degraded image r_{41}. (e): Recovered image.

(v) The mean pixel value $\rho = 137$ of $\overline{x}$ is known. This information is imposed on a candidate solution $x \in \mathbb{R}^N$ via the equation $\langle x \,|\, \mathbf{1} \rangle = N\rho$, where $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^N$, which corresponds to the model $r_{61} = F_{61}(L_{61}x)$, with $G_{61} = \mathbb{R}$, $L_{61} = \langle \cdot \,|\, \mathbf{1} \rangle$, $r_{61} = N\rho$, and $F_{61} = \mathrm{Id}$.

(vi) The phase $\theta \in [-\pi, \pi]^N$ of the two-dimensional discrete Fourier transform of a noise-corrupted version of $\overline{x}$, i.e., $\theta = \angle \mathrm{DFT}(\overline{x} + w_{62})$, where $w_{62} \in \mathbb{R}^N$ is uniformly distributed in $[-3, 3]$. This information is enforced by forcing a candidate solution to lie in the closed convex set $C_{62} = \big\{x \in \mathbb{R}^N \mid \angle \mathrm{DFT}(x) = \theta\big\}$, i.e., by enforcing the constraint $x = \mathrm{proj}_{C_{62}} x$. This constraint corresponds to the model $r_{62} = F_{62}(L_{62}x)$, with $G_{62} = \mathbb{R}^N$, $L_{62} = \mathrm{Id}$, $r_{62} = 0$, and $F_{62} = \mathrm{Id} - \mathrm{proj}_{C_{62}}$, that is [43],

$$F_{62} \colon \mathbb{R}^N \to \mathbb{R}^N \colon x \mapsto x - \mathrm{IDFT}\Big(\big|\mathrm{DFT}\, x\big| \max\big\{\cos\big(\angle(\mathrm{DFT}\, x) - \theta\big), 0\big\} \exp(\imath\theta)\Big). \tag{5.19}$$

Due to the presence of the measurement errors $(w_k)_{1 \leqslant k \leqslant 60}$ and w_{62}, problem (5.12) is inconsistent and we approximate it by (5.15)–(5.16), where $\alpha_1 = \cdots = \alpha_{62} = 1$. To implement the algorithms of Sections 3.2–3.4, we require the expressions of the resolvent of the operators N_C and $(B_k)_{1 \leqslant k \leqslant p}$. The former is just

$$J_{N_C} = \mathrm{proj}_C \colon (\xi_j)_{1 \leqslant j \leqslant N} \mapsto \big(\min\{\max\{0, \xi_j\}, 255\}\big)_{1 \leqslant j \leqslant N}. \tag{5.20}$$

For the remaining cases, it follows from (5.15) that the operators $(B_k)_{1 \leqslant k \leqslant p}$ are firmly nonexpansive. We therefore invoke Lemma 2.1 to compute their resolvents. Let $\gamma \in \left]0, +\infty\right[$ and note that [6, Proposition 23.17(ii)] entails that

$$(\forall k \in \{1, \ldots, p\}) \quad J_{\gamma B_k} = J_{\gamma F_k}(\cdot + \gamma r_k). \tag{5.21}$$

First, set $k \in \{1, \ldots, 20\}$. Then $F_k = \mathrm{proj}_{C_1} = J_{N_{C_1}}$. Hence, upon setting $r_k = (\rho_{k,j})_{1 \leqslant j \leqslant N}$, we deduce from Lemma 2.1(ii) and (5.21) that

$$J_{\gamma B_k} \colon (\xi_j)_{1 \leqslant j \leqslant N} \mapsto \left(\xi_j + \gamma\rho_{k,j} - \gamma \min\left\{\max\left\{0, \frac{\xi_j + \gamma\rho_{k,j}}{1 + \gamma}\right\}, 60\right\}\right)_{1 \leqslant j \leqslant N}. \tag{5.22}$$

On the other hand, for $k \in \{21, \ldots, 60\}$, $J_{\gamma F_k} \colon (\eta_j)_{1 \leqslant j \leqslant N} \mapsto (\zeta_j)_{1 \leqslant j \leqslant N}$, where

$$(\forall j \in \{1, \ldots, N\}) \quad \zeta_j = \begin{cases} \dfrac{\eta_j - 90(1 + \gamma) + \sqrt{|\eta_j - 90(1 + \gamma)|^2 + 360\eta_j}}{2}, & \text{if } \eta_j \geqslant 0; \\ \eta_j, & \text{otherwise.} \end{cases} \tag{5.23}$$

Thus, we derive from (5.21) the expressions for $J_{\gamma B_k}$. Next, we have $J_{\gamma B_{61}} = (1 + \gamma)^{-1}(\cdot + \gamma N\rho)$ as a result of $J_{\gamma F_{61}} = (1 + \gamma)^{-1}\mathrm{Id}$ and (5.21). Finally, we deduce from [6, Proposition 23.20] that

$$F_{62} = \mathrm{Id} - \mathrm{proj}_{C_{62}} = J_{N_{C_{62}}^{-1}} \quad \text{and} \quad J_{(1+\gamma)^{-1}N_{C_{62}}^{-1}} \circ (1 + \gamma)^{-1}\mathrm{Id} = \frac{\mathrm{Id} - J_{N_{C_{62}}}}{1 + \gamma} = \frac{\mathrm{Id} - \mathrm{proj}_{C_{62}}}{1 + \gamma}. \tag{5.24}$$

Hence, it follows from Lemma 2.1(ii) that $J_{\gamma B_{62}} = J_{\gamma F_{62}} = (1 + \gamma)^{-1}(\mathrm{Id} + \gamma\, \mathrm{proj}_{C_{62}})$, i.e.,

$$J_{\gamma B_{62}} \colon y \mapsto \frac{y}{1 + \gamma} + \frac{\gamma}{1 + \gamma}\, \mathrm{IDFT}\Big(\big|\mathrm{DFT}\, y\big| \max\big\{\cos\big(\angle(\mathrm{DFT}\, y) - \theta\big), 0\big\} \exp(\imath\theta)\Big). \tag{5.25}$$

Finally, we implement the inversions of linear operators using the fast Fourier transform and Example 3.12(ii).

We employ the three frameworks of Sections 3.2–3.4 to solve (5.16), where Proposition 3.7 uses the operator **E** defined in Example 3.8. Two experiments are conducted: the random variable $\boldsymbol{\varepsilon}_0$ produces (a) 1 activation with 1 core; and (b) 8 activations with 8 cores. We compare with algorithm (2.13), where the random variable $\boldsymbol{\varepsilon}_0$ activates (a) 1; and (b) 8 indices in $\{1, \dots, p\}$ with a uniform distribution. The solution produced by Framework 3 is shown in Figure 7(e). We display in Figure 8 the normalized error versus execution time on a single-processor machine.

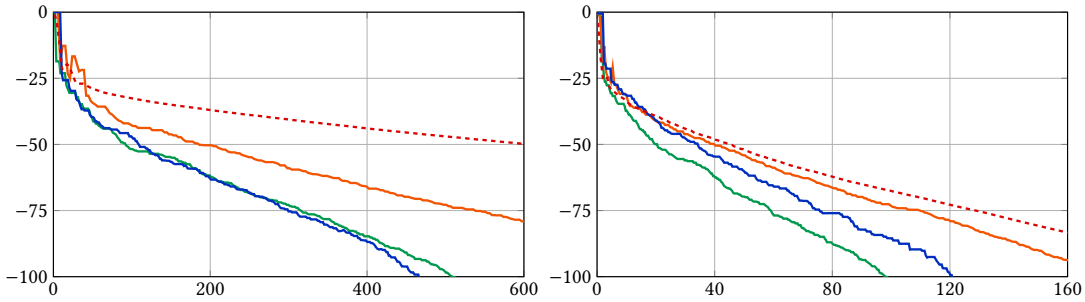


Figure 8: Experiment of Section 5.5: Normalized error $20 \log_{10}(\|x_{1,n} - x_\infty\| / \|x_{1,0} - x_\infty\|)$ (dB) versus execution time (s). (a): Block size 1 with 1 core. (b): Block size 8 with 8 cores. Green: Framework 1. Orange: Framework 2. Blue: Framework 3 with Example 3.8. Dashed red: Algorithm (2.13).

5.6. Discussion

The three proposed frameworks differ in terms of storage requirements, use of resolvent operators, and use of linear operators.

- Framework 1: It stores $2p + 3$ vectors. In addition, for each of the $p + 1$ random activation indices, there is one resolvent evaluation.
- Framework 2: It stores $4p + 5$ vectors. Out of the $p + 2$ random activation indices, those in $\{1, \dots, p + 1\}$ involve the evaluation of a resolvent. In addition, the linear operators are used only if index $p + 2$ is activated.
- Framework 3: It stores $2p + 2r + 2$ vectors. Moreover, out of the $p + r + 1$ random activation indices, those in $\{1, \dots, p + 1\}$ involve the evaluation of a resolvent operator, while those in $\{p + 2, \dots, p + r + 1\}$ do not require a resolvent evaluation.

Although Framework 1 is the most efficient in terms of storage, it may not always be the fastest, especially when resolvents are computationally expensive. For instance, in Section 5.5, where it is the case, Framework 3 is the fastest. Framework 2 has an advantage when the linear operators are costly, which is the case in Section 5.3. Finally, we observe that the existing algorithms (2.11) and (2.13) which, as discussed in Section 2.2, do not satisfy conditions **R2**–**R3**, are consistently slower than the methods proposed in Sections 3.2–3.4.

References

[1] A. Alacaoglu, O. Fercoq, and V. Cevher, On the convergence of stochastic primal-dual hybrid gradient, *SIAM J. Optim.*, vol. 32, pp. 1288–1318, 2022.

[2] H. C. Andrews and B. R. Hunt, *Digital Image Restoration.* Prentice-Hall, Englewood Cliffs, NJ, 1977.

[3] H. Attouch, L. M. Briceño–Arias, and P. L. Combettes, A strongly convergent primal-dual method for nonoverlapping domain decomposition, *Numer. Math.*, vol. 133, pp. 433–470, 2016.

[4] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, Optimization with sparsity-inducing penalties, *Found. Trends Machine Learn.*, vol. 4, pp. 1–106, 2012.

[5] S. Banert, A. Ringh, J. Adler, J. Karlsson, and O. Öktem, Data-driven nonsmooth optimization, *SIAM J. Optim.*, vol. 30, pp. 102–131, 2020.

[6] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, 2nd ed. Springer, New York, 2017.

[7] M. Benning and M. Burger, Modern regularization methods for inverse problems, *Acta Numer.*, vol. 27, pp 1–111, 2018.

[8] J. R. Blum, Multidimensional stochastic approximation methods, *Ann. Math. Stat.*, vol. 25, pp. 737–744, 1954.

[9] A. Chambolle, C. Delplancke, M. J. Ehrhardt, C.-B. Schönlieb, and J. Tang, Stochastic primal-dual hybrid gradient algorithm with adaptive step sizes, *J. Math. Imaging Vision*, vol. 66, pp. 294–313, 2024.

[10] A. Chambolle and T. Pock, An introduction to continuous optimization for imaging, *Acta Numer.*, vol. 25, pp. 161–319, 2016.

[11] P. L. Combettes, The geometry of monotone operator splitting methods, *Acta Numer.*, vol. 33, pp. 487–632, 2024.

[12] P. L. Combettes and J.-C. Pesquet, Proximal splitting methods in signal processing, in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pp. 185–212. Springer, New York, 2011.

[13] P. L. Combettes and J.-C. Pesquet, Stochastic quasi-Fejér block-coordinate fixed point iterations with random sweeping, *SIAM J. Optim.*, vol. 25, pp. 1221–1248, 2015.

[14] P. L. Combettes and J.-C. Pesquet, Deep neural network structures solving variational inequalities, *Set-Valued Var. Anal.*, vol. 28, pp. 491–518, 2020.

[15] P. L. Combettes and J.-C. Pesquet, Fixed point strategies in data science, *IEEE Trans. Signal Process.*, vol. 69, pp. 3878–3905, 2021.

[16] P. L. Combettes and V. R. Wajs, Signal recovery by proximal forward-backward splitting, *Multiscale Model. Simul.*, vol. 4, pp. 1168–1200, 2005.

[17] P. L. Combettes and Z. C. Woodstock, Reconstruction of functions from prescribed proximal points, *J. Approx. Theory*, vol. 268, art. 105606, 2021.

[18] P. L. Combettes and Z. C. Woodstock, A variational inequality model for the construction of signals from inconsistent nonlinear equations, *SIAM J. Imaging Sci.*, vol. 15, pp. 84–109, 2022.

[19] L. Condat and P. Richtárik, Randprox: Primal-dual optimization algorithms with randomized proximal updates, *Proc. Int. Conf. Learn. Represent.* Kigali, Rwanda, May 1–5, 2023.

[20] D. Davis, Variance reduction for root-finding problems, *Math. Program.*, vol. A197, pp. 375–410, 2023.

[21] A. Dieuleveut, G. Fort, E. Moulines, and H.-T. Wai, Stochastic approximation beyond gradient for signal processing and machine learning, *IEEE Trans. Signal Process.*, vol. 71, pp. 3117–3148, 2023.

[22] M. Duflo, *Méthodes Récursives Aléatoires.* Masson, Paris, 1990. English translation: *Random Iterative Models.* Springer, New York, 1997.

[23] L. Euler, *Recherches sur la Question des Inégalités du Mouvement de Saturne et de Jupiter*. Martin, Coignard et Guerin, Paris, 1749.

[24] F. Florez-Ospina, D. A. Jimenez-Sierra, H. D. Benitez-Restrepo, and G. R. Arce, Exploiting variational inequalities for generalized change detection on graphs, *IEEE Trans. Geosci. Remote Sensing*, vol. 61, pp. 1–16, 2023.

[25] R. Glowinski, S. J. Osher, and W. Yin (eds.), *Splitting Methods in Communication, Imaging, Science, and Engineering*. Springer, New York, 2016.

[26] T. Hytönen, J. van Neerven, M. Veraar, and L. Weis, *Analysis in Banach Spaces. Volume I: Martingales and Littlewood–Paley Theory*. Springer, New York, 2016.

[27] A. B. Juditsky and A. S. Nemirovski, Signal recovery by stochastic optimization, *Autom. Remote Control*, vol. 80, pp. 1878–1893, 2019.

[28] A. B. Juditsky and A. S. Nemirovski, *Statistical Inference via Convex Optimization.* Princeton University Press, Princeton, NJ, 2020.

[29] M. Ledoux and M. Talagrand, *Probability in Banach Spaces: Isoperimetry and Processes.* Springer, New York, 1991.

[30] Y. Malitsky and M. K. Tam, A forward-backward splitting method for monotone inclusions without cocoercivity, *SIAM J. Optim.*, vol. 30, pp. 1451–1472, 2020.

[31] J. T. Mayer, Abhandlung über die Umwälzung des Monds um seine Axe und die scheinbare Bewegung der Mondflecken, *Kosmographische Nachrichten und Sammlungen*, vol. 1, pp. 52–183, 1748.

[32] D. W. Mimouni, P. Malisani, J. Zhu, and W. de Oliveira, Computing Wasserstein barycenters via operator splitting: The method of averaged marginals, *SIAM J. Data Sci.*, vol. 6, pp. 1000–1026, 2024.

[33] Z. Peng, Y. Xu, M. Yan, and W. Yin, ARock: An algorithmic framework for asynchronous parallel coordinate updates, *SIAM J. Sci. Comput.*, vol. 38, pp. A2851–A2879, 2016.

[34] J.-C. Pesquet and A. Repetti, A class of randomized primal-dual algorithms for distributed optimization, *J. Nonlinear Convex Anal.*, vol. 16, pp. 2453–2490, 2015.

[35] J.-C. Pesquet, A. Repetti, M. Terris, and Y. Wiaux, Learning maximally monotone operators for image recovery, *SIAM J. Imaging Sci.*, vol. 14, pp. 1206–1237, 2021.

[36] B. J. Pettis, On integration in vector spaces, *Trans. Amer. Math. Soc.*, vol. 44, pp. 277–304, 1938.

[37] H. Robbins and S. Monro, A stochastic approximation method, *Ann. Math. Statist.*, vol. 22, pp. 400–407, 1951.

[38] A. Sadiev, L. Condat, and P. Richtárik, Stochastic proximal point methods for monotone inclusions under expected similarity, arxiv, 2024. https://arxiv.org/pdf/2405.14255

[39] S. Theodoridis, *Machine Learning: A Bayesian and Optimization Perspective*, 2nd ed. Amsterdam: Elsevier, 2020.

[40] C. Traoré, V. Apidopoulos, S. Salzo, and S. Villa, Variance reduction techniques for stochastic proximal point algorithms, *J. Optim. Theory Appl.*, vol. 203, pp. 1910–1939, 2024.

[41] E. Winston and J. Z. Kolter, Monotone operator equilibrium networks, *Proc. Adv. Neural Inform. Process. Syst.*, vol. 22, pp. 10718–10728, 2020.

[42] P. Yi and S. Ching, Synthesis of recurrent neural dynamics for monotone inclusion with application to Bayesian inference, *Neural Networks*, vol. 131, pp. 231–241, 2020.

[43] D. C. Youla, Mathematical theory of image restoration by the method of convex projections, in: H. Stark (ed.) *Image Recovery: Theory and Application*, pp. 29–77. Academic Press, San Diego, CA, 1987.

[44] Y.-L. Yu, Better approximation and faster algorithm using the proximal average, *Proc. Conf. Adv. Neural Inform. Process. Syst.*, pp. 458–466, 2013.