Journal of Machine Learning Research 23 (2022) 1-44 Submitted 6/19; Revised 2/22; Published 3/22

A Unifying Framework for Variance-Reduced Algorithms for Findings Zeroes of Monotone Operators

Xun Zhang XUNZHANG@U.NUS.EDU
Department of Industrial and Systems Engineering
National University of Singapore
21 Lower Kent Ridge Rd, Singapore

William B. Haskell WBHASKELL@GMAIL.COM
Krannert School of Management
Purdue University
West Lafayette, IN 47907

Zhisheng Ye YEZ@NUS.EDU.SG
Department of Industrial and Systems Engineering
National University of Singapore
21 Lower Kent Ridge Rd, Singapore

Editor: Sathiya Keerthi

Abstract

It is common to encounter large-scale monotone inclusion problems where the objective has a finite sum structure. We develop a general framework for variance-reduced forward-backward splitting algorithms for this problem. This framework includes a number of existing deterministic and variance-reduced algorithms for function minimization as special cases, and it is also applicable to more general problems such as saddle-point problems and variational inequalities. With a carefully constructed Lyapunov function, we show that the algorithms covered by our framework enjoy a linear convergence rate in expectation under mild assumptions. We further consider Catalyst acceleration and asynchronous implementation to reduce the algorithmic complexity and computation time. We apply our proposed framework to a policy evaluation problem and a strongly monotone two player game, both of which fall outside the realm of function minimization.

Keywords: finite sum minimization, monotone inclusion, monotone operators, variance-reduced algorithms, Catalyst acceleration, asynchronous computing

1. Introduction

In the field of convex optimization, many problems can be recast as *monotone inclusion* problems, where the goal is to find zeros of appropriate set-valued mappings called *monotone operators* (Bauschke and Combettes, 2011; Ryu and Boyd, 2016). Some typical examples include convex function minimization (Boyd and Vandenberghe, 2004), convex-concave saddle-point problems (Ryu and Boyd, 2016), and multi-agent decision making (Harker and Pang, 1990), among others.

In many cases, these monotone inclusion problems have a finite sum structure. For example, finite sum minimization often arises in machine learning and statistics where we

minimize the empirical risk (that is, the average of individual loss functions corresponding to individual observations) (Cortes and Vapnik, 1995; Dumais et al., 1998). In addition, using the "factored" technique (Balamurugan and Bach, 2016), the saddle-differential operators of bilinear saddle-point problems can also be split into a finite sum. Furthermore, the game mappings of many multi-agent problems are expectations over a set of scenarios (Kattsoff, 1945). Sample average approximation (SAA) can be applied to get a numerical solution, and the resulting approximation has a finite sum structure.

As a special case of the finite sum monotone inclusion problem, finite sum function minimization is ubiquitous in machine learning. The most well known algorithm for finite sum minimization is full gradient descent (GD) (Nesterov, 2014), which has a linear convergence rate but is computationally expensive for large-scale problems. In view of this weakness, stochastic gradient descent (SGD) was proposed in Robbins and Monro (1951). SGD only computes the gradient of one function in each iteration, and thus it requires much less computational power per-iteration compared to GD. Given the same number of gradient evaluations, it has been shown both theoretically and empirically that SGD is able to achieve a lower empirical loss compared to GD, especially in early iterations (Nocedal, 1980; Liu and Nocedal, 1989; Bubeck, 2015).

Although SGD has many advantages over GD, an intrinsic disadvantage of SGD is that it only has a sublinear convergence rate in expectation (Robbins and Monro, 1951). To improve upon the performance of SGD while retaining its advantages, a large class of variance-reduced algorithms has been proposed. This class of algorithms includes stochastic variance-reduced gradient (SVRG) (Johnson and Zhang, 2013; Xiao and Zhang, 2014), SAGA (Defazio et al., 2014), hybrid stochastic average gradient (HSAG) (Reddi et al., 2015), and stochastic average gradient (SAG) (Nicolas et al., 2012; Schmidt et al., 2017), among others. The core idea of these algorithms is to replace the random gradient estimator in SGD with a dedicated variance-reduced estimator. All of these algorithms achieve a linear convergence rate in expectation and still inherit the low per-iteration cost of SGD. Many numerical experiments and real world applications have demonstrated that variance-reduced algorithms are able to achieve a lower empirical loss than both GD and SGD given the same computational budget (Nicolas et al., 2012; Johnson and Zhang, 2013; Defazio et al., 2014; Xiao and Zhang, 2014; Reddi et al., 2015; Schmidt et al., 2017).

In view of the effectiveness of variance-reduced algorithms for finite sum function minimization, a natural idea is to use similar algorithms to solve the more general finite sum monotone inclusion problem. In the original papers (Johnson and Zhang, 2013; Defazio et al., 2014; Xiao and Zhang, 2014; Reddi et al., 2015), the convergence analyses of these variance-reduced algorithms (for function minimization) are all done in a case by case manner. The main objective of this study is to bridge this gap by providing a unifying convergence analysis of a class of variance-reduced algorithms for finite sum monotone inclusion problems. It is known that GD and SGD can be viewed as special cases of the deterministic/randomized forward-backward (FB) splitting method for solving monotone inclusion problems, respectively (Ryu and Boyd, 2016; Rosasco et al., 2016). In the existing literature, Balamurugan and Bach (2016) established the linear convergence of SVRG and SAGA for saddle-point problems. Our present work builds on Balamurugan and Bach (2016) by unifying and generalizing their convergence analysis to other variance-reduced algorithms for solving finite sum monotone inclusion problems.

In our framework, we maintain a proxy for each of the individual operators in the forward step of the randomized FB splitting. These proxies are then used to construct an unbiased estimator of the exact operator in the forward step. By carefully designing the proxies and the estimator, our framework is able to recover many existing algorithms for finite sum minimization such as GD, SGD, SAGA, SVRG, and HSAG. We show that after each iteration, the expected distance to the optimum shrinks geometrically subject to a disturbance which depends on the variance of the estimator. By controlling the variance of the proxies, we are able to construct a Lyapunov function which consists of the distance to the optimum plus an error term determined by the value of the proxies, and we show that this Lyapunov function decreases geometrically in expectation. This result is then applied to GD, SAGA, HSAG, and SVRG with possibly varying epoch lengths. Further, this framework allows us to propose some new variance-reduced algorithms within our framework that have a linear convergence rate in expectation. As an example, we propose a new algorithm called SVRG-rand which randomly does a full operator evaluation, instead of following a fixed epoch schedule. Numerical studies reveal that SVRG-rand performs better than both classical SVRG and SVRG with increasing epochs.

To further accelerate these variance-reduced algorithms, we extend Catalyst in Lin et al. (2015) for finite sum minimization problems to our variance-reduced FB splitting monotone inclusion framework. We show that in the "ill-conditioned" setting, many algorithms covered by our general randomized FB splitting framework attain a lower complexity after Catalyst acceleration. Apart from Catalyst acceleration, we also consider parallel computing to reduce the computation time. We find that our proposed framework can be extended to asynchronous versions of variance-reduced randomized forward-step algorithms. In the asynchronous case, in contrast to the synchronous case, we have a multicore structure for parallel computing where each of the cores may not have up-to-date information (Mania et al., 2017). Under similar assumptions on the proxies as for synchronous algorithms, we show that the iterates generated by asynchronous variance-reduced algorithms converge linearly in expectation to within a tolerance of the optimum. Specifically, our results apply to asynchronous variants of SVRG, SAGA, HSAG, and SVRG-rand. In contrast to previous work on asynchronous algorithms which required sparsity assumptions (Niu et al., 2011; **?**; Mania et al., 2017), our results hold more generally. In particular, the sparsity assumption is only applicable to finite sum minimization but does not hold for monotone inclusion problems (see Du et al. (2017)).

This paper is organized as follows. Section 2 introduces the notation and basic concepts of monotone operators and our problem setting. Section 3 presents a unifying framework for randomized FB splitting that recovers many classical algorithms. Here we also provide the convergence analysis for our general framework. Section 4 presents Catalyst acceleration for monotone inclusion problems. The asynchronous extension of our general algorithmic framework is investigated in Section 5. Then in Section 6, we apply the framework to existing algorithms and propose new algorithms. Section 7 provides a detailed comparison of the numerical performance for some of the classical algorithms as well as our new algorithms. We conclude the paper with a discussion of our broader themes and possible future research topics. All proofs are organized together in the Appendix.

2. Preliminaries

We first define key concepts from the monotone operator literature. Let $\|\cdot\|$ denote the Euclidean norm on $\mathbb{R}^d$ throughout.

Definition 1 *(i) An* operator $F : \mathbb{R}^d \to \mathbb{R}^d$ *is a subset of* $\mathbb{R}^d \times \mathbb{R}^d$*, where* $F(x) = \{ u \in \mathbb{R}^d \mid (x, u) \in F \}$.

(ii) An operator F *is* monotone *if* $(u - v)^\top (x - y) \geq 0$ *for all* $u \in F(x)$ *and* $v \in F(y)$.

(iii) An operator F *is* μ-strongly monotone *for* $\mu > 0$ *if* $(u - v)^\top (x - y) \geq \mu \|x - y\|^2$ *for all* $u \in F(x)$ *and* $v \in F(y)$.

(iv) An operator is maximal monotone *if it is monotone and there is no monotone operator that properly contains it.*

(v) An operator F *on* $\mathbb{R}^d$ *is* L-Lipschitz *if* $\|u - v\| \leq L\|x - y\|$ *for all* $u \in F(x)$ *and* $v \in F(y)$.

If F is Lipschitz, then it is single-valued and we may write $F(x) = y$ (Ryu and Boyd, 2016). When $L < 1$, F is a *contraction*; when $L = 1$, F is *non-expansive*.

Now we introduce our problem setting. Suppose we have two maximal monotone operators A and B on $\mathbb{R}^d$, where $B = \frac{1}{n}\sum_{i=1}^n B_i$ is the average of a collection of operators $\{B_i\}_{i=1}^n$ (indexed by $[n] \triangleq \{1, \ldots, n\}$). The monotone inclusion problem is to find $x \in \mathbb{R}^d$ such that:

$$0 \in A(x) + \frac{1}{n}\sum_{i=1}^n B_i(x). \tag{1}$$

We introduce the following assumptions on the operators A and $\{B_i\}_{i\in[n]}$.

Assumption 1 *(i)* A *is maximal monotone on* $\mathbb{R}^d$.

(ii) $B = \frac{1}{n}\sum_{i=1}^n B_i$ *is* μ*-strongly maximal monotone on* $\mathbb{R}^d$*, and each* B_i *is* L*-Lipschitz.*

Our assumptions are slightly different from those in Balamurugan and Bach (2016). We assume strong monotonicity for B and Lipschitz continuity for all B_i to align our work with the literature on finite sum minimization.

By Assumption 1, $A + B$ is strongly monotone and so Problem (1) has a unique solution $x^* \in \mathbb{R}^d$. Any $x_\epsilon^* \in \mathbb{R}^d$ such that $\|x_\epsilon^* - x^*\|^2 \leq \epsilon$ is called an ϵ-*solution* of Problem (1). The condition number of Problem (1) is $\kappa \triangleq L/\mu$, it will appear frequently in our complexity bounds.

Problem (1) is general and accommodates many problems in optimization as seen in the following examples.

Example 1 (Finite sum minimization) *Let* $f_i : \mathbb{R}^d \to \mathbb{R}$ *for* $i \in [n]$ *be convex,* $h : \mathbb{R}^d \to \mathbb{R}$ *be convex, and* $C \subset \mathbb{R}^d$ *be a convex set. The finite sum minimization problem is:*

$$\min_{x\in C} \; f(x) + h(x) \text{ where } f(x) \triangleq \frac{1}{n}\sum_{i=1}^n f_i(x). \tag{2}$$

The function f *in (2) is usually assumed to be both strongly convex (corresponding to strong monotonicity of its gradient) and strongly smooth (corresponding to Lipschitz continuity of*

its gradient). Let ∂ be the subdifferential, and I_C be the indicator function of the set C. Then, Problem (2) is equivalent to (1) where $A = \partial h(x) + \partial I_C(x)$ and $B_i = \partial f_i(x)$ for all $i \in [n]$.

Example 2 (Finite sum saddle-point problems) *Let $f_i : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ for $i \in [n]$ be convex/concave, $h : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ be convex/concave, and $C_1 \subset \mathbb{R}^{d_1}$ and $C_2 \subset \mathbb{R}^{d_2}$ be convex sets. The finite sum saddle-point problem is:*

$$\min_{x \in C_1} \max_{y \in C_2} f(x,\, y) + h(x,\, y) \text{ where } f(x,\, y) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i\,(x,\, y)\,. \tag{3}$$

Problem (3) is equivalent to (1) where $A = (\partial_x h(x,\, y), -\partial_y h(x,\, y)) + (\partial I_{C_1}(x), -\partial I_{C_2}(y))$ and $B_i = (\partial_x f_i(x,\, y), -\partial_y f_i(x,\, y))$ for all $i \in [n]$.

Example 3 (Finite sum variational inequalities) *Let $B_i : \mathbb{R}^d \to \mathbb{R}^d$ for $i \in [n]$ be monotone, and let $C \subset \mathbb{R}^d$ be a convex set. The finite sum variational inequality is to find $x^* \in \mathbb{R}^d$ such that:*

$$B(x^*)^\top (y - x^*) \geq 0,\, \forall y \in C. \tag{4}$$

Let N_C be the normal cone to the set C. Then, Problem (4) is equivalent to (1) where $A = N_C$ and B_i is as given for all $i \in [n]$.

The most classical algorithm for solving (1) is forward-backward (FB) splitting (Passty, 1979; Bauschke and Combettes, 2011) given by:

$$x_{k+1} \quad = \quad (I + \gamma\, A)^{-1}\,(I - \gamma\, B)\,(x_k),\, \forall k \geq 0, \tag{5}$$

where $\gamma > 0$ is the step-size. The following theorem demonstrates the linear convergence of the sequence produced by (5) in terms of the distance to the solution of (1).

Theorem 2 *Suppose Assumption 1 holds and let $\{x_k\}_{k \geq 0}$ be the sequence produced by (5). Then, for all $k \geq 0$ we have*

$$\|x_{k+1} - x^*\|^2 \leq \left(1 - 2\,\gamma\,\mu + \gamma^2 L^2\right) \|x_k - x^*\|^2\,.$$

When $\gamma \in (0,\, 2\,\mu/L)$, $\{x_k\}_{k \geq 0}$ converges linearly to the unique solution x^* of (1).

3. Variance-reduced forward-backward splitting

In this section, we present a framework for variance-reduced FB splitting for solving (1). In classical FB splitting, we need to compute $B(x_k)$ exactly in every iteration $k \geq 0$, but this step requires computation of $B_i(x_k)$ for every $i \in [n]$. Our framework is based on introducing a proxy variable for every operator $\{B_i\}_{i \in [n]}$ which records the most recent evaluation. We use the proxies to construct an estimator for $B(x_k)$ and then strategically update the proxies to avoid evaluating all $\{B_i\}_{i \in [n]}$ in every iteration. This strategy is generally true for variance-reduced first-order methods for optimization.

In iteration $k \geq 0$, we denote the proxies by $\phi^k = (\phi_i^k)_{i\in[n]}$ for all $k \geq 0$, where ϕ_i^k is the proxy for the most recent evaluation of B_i. We also introduce a sequence of i.i.d. discrete uniform random variables $\{I_k\}_{k\geq 0}$ with support on $[n]$. Then, we define the estimators:

$$\mathcal{G}(x_k,\, \phi^k, I_k) \triangleq B_{I_k}(x_k) - \phi_{I_k}^k + \frac{1}{n}\sum_{i=1}^{n} \phi_i^k,\ \forall k \geq 0, \tag{6}$$

which we use to estimate $B(x_k)$. The variance-reduced FB splitting method is then

$$x_{k+1} = (I + \gamma_k A)^{-1}\Big(I - \gamma_k\, \mathcal{G}(x_k,\, \phi^k,\, I_k)\Big),\ \forall k \geq 0, \tag{7}$$

where $\{\gamma_k\}_{k\geq 0}$ is a sequence of positive step-sizes. The proxies are updated according to some scheme which we denote by:

$$\phi^{k+1} = \mathcal{U}_k\Big(x_k,\, \phi^k,\, I_k\Big),\ \forall k \geq 0. \tag{8}$$

The rule for updating the proxies ϕ^k depends on the specific algorithm. This update rule is allowed to depend on the iteration number to accommodate epoch-based algorithms like SVRG. A complete description of the variance-reduced FB splitting is given in Algorithm 1. This algorithm is essentially determined by the update rule $\{\mathcal{U}_k\}_{k\geq 0}$ in (8) since the gradient estimator (6) is the same in each iteration.

Let $\{\mathcal{F}_k\}_{k\geq 0}$ be a filtration where $\mathcal{F}_k$ denotes the history of the algorithm up to the kth iteration. Because I_k is uniform, it is readily seen that $\mathcal{G}\big(x_k,\, \phi^k, I_k\big)$ is an unbiased estimator of $B(x_k)$ for any proxy updating scheme.

Lemma 3 *For all $k \geq 0$, $\mathbb{E}\left[\mathcal{G}\big(x_k,\, \phi^k, I_k\big)|\mathcal{F}_k\right] = B(x_k)$.*

Algorithm 1 Variance-reduced forward-backward splitting

Input:

Initial values: $x_0 \in \mathbb{R}^d$ and $\phi_i^0 \in \mathbb{R}^d$ for all $i \in [n]$, total number of iterations T.

1: **for** each $k = 0, 1, \ldots, T$ **do**
2: Generate I_k uniformly in $[n]$.
3: $x_{k+1} = (I + \gamma_k A)^{-1}\big(x_k - \gamma_k\, \mathcal{G}\big(x_k,\, \phi^k, I_k\big)\big)$.
4: $\phi^{k+1} = \mathcal{U}_k(x_k,\, \phi^k,\, I_k)$.
5: **end for**

Output:

6: x_{T+1}

We start our analysis by showing how the expected distance to the optimal solution x^* changes after one iteration. This upcoming inequality determines the convergence rate of Algorithm 1 and shows how it depends on the variance of the estimator $\mathcal{G}\big(x_k,\, \phi^k, I_k\big)$.

Lemma 4 *Suppose Assumption 1 holds and let $\{x_k\}_{k\geq 0}$ be produced by Algorithm 1. Then, for all $k \geq 0$ we have*

$$\mathbb{E}\,\|x_{k+1} - x^*\|^2 \leq \big(1 - 2\gamma_k\,\mu + \gamma_k^2 L^2\big)\,\mathbb{E}\,\|x_k - x^*\|^2 + \gamma_k^2 \mathbb{E}\left\|\mathcal{G}\big(x_k,\, \phi^k, I_k\big) - B(x_k)\right\|^2. \tag{9}$$

In the next lemma, we bound the conditional variance of $\mathcal{G}\big(x_k,\, \phi^k, I_k\big)$.

Lemma 5 *Suppose Assumption 1 holds and let $\{x_k\}_{k\geq 0}$ be generated by Algorithm 1. Then, for all $k \geq 0$ we have*

$$\mathbb{E}\left[\|\mathcal{G}\big(x_k,\, \phi^k, I_k\big) - B(x_k)\|^2 | \mathcal{F}_k\right] \leq 2\left(L^2\,\mathbb{E}\,\|x_k - x^*\|^2 + \mathbb{E}\frac{1}{n}\sum_{i=1}^{n}\|\phi_i^k - B_i(x^*)\|^2\right).$$

The following corollary is immediate from the bounds in Lemmas 4 and 5.

Corollary 6 *Suppose Assumption 1 holds and let $\{x_k\}_{k\geq 0}$ be generated by Algorithm 1. Then, for all $k \geq 0$ we have*

$$\mathbb{E}\,\|x_{k+1} - x^*\|^2 \leq (1 - 2\gamma_k\mu + 3\gamma_k^2 L^2)\mathbb{E}\,\|x_k - x^*\|^2 + \frac{2\gamma_k^2}{n}\mathbb{E}\sum_{i=1}^{n}\|\phi_i^k - B_i(x^*)\|^2.$$

Suppose $1 - 2\,\gamma_k\mu + 3\,\gamma_k^2 L^2 < 1$ for all $k \geq 0$. Then, in each iteration, the expected distance to x^* is upper bounded by a factor (less than one) of the current distance to x^*, plus an additional disturbance. The additional disturbance depends on how close the proxies are to the associated operator values $\{B_i(x^*)\}_{i\in[n]}$ at x^*. Different algorithms give rise to different disturbances (because they update the proxies differently), and thus yield different convergence rates.

To allow for epoch-based algorithms, we introduce the sequence $\{m_i\}_{i\geq 1}$ of epoch lengths where m_i is the length of epoch i and all $m_i \geq 1$. We also introduce the sequence $\{S_i\}_{i\geq 0}$ with $S_0 = 0$ and $S_i = \sum_{j=1}^{i} m_j$ for all $i \geq 1$, to denote the starting indices of each epoch. We then let

$$\tilde{x}_k \triangleq x_{S_k}, \quad \tilde{\phi}^k \triangleq \phi^{S_k}, \quad \forall k \geq 0,$$

denote the iterates obtained at the end of each epoch. For non-epoch based algorithms, we have $m_i = 1$ and $S_i = i$ for all $i \geq 1$.

In epoch-based algorithms, some proxies are only updated at the end of an epoch. To model this scheme, we divide the index set $[n]$ into a set $\mathcal{S} \subseteq [n]$ and its complement $\mathcal{S}^c$. The proxies in $\mathcal{S}$ follow an algorithm-specific update, while the proxies in $\mathcal{S}^c$ are only updated at the end of each epoch.

Now we construct a Lyapunov function to analyze Algorithm 1. First, we define

$$\begin{aligned} G(\phi^k) &\triangleq \frac{1}{n}\sum_{i\in\mathcal{S}}\|\phi_i^k - B_i(x^*)\|^2,\ \forall k \geq 0, \\ H(\phi^k) &\triangleq \frac{1}{n}\sum_{i\notin\mathcal{S}}\|\phi_i^k - B_i(x^*)\|^2,\ \forall k \geq 0, \end{aligned}$$

where we stipulate that $G(\phi^k) \equiv 0$ if $\mathcal{S} = \emptyset$ and $H(\phi^k) \equiv 0$ if $\mathcal{S}^c = \emptyset$. It is readily seen that $G(\phi^k) + H(\phi^k) = \frac{1}{n}\sum_{i=1}^{n}\|\phi_i^k - B_i(x^*)\|^2$, so we have split all the proxies into two groups which follow different update schemes. For example, in HSAG (Reddi et al., 2015),

G and H will capture the proxies that follow SAGA-type updates and SVRG-type updates, respectively. Now for $\rho \geq 0$, we define the Lyapunov function

$$L_\rho\left(x_k, \phi^k\right) \triangleq \|x_k - x^*\|^2 + \rho\, G(\phi^k).$$

Our analysis will focus on $L_\rho(\tilde{x}_k, \tilde{\phi}^k)$, the value of the Lyapunov function at the end of each epoch.

Next we introduce our assumptions on G and H. Under these assumptions, we can show that the sequence $\{L_\rho(\tilde{x}_k, \tilde{\phi}^k)\}_{k\geq 0}$ converges geometrically to zero in expectation which gives the linear convergence rate of Algorithm 1.

Assumption 2

(2.1) There exist constants $0 \leq c_1 < 1$ and $c_2 \geq 0$ such that $\mathbb{E}G(\phi^{k+1}) \leq c_1 \mathbb{E}G(\phi^k) + c_2 \mathbb{E}\|x_k - x^*\|^2$.

(2.2) There exists a constant $c_3 \geq 0$ such that $H(\phi^k) \leq c_3 L_\rho(x_{S_i}, \phi^{S_i})$ for all $S_i \leq k < S_{i+1}$. Furthermore, $\bar{m} \triangleq \sup_{i\geq 1} m_i < \infty$.

(2.3) The step-size is constant, that is, $\gamma_k = \gamma$ for all $k \geq 0$.

We will show that these assumptions are sufficient for Algorithm 1 to have a linear convergence rate. Assumption 2.1 corresponds to the proxies in $\mathcal{S}$ that are updated in every iteration and Assumption 2.2 corresponds to the proxies in $\mathcal{S}^c$ that are updated only at the end of each epoch. When $\mathcal{S} = \emptyset$, Assumption 2.1 holds automatically since $G(\phi^k) \equiv 0$. On the other hand, when $\mathcal{S} = [n]$ then Assumption 2.2 holds automatically since $H(\phi^k) \equiv 0$.

Assumption 2.1 requires $G(\phi^k)$ (the sum of the errors for proxies in $\mathcal{S}$) to contract by a factor less than one, plus an additional disturbance which depends on $\|x_k - x^*\|$. This is essentially a condition on the proxy update scheme (8) to ensure that the proxies are updated "frequently enough".

Assumption 2.2 requires $H(\phi^k)$ to be uniformly bounded over each epoch, where the upper bound depends on the value of the Lyapunov function at the beginning of the epoch (multiplied by a constant). When proxies captured by $H(\phi^k)$ are updated only at $k = S_k$, then $H(\phi^k)$ is unchanged for all iterations $S_i \leq k < S_{i+1}$. Furthermore, $H(\phi^k)$ usually depends on the distance to x^* which can be further bounded by the value of the Lyapunov function.

Next we present our main result on the linear convergence of Algorithm 1. We define the following two constants:

$$\theta \triangleq \max\left\{1 - 2\gamma\mu + 3\gamma^2 L^2 + c_2\rho,\ \frac{2\gamma^2}{\rho} + c_1\right\}, \tag{10}$$

$$\lambda \triangleq \theta + 2\gamma^2 \bar{m} c_3. \tag{11}$$

We also introduce the following step-size rule:

$$\gamma < \min\left\{\left(\frac{2\mu}{\frac{3(1-c_1)L^2}{2} + (1-c_1)c_3\bar{m} + c_2}\right)^2,\ \left(\frac{1-c_1}{2 + 2\bar{m}\left(\frac{1-c_1}{2}\right)^3 c_3}\right)^2\right\}. \tag{12}$$

Theorem 7 *Suppose Assumptions 1 and 2 hold, the step-size γ satisfies (12), and let $\{x_k\}_{k\geq 0}$ be generated by Algorithm 1. Then for $\rho = \gamma^{1.5}$, we have $\lambda \in [0,1)$ and*

$$\mathbb{E}L_\rho\left(\tilde{x}_k, \tilde{\phi}^k\right) \leq \lambda\, \mathbb{E}L_\rho\left(\tilde{x}_{k-1}, \tilde{\phi}^{k-1}\right), \forall k \geq 1. \tag{13}$$

Furthermore,

$$\mathbb{E}\|\tilde{x}_k - x^*\|^2 \leq \lambda^k(1+\rho L^2)\|x_0 - x^*\|^2, \forall k \geq 1. \tag{14}$$

Theorem 7 is based on a generic Lyapunov function that can be used for many variance-reduced algorithms. In contrast, the Lyapunov functions and the corresponding convergence analysis that appear in Johnson and Zhang (2013), Defazio et al. (2014), Xiao and Zhang (2014), and Reddi et al. (2015), are all designed for the specific algorithms at hand. We note that $\rho = \gamma^{1.5}$ is not the only allowable choice for the parameter of the Lyapunov function. Using very similar arguments as Theorem 7, we can extend the choice $\rho = \gamma^{1.5}$ to $\rho = \gamma^t$ for $1 < t < 2$ and still retain linear convergence. However, the expression for the resulting step-size rule is more complicated.

4. Acceleration by Catalyst

Catalyst proposed by Lin et al. (2015) is a well-known method for acceleration of first-order algorithms for finite sum minimization problems. To accelerate a generic first-order algorithm $\mathcal{M}$ for Problem (2), Lin et al. (2015) uses a method based on an inner-outer loop structure. In the inner loop, the algorithm $\mathcal{M}$ is called to solve an auxiliary problem where a quadratic regularizer is added to the original objective function to increase its strong convexity. The inner loop terminates when the auxiliary problem is solved by $\mathcal{M}$ up to a pre-determined error tolerance. Then, in the outer loop, the iterate generated by the inner loop is used to define a new regularizer for the auxiliary problem of the next inner loop. For "ill-conditioned" problems where the condition number $\kappa = L/\mu$ is large compared to n, Catalyst is able to accelerate a large class of finite sum minimization algorithms.

In this section, we show that the regularization technique of Catalyst can also accelerate Algorithm 1 for solving (1). To this end, we adopt the inner-outer loop structure from Lin et al. (2015). In the inner loop, we call an algorithm $\mathcal{M}$ to solve an auxiliary problem where an affine operator that mimics the role of the regularizer is added to the RHS of (1). Then, in the outer loop, we use the iterate generated by the inner loop to update the auxiliary problem. In the following, we will compare the overall complexity of variance-reduced FB splitting with and without Catalyst acceleration.

Let $\mathcal{M}(A,B)$ denote a generic first-order algorithm for solving (1). We assume $\mathcal{M}(A,B)$ has a linear convergence rate, that is, there exists a constant $\pi_{\mathcal{M}(A,B)} > 0$ such that $\mathcal{M}(A,B)$ requires at most $O(\pi_{\mathcal{M}(A,B)}\log(1/\epsilon))$ operator evaluations to find an ϵ-solution satisfying $\mathbb{E}\|x_\epsilon^* - x^*\|^2 \leq \epsilon$. The constant $\pi_{\mathcal{M}(A,B)}$ usually depends on the condition number $\kappa = L/\mu$. When extended to the monotone operator setting, it can be readily shown that popular algorithms like SVRG, SAGA, and HSAG, all have complexity $O\left(n\log\left(1/\epsilon\right)\right)$ when the problem is well-conditioned and $\kappa^2 \leq n$ (see Table 2). This complexity is already optimal (Lin et al., 2015; Agarwal and Bottou, 2015). We focus on the "ill-conditioned" case where $\kappa^2 \geq n$ and the complexity is dominated by κ^2.

Algorithm 2 Catalyst

Input:

An initial value $\check{x}_0 \in \mathbb{R}^d$, the total number of inner-outer loops T, and parameter σ.

for each $k = 0, 1, \ldots, T$ **do**
  set $\bar{x} = \check{x}_k$ and consider Problem (15). Set the index for inner-loop $s = 0$ and initialize the inner-loop with $x_0 = \bar{x}$.
  while the desired stopping criterion is not satisfied **do**
    Starting with x_0, use $\mathcal{M}(A,\, B + \sigma(I - \bar{x}))$ to generate the sequence $\{x_s\}_{s\geq 0}$.
    If (16) holds for some x_s, then stop.
  end while
  $\check{x}_{k+1} = x_s$.
end for

Output:

$\check{x}_{T+1}$

The details of Catalyst for solving (1) are shown in Algorithm 2. We let $\{\check{x}_k\}_{k\geq 0}$ be the sequence of iterates at the beginning of each outer loop in Catalyst. In the kth outer loop we initialize with $\bar{x} = \check{x}_k$. Then, in the inner loop we call $\mathcal{M}\big(A,\, B + \sigma(I - \bar{x})\big)$ to solve the auxiliary monotone inclusion problem:

$$0 \in A(x) + B(x) + \sigma(x - \bar{x}), \tag{15}$$

where $\sigma \geq 0$ is a regularization parameter to be specified later. In (15), the operator $B + \sigma(I - \bar{x})$ is $(L + \sigma)$-Lipschitz and $(\mu + \sigma)$-strongly monotone, while the original operator B is L-Lipschitz and μ-strongly monotone. Catalyst runs $\mathcal{M}(A,\, B + \sigma(I - \bar{x}))$ to solve (15) starting with $x_0 = \bar{x}$ and generates a sequence $\{x_s\}_{s\geq 0}$, until we find an iterate x_s such that

$$\mathbb{E}\|x_s - x^*(\bar{x})\|^2 \leq \frac{\mathbb{E}\|x_0 - x^*(\bar{x})\|^2}{4(1 + \sigma/\mu)^2}, \tag{16}$$

where $x^*(\bar{x})$ is the unique solution of Problem (15). Then, Catalyst takes $\check{x}_{k+1} = x_s$ and starts the next outer loop.

The following lemma establishes the linear convergence of Catalyst across outer loops.

Lemma 8 *Suppose Assumption 1 holds and let $\{\check{x}_k\}_{k\geq 1}$ be generated by Algorithm 2. Then, for all $k \geq 0$ we have*

$$\mathbb{E}\|\check{x}_{k+1} - x^*\|^2 \leq \left(1 - \frac{1}{2(1 + \sigma/\mu)}\right) \mathbb{E}\|\check{x}_k - x^*\|^2.$$

We know that $(1 - 1/x)^x \approx 1/e$ for $x \gg 0$. The parameter σ depends on κ and is often large for ill-conditioned problems (see Section 6). It follows that Catalyst needs $O(\frac{2(\mu+\sigma)}{\mu} \log(1/\epsilon))$ outer loops to compute an $\check{x}_k$ such that $\mathbb{E}\|\check{x}_k - x^*\|^2 \leq \epsilon$. For each outer loop, we call $\mathcal{M}(A,\, B + \sigma(I - \bar{x}))$ to obtain a solution of accuracy $\frac{1}{4(1+\sigma/\mu)^2}$ for Problem (15), and the required number of operator evaluations is $O(\pi_{\mathcal{M}(A,B+\sigma(I-\bar{x}))} \log(\frac{2(\mu+\sigma)}{\mu}))$. In each inner loop, the parameter $\bar{x}$ is fixed and does not affect the order of $\pi_{\mathcal{M}(A,B+\sigma(I-\bar{x}))}$ and so it

does not affect the overall complexity. We then arrive at the following theorem on the total complexity of Catalyst for (1).

Theorem 9 *Suppose Assumption 1 holds and let* $\{\check{x}_k\}_{k=1}^T$ *be generated by Algorithm 2. Then, the complexity of Catalyst to achieve* $\mathbb{E}\|\check{x}_T - x^*\|^2 \leq \epsilon$ *is*

$$O\left(\frac{\sigma+\mu}{\mu}\pi_{\mathcal{M}(A,B+\sigma I)}\log\left(\frac{2(\mu+\sigma)}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)\right). \tag{17}$$

Note that when $\sigma = 0$, the complexity of Catalyst is the same as that of the original $\mathcal{M}$. The optimal value σ^* for σ to achieve the optimal complexity is algorithm-dependent. The following table compares the complexity of various algorithms before and after Catalyst acceleration.

Algorithms	without Catalyst	with Catalyst	optimal σ
SAGA	$O(\kappa^2\log(1/\epsilon))$	$\tilde{O}(\kappa\sqrt{n}\log(1/\epsilon))$	$O(\kappa\mu/\sqrt{n})$
SVRG	$O(\kappa^2\log(1/\epsilon))$	$\tilde{O}(\kappa\sqrt{n}\log(1/\epsilon))$	$O(\kappa\mu/\sqrt{n})$
HSAG	$O(\kappa^2\log(1/\epsilon))$	$\tilde{O}(\kappa\sqrt{n}\log(1/\epsilon))$	$O(\kappa\mu/\sqrt{n})$
SVRG-rand	$O(\kappa^2\log(1/\epsilon))$	$\tilde{O}(\kappa\sqrt{n}\log(1/\epsilon)$	$O(\kappa\mu/\sqrt{n})$
SAGA+SVRG-rand	$O(\kappa^2\log(1/\epsilon))$	$\tilde{O}(\kappa\sqrt{n}\log(1/\epsilon)$	$O(\kappa\mu/\sqrt{n})$
SAGD	$O(\kappa^2\log(1/\epsilon))$	$\tilde{O}(\kappa\sqrt{n}\log(1/\epsilon))$	$O(\kappa\mu/\sqrt{n})$
FB-splitting	$O(\kappa^2 n\log(1/\epsilon))$	$\tilde{O}(\kappa n\log(1/\epsilon))$	$O(\kappa\mu)$
Ac-FB splitting	$O(\kappa n\log(1/\epsilon))$	no acceleration	no acceleration

Table 1: Algorithmic complexity before and after Catalyst acceleration when the problem is ill-conditioned ($\kappa^2 \geq n$). Here, $\tilde{O}$ hides a logarithmic factor, and Ac-FB splitting is the accelerated FB splitting proposed in Balamurugan and Bach (2016) by assuming a linear B.

5. Asynchronous implementation

In this section we study the asynchronous extension of Algorithm 1 for the pure forward step method:

$$x_{k+1} = (I - \gamma B)(x_k) = x_k - \gamma B(x_k),\ \forall k \geq 0,$$

where $A \equiv 0$, and the unique solution satisfies $B(x^*) = 0$.

Our asynchronous setting is similar to Hogwild! (Niu et al., 2011), AsySCD (Liu et al., 2015), and PASSCoDe (Hsieh et al., 2015). We assume a multicore architecture where each core makes updates to a centrally stored vector x in an asynchronous manner to reduce the computation time. The framework of the general asynchronous randomized forward-step algorithm is described in Algorithm 3.

The main steps of Algorithm 3 are read, evaluate, and update. Following the notation in Reddi et al. (2015), we use a global counter k to track the number of updates that are successfully executed to the centrally stored x. Such an after-write approach has become a

Algorithm 3 Asynchronous randomized forward-step algorithm

Input:

An initial value $x_0 \in \mathbb{R}^d$, an algorithm-specific $\phi_i^0 \in \mathbb{R}^d$ for all $i \in [n]$, the total number of iterations T.

1: **While** the number of updates $\leq T$ **do in parallel**
2: Read the central x and the proxies ϕ in the central memory
3: Randomly sample an integer I_k from $\{1, 2, \ldots, n\}$, compute $B_{I_k}(x)$
4: Add $\gamma(B_{I_k}(x) - \phi_{I_k} + \frac{1}{n}\sum_{i=1}^n \phi_i)$ to the centrally stored x and update proxies according to the specific update scheme in Algorithm 1.
5: **End while**

Output:

6: x_{T+1}

standard global labeling scheme in the literature (Niu et al., 2011; Liu et al., 2015; Reddi et al., 2015; Hsieh et al., 2015; Leblond et al., 2017). The value of the centrally stored x and $\phi = (\phi_i)_{i\in[n]}$ after k updates are denoted as x_k and $\phi^k = (\phi_i^k)_{i\in[n]}$, respectively. Each processor does the read-evaluate-update steps simultaneously, and so x and ϕ can have different time labels in the read and update steps. We use $D(k) \in \{1, 2, \ldots, k\}$ to denote the time label of the particular x and proxies used in the read step of the $(k+1)$th update. This means that $\phi^{D(k)}$ and $x_{D(k)}$ are used to compute the value added to the centrally stored x in the $(k+1)$th update, and we define

$$\hat{x}_k \triangleq x_{D(k)}, \quad \hat{\phi}_i^k \triangleq \phi_i^{D(k)}, \forall i \in [n], \quad \forall k \geq 0.$$

We make the following assumption for the convergence analysis of the asynchronous Algorithm 3.

Assumption 3

(3.1) There exists an integer $\tau \geq 0$ such that $0 \leq k - D(k) \leq \tau$ for all $k \geq 0$.

(3.2) There exists a constant $M_{\phi,B} \geq 0$ such that $\|\phi_i^k\| \leq M_{\phi,B}$ and $\|B_i(x_k)\| \leq M_{\phi,B}$ for all $k \geq 0$ and $i \in [n]$.

The first assumption bounds the delay between the read and update times by τ. This assumption is typical in asynchronous systems (Reddi et al., 2015; Mania et al., 2017). Following Mania et al. (2017), we also assume that $\|B_i(x_k)\|$ and $\|\phi_i^k\|$ for all $i \in [n]$ are bounded for all $k \geq 0$.

The asynchronous variance-reduced forward step can be analyzed with the perturbed iterate framework from Mania et al. (2017) which has the following form

$$x_{k+1} = x_k - \gamma\, \mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k),\ k \geq 0, \tag{18}$$

where

$$\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k) \triangleq B_{I_k}(\hat{x}_k) - \hat{\phi}_{I_k}^k + \frac{1}{n}\sum_{i=1}^n \hat{\phi}_i^k, \quad k \geq 0.$$

Here, $\hat{x}_k$ and $\hat{\phi}^k$ can be interpreted as perturbed versions of x_k and ϕ^k due to asynchronous updating. It is readily seen that $\mathbb{E}\left[\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k) \,|\, \mathcal{F}_k\right] = B(\hat{x}_k)$, and so this estimator is unbiased. Applying (18) recursively shows that

$$x_k = \overbrace{\underbrace{x_0 - \gamma\,\mathcal{G}(\hat{x}_0, \hat{\phi}^0, I_0)}_{x_1} - \gamma\,\mathcal{G}(\hat{x}_1, \hat{\phi}^1, I_1)}^{x_2} - \cdots - \gamma\,\mathcal{G}(\hat{x}_{k-1}, \hat{\phi}^{k-1}, I_{k-1}),\ \forall k \geq 1. \tag{19}$$

Since $\hat{x}_k = x_k$ when $\tau = 0$, the above scheme recovers the synchronous case when $\tau = 0$.

The following lemma is the key to our analysis of Algorithm 3. It shows how the expected distance to the optimal solution x^* changes after one iteration. Similar to Lemma 4, we show that the expected distance to x^* contracts by a factor strictly less than one plus an additional disturbance. This additional disturbance depends on the variance of the estimator and the error due to the asynchronous updates.

Lemma 10 *Suppose Assumptions 1 and 3 hold, and let $\{x_k\}_{k\geq 0}$ be generated by Algorithm 3. Then, for all $k \geq 0$ we have*

$$\begin{aligned}\mathbb{E}\|x_{k+1} - x^*\|^2 \leq (1-\gamma\mu)\mathbb{E}\|x_k - x^*\|^2 + 2\gamma\mu\mathbb{E}\|\hat{x}_k - x_k\|^2 \\ + \gamma^2\mathbb{E}\|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)\|^2 + 2\gamma\mathbb{E}(\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k))^\top(\hat{x}_k - x_k).\end{aligned} \tag{20}$$

Eq. (20) shows that the convergence of Algorithm 3 is closely related to three error terms: the distance between the true x_k and its perturbed version $\hat{x}_k$, the norm of the estimator $\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)$, and the inner product of the mismatch $x_k - \hat{x}_k$ and the estimator $\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)$. The following lemma bounds these three error terms using the constant $M \triangleq 3M_{\phi,B}$.

Lemma 11 *Suppose Assumption 1 and Assumption 3 hold, and let $\{x_k\}_{k\geq 0}$ be generated by Algorithm 3. Then, for all $k \geq 0$ we have*

$$\mathbb{E}\|x_k - \hat{x}_k\|^2 \leq \gamma^2\tau^2M^2, \quad \mathbb{E}(\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k))^\top(\hat{x}_k - x_k) \leq \gamma\tau M^2,$$

and

$$\begin{aligned}\mathbb{E}\|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)\|^2 \leq 6L^2\gamma^2\tau^2M^2 + 6L^2\mathbb{E}\|x_k - x^*\|^2 + \frac{4}{n}\mathbb{E}\sum_{i=1}^n \|\phi_i^k - B_i(x^*)\|^2 \\ + \frac{4}{n}\mathbb{E}\sum_{i=1}^n \|\phi_i^k - \hat{\phi}_i^k\|^2.\end{aligned}$$

We define an additional constant

$$\mathcal{E}_0 \triangleq 2\gamma^3\mu\tau^2M^2 + 6\gamma^4L^2\tau^2M^2 + 2\gamma^2\tau M^2,$$

and the sequence $a_k \triangleq \mathbb{E}\|x_k - x^*\|^2$ for all $k \geq 0$ to succinctly express Lemmas 10 and 11 in the following corollary.

Corollary 12 *Suppose Assumptions 1 and 3 hold, and let $\{x_k\}_{k\geq 0}$ be generated by Algorithm 3. Then, for all $k \geq 0$ we have*

$$a_{k+1} \leq (1 - \gamma\mu + 6\gamma^2 L^2)a_k + \frac{4\gamma^2}{n}\,\mathbb{E}\sum_{i=1}^{n} \|\phi_i^k - B_i(x^*)\|^2 + \frac{4\gamma^2}{n}\mathbb{E}\sum_{i=1}^{n} \|\phi_i^k - \hat{\phi}_i^k\|^2 + \mathcal{E}_0. \quad (21)$$

Corollary 12 is similar to Corollary 6. However, (21) includes two additional terms from the error caused by asynchronous updates.

To continue, we introduce the following assumptions on $G(\phi^k)$ and $H(\phi^k)$. We introduce two new non-negative parameters $\mathcal{E}_1$ and $\mathcal{E}_2$ to capture the error caused by the asynchronous updates. We show how to choose these later for specific algorithms.

Assumption 4

(4.1) There exist constants c_1, c_2, and $\mathcal{E}_1$ with $0 \leq c_1 < 1$, $c_2 \geq 0$, and $\mathcal{E}_1 \geq 0$, such that $\mathbb{E}G(\phi^{k+1}) \leq c_1\mathbb{E}G(\phi^k) + c_2\mathbb{E}\,\|x_k - x^*\|^2 + \mathcal{E}_1$.

(4.2) There exists a constant c_3 such that $H(\phi^k) \leq c_3 L_\rho(x_{S_i}, \phi^{S_i})$ for all $S_i \leq k < S_{i+1}$.

(4.3) There exists a constant $\mathcal{E}_2 \geq 0$ such that $\sum_{i=1}^n \mathbb{E}\|\phi_i^k - \hat{\phi}_i^k\|^2/n \leq \mathcal{E}_2$.

(4.4) The system is fully synchronized at the end of every epoch.

The above assumption is similar to Assumption 2 with the additional requirement that the average expected distance between $\hat{\phi}_i^k$ and ϕ_i^k is bounded by a constant. Assumption 4.4 also appears in Reddi et al. (2015). Then, we define

$$\theta \triangleq \max\left\{1 - \gamma\mu + 6\gamma^2 L^2 + c_2\rho, \frac{4\gamma^2}{\rho} + c_1\right\},$$
$$\lambda \triangleq \theta + 4\gamma^2\bar{m}c_3,$$

and

$$\mathcal{E}_3 \triangleq 4\gamma^2\mathcal{E}_2 + \mathcal{E}_0 + \rho\mathcal{E}_1,$$

where we recall $\bar{m}$ is the upper bound on the epoch lengths from Assumption 2. We have the following step-size rule:

$$\gamma < \min\left\{\left(\frac{\mu}{\frac{3(1-c_1)L^2}{2} + (1-c_1)\bar{m}c_3 + c_2}\right)^2, \left(\frac{1-c_1}{4 + 4\bar{m}(\frac{1-c_1}{4})^3 c_3}\right)^2\right\}. \quad (22)$$

Theorem 13 *Suppose Assumptions 1, 3, and 4 hold, and that the step-size γ satisfies* (22)*. Then, for $\rho = \gamma^{1.5}$, we have θ, $\lambda \in [0, 1)$, and*

$$\mathbb{E}L_\rho(\tilde{x}_k, \tilde{\phi}^k) \leq \lambda\,\mathbb{E}L_\rho(\tilde{x}_{k-1}, \tilde{\phi}^{k-1}) + \frac{\mathcal{E}_3}{1-\theta},\ \forall k \geq 1. \quad (23)$$

Theorem 13 shows that the iterates generated by Algorithm 3 converge linearly to a neighborhood of x^*. We will see in our examples that if the delay $\tau = 0$, then the parameters $\mathcal{E}_0$, $\mathcal{E}_1$, $\mathcal{E}_2$, and $\mathcal{E}_3$ are all zero, so that Theorem 13 matches the synchronous case.

6. Examples

This section considers several specific algorithms within our framework. For these algorithms, we first verify Assumption 2, then we analyze the number of operator evaluations required to find x_ϵ^* such that $\mathbb{E}\|x_\epsilon^* - x^*\|^2 \leq \epsilon$. We also compute the complexity of Catalyst, and verify Assumption 4 for asynchronous implementation where applicable. The detailed computations are in Appendix E. Throughout, we let $\{W_k\}_{k\geq 0}$ be a sequence of i.i.d. standard uniform random variables. We also use the common notation $\tilde{O}$ to hide logarithmic factors (Lin et al., 2015, 2018).

6.1 GD

Full gradient descent (see Nesterov (2014)) follows (7) where the proxies satisfy $\phi_i^k = B_i(x_k)$ for all $i \in [n]$, and so $\mathcal{G}\big(x_k, \phi^k, I_k\big) = \frac{1}{n}\sum_{i=1}^n B_i(x_k) = B(x_k)$ for all $k \geq 0$. We choose $\mathcal{S} = \emptyset$ and so

$$H(\phi^k) = \frac{1}{n}\|\phi_i^k - B_i(x^*)\|^2 = \frac{1}{n}\sum_{i=1}^n \|B_i(x_k) - B_i(x^*)\|^2 \leq L^2\|x_k - x^*\|.$$

It is then readily seen that Assumption 2 holds with $c_1 = c_2 = 0$, $c_3 = L^2$, and $m_i = 1$ for all $i \geq 1$.

Since $\mathcal{G}\big(x_k, \phi^k, I_k\big) = B(x_k)$, we can choose a constant step-size $\gamma = \mu/L^2$ to minimize the coefficient in (9) and then use Lemma 4 to obtain

$$\|x_k - x^*\|^2 \leq \left(1 - \frac{1}{\kappa^2}\right)\|x_{k-1} - x^*\|^2, \forall k \geq 1.$$

Thus, GD requires $O(\kappa^2 \log(1/\epsilon))$ iterations to obtain an ϵ-solution, and each iteration requires n operator evaluations. The total complexity is then $O(n\kappa^2 \log(1/\epsilon))$, that is, $\pi_{\mathcal{M}(A,B)} = n\kappa^2$. Substitute this value of $\pi_{\mathcal{M}(A,B)}$ into (17) to see that the complexity of GD with Catalyst is

$$\tilde{O}\left(n\frac{\sigma+\mu}{\mu}\left(\frac{L+\mu}{\sigma+\mu}\right)^2 \log\left(\frac{1}{\epsilon}\right)\right).$$

For $\sigma = (\kappa - 1)\mu$, the complexity of GD with Catalyst is $\tilde{O}(n\kappa \log(1/\epsilon))$. This is the same as the complexity of the accelerated forward-backward algorithm from Balamurugan and Bach (2016) (up to a logarithmic factor). However, Balamurugan and Bach (2016) requires B to be *linear*, while our analysis holds for general B satisfying Assumption 1.

6.2 SVRG

SVRG (Johnson and Zhang, 2013; Xiao and Zhang, 2014; Konečný and Richtárik, 2017) is epoch-based. Here we allow unequal epoch lengths as long as $\bar{m} < \infty$. SVRG follows (7) where the proxies satisfy:

$$\phi_i^t = B_i(x_{S_k}), \forall i \in [n],\ S_k \leq t < S_{k+1}.$$

We select $\mathcal{S} = \emptyset$ and thus Assumption 2.1 holds with $c_1 = c_2 = 0$. Since $\phi_i^t = B_i(x_{S_k})$ for all $i \in [n]$ and $S_k \leq t < S_{k+1}$, we have

$$H(\phi^t) = \frac{1}{n}\sum_{i=1}^{n} \|B_i(x_{S_k}) - B_i(x^*)\|^2 \leq L^2 \|x_{S_k} - x^*\|^2 \leq L^2 L_\rho \left(x_{S_k}, \phi^{S_k}\right),$$

where the first inequality is due to L-Lipschitz continuity of each B_i, and the second inequality is by definition of L_ρ. Therefore, Assumption 2.2 holds with $c_3 = L^2$.

We consider the complexity of SVRG for constant epoch length $m \geq 1$ and constant step-size $\gamma > 0$. We choose step-size $\gamma = (\mu/3L^2)$ and epoch length $m = O(\kappa^2)$ so that the expected distance to x^* contracts by a factor of at least $3/4$ after every epoch. To achieve an ϵ-solution, we need $O(\log(1/\epsilon))$ epochs. In the beginning of each epoch, the proxy update requires n operator evaluations, and each of the inner m iterations requires two operator evaluations. With $m = O(\kappa^2)$, the total number of operator evaluations is $O((\kappa^2 + n)\log(1/\epsilon))$. In the ill-conditioned setting with $n \leq \kappa^2$ (Lin et al., 2015), the total complexity is $O\left(\kappa^2 \log(1/\epsilon)\right)$.

Now we consider SVRG with Catalyst for $n \leq \kappa^2$ where $\pi_{\mathcal{M}(A,B)} = n + \kappa^2$ and $\pi_{\mathcal{M}(A,B+\sigma I)} = n + ((L+\sigma)/(\mu+\sigma))^2$. Choosing $\sigma^* = O(\kappa\,\mu/\sqrt{n})$ minimizes the complexity of (17), and so the complexity of SVRG with Catalyst is $O(\kappa\sqrt{n}\log(1/\epsilon))$. This is less than the complexity of SVRG (that is, $O(\kappa^2 \log(1/\epsilon))$) when $\kappa^2 \geq n$.

In asynchronous SVRG, the system is synchronized after every epoch so we have $S_k \leq D(t) \leq t$ for all $t \geq S_k$ (Reddi et al., 2015, Section 3). The proxies do not change within an epoch and so $\phi^k = \phi^{D(k)} = \hat{\phi}^k$. As a result, Assumption 4.3 holds with $\mathcal{E}_2 = 0$. To verify Assumption 4.1, we have $G(\phi^k) \equiv 0$ since $\mathcal{S} = \emptyset$. Assumption 4.2 is the same as Assumption 2.2. So Assumption 4 holds with $\mathcal{S} = \emptyset$, $c_1 = c_2 = \mathcal{E}_1 = \mathcal{E}_2 = 0$, and $c_3 = L^2$.

6.3 SAGA

SAGA (Defazio et al., 2014) is initialized with $\phi_i^0 = B_i(x_0)$ for all $i \in [n]$ and the proxies are updated according to:

$$\phi_i^{k+1} = \begin{cases} B_i(x_k), & i = I_k, \\ \phi_i^k, & i \neq I_k. \end{cases}$$

Let $\mathcal{S} = [n]$ and $m_i = 1$ for all $i \geq 1$. To verify Assumption 2.1, observe that ϕ_i^k changes to $B_i(x_{k-1})$ only when $I_{k-1} = i$, and so for all $k \geq 1$ we have

$$\begin{aligned}\mathbb{E}\left[G(\phi^k)|\mathcal{F}_{k-1}\right] =& \frac{1}{n}\sum_{i=1}^{n}\left(G(\phi^{k-1}) + \frac{1}{n}\left(\|(B_i(x_{k-1}) - B_i(x^*)\|^2 - \|\phi_i^{k-1} - B_i(x^*)\|^2\right)\right) \\ =& \left(1 - \frac{1}{n}\right) G(\phi^{k-1}) + \frac{1}{n}\sum_{i=1}^{n}\frac{\|B_i(x_{k-1}) - B_i(x^*)\|^2}{n}.\end{aligned}$$

Since all $\{B_i\}_{i\in[n]}$ are L-Lipschitz, we have

$$\mathbb{E}G(\phi^k) \leq \left(1 - \frac{1}{n}\right)\mathbb{E}G(\phi^{k-1}) + \frac{L^2}{n}\mathbb{E}\|x_{k-1} - x^*\|^2, \quad \forall k \geq 1. \tag{24}$$

We conclude that Assumption 2.1 holds with $c_1 = 1 - 1/n$ and $c_2 = L^2/n$. Furthermore, Assumption 2.2 holds with $c_3 = 0$ since $H(\phi^k) \equiv 0$. When $\gamma = O(\mu/L^2)$, $\pi_{\mathcal{M}(A,B)} = \max\left\{n, \kappa^2\right\}$. If $n \leq \kappa^2$, then this complexity is $O\left(\kappa^2 \log(1/\epsilon)\right)$ (the same as SVRG).

To analyze SAGA with Catalyst, recall $\pi_{\mathcal{M}(A,B)} = \max\left\{n, \kappa^2\right\}$ and

$$\pi_{\mathcal{M}(A,B+\sigma(I-\bar{x}))} = \max\left\{n, \left(\frac{L+\sigma}{\mu+\sigma}\right)^2\right\}.$$

Choose $\sigma^* = O(\kappa\,\mu/\sqrt{n})$ to minimize (17) so the overall complexity of SAGA with Catalyst is $O(\kappa\sqrt{n}\log(1/\epsilon))$, which is smaller than the complexity of classical SAGA under the condition $\kappa^2 \geq n$.

In asynchronous SAGA, Assumption 4.2 holds with $c_3 = 0$ since $H(\phi^k) \equiv 0$. To verify Assumption 4.3, observe that SAGA only changes one proxy in each iteration, and so there are at most τ terms among all $\{\hat{\phi}_i^k\}_{i=1}^n$ that are different from their counterparts in $\{\phi_i^k\}_{i=1}^n$. Use Assumption 3.2 to see that $\sum_{i=1}^n \mathbb{E}\|\phi_i^k - \hat{\phi}_i^k\|^2/n \leq 4\tau M_{\phi,B}^2/n$. Finally, Assumption 4 holds with $\mathcal{S} = [n]$, $m_i = 1$ for all $i \geq 1$, $\mathcal{E}_1 = 2L^2\gamma^2\tau^2M^2/n$, $c_1 = 1 - 1/n$, $c_2 = 2L^2/n$, $c_3 = 0$, and $\mathcal{E}_2 = 4\tau M_{\phi,B}^2/n$.

6.4 SVRG-rand

Empirically, SVRG tends to be slower than SAGA given the same computational budget, while SAGA incurs greater storage cost (Defazio et al., 2014; Reddi et al., 2015). We propose a new algorithm based on SVRG which we call SVRG-rand that allows for random epoch lengths. SVRG-rand does not require additional storage cost, and our simulations suggest that it performs better than classical SVRG.

We introduce the sequence $\{p_k\}_{k\geq 0}$ where $p_k \in [0,1]$ is the probability of doing a full proxy update in iteration $k \geq 0$. We also define $\underline{p} \triangleq \inf_{k\geq 0}\{p_k\}$ and $\overline{p} \triangleq \sup_{k\geq 0}\{p_k\}$. Accordingly, the proxies are updated according to:

$$\phi_i^{k+1} = \mathbb{I}(W_k \leq p_k)B_i(x_k) + \mathbb{I}(W_k > p_k)\phi_i^k,\ \forall i \in [n]. \tag{25}$$

We initialize with $\phi_i^0 = 0$ for all $i \in [n]$ which differs from the initialization in SVRG. As long as $\underline{p} > 0$, SVRG-rand satisfies Assumption 2 with $\mathcal{S} = [n]$, $m_i = 1$ for all $i \geq 1$, $c_1 = 1 - \underline{p}$, and $c_2 = \overline{p}L^2$ (see Theorem 14). SVRG-rand is essentially not epoch-based, nevertheless as an extension of SVRG it does not require additional storage cost. Classical SVRG is recovered as a special case of SVRG-rand if $p_k = 1$ when k is a multiple of m, and $p_k = 0$ otherwise. SVRG with increasing epochs can be recovered similarly (see (Allen-Zhu and Yuan, 2016)).

We analyze the complexity of SVRG-rand where $p_k = 1/n$ for all $k \geq 0$. Then $\overline{p} = \underline{p} = 1/n$, and (24) holds for SVRG-rand by Theorem 14. The rest of the complexity analysis is then the same as SAGA and we achieve a complexity of $O(\max\{n, \kappa^2\}\log(1/\epsilon))$. If $n \leq \kappa^2$, then this complexity is $O\left(\kappa^2\log(1/\epsilon)\right)$ (the same as SVRG and SAGA). Based on (17), the complexity of SVRG-rand with Catalyst is also the same as SAGA and SVRG, that is, $O(\kappa\sqrt{n}\log(1/\epsilon))$.

We now consider asynchronous SVRG-rand. Assumption 4.2 holds with $c_3 = 0$ since $H(\phi^k) \equiv 0$. Because the system is synchronized after every epoch, $\phi^k = \phi^{D(k)} = \hat{\phi}^k$ as in

SVRG. As a result, Assumption 4.3 holds with $\mathcal{E}_2 = 0$. Similar to Theorem 14, we have

$$\mathbb{E}G(\phi^k) \leq (1-\underline{p})\mathbb{E}G(\phi^{k-1}) + \overline{p}L^2\mathbb{E}\|\hat{x}_{k-1} - x^*\|^2, \forall k \geq 1.$$

By adding and subtracting x_{k-1} in the second term of the above inequality, we have

$$\mathbb{E}G(\phi^k) \leq (1-\underline{p})\mathbb{E}G(\phi^{k-1}) + 2\overline{p}L^2\mathbb{E}\|x_{k-1} - x^*\|^2 + 2\overline{p}L^2\gamma^2\tau^2M^2, \forall k \geq 1.$$

Therefore, Assumption 4 holds with $\mathcal{S} = [n]$, $m_i = 1$ for all $i \geq 1$, $c_1 = 1 - \underline{p}$, $c_2 = 2\overline{p}L^2$, $\mathcal{E}_1 = 2\overline{p}L^2\gamma^2\tau^2M^2$, $c_3 = 0$, and $\mathcal{E}_2 = 0$.

6.5 SAGD

SAGD (Bibi et al., 2018) is based on probabilistic interpolation between GD and SAGA. In each iteration, SAGD takes a GD step with probability q and a SAGA step with probability $1-q$. The proxies are then updated according to:

$$\phi_i^{k+1} = \{\mathbb{I}(W_k \leq q) + \mathbb{I}(W_k > q)\mathbb{I}(I_k = i)\}\, B_i(x_k) + \mathbb{I}(W_k > q)\mathbb{I}(I_k \neq i)\phi_i^k. \tag{26}$$

That is, SAGD updates all the proxies in a GD step and it updates only one proxy in a SAGA step. We initialize $\phi_i^0 = 0$ for all $i \in [n]$ as in SVRG-rand. Then, we choose $\mathcal{S} = [n]$ and compute:

$$\begin{aligned}\mathbb{E}\left[G(\phi^{k+1})\middle|\mathcal{F}_k\right] &= \left(q + \frac{1-q}{n}\right)\frac{1}{n}\sum_{i=1}^n \|B_i(x_k) - B_i(x^*)\|^2 + (1-q)\left(1-\frac{1}{n}\right)G(\phi^k) \\ &\leq (1-q)\left(1-\frac{1}{n}\right)G(\phi^k) + \left(q + \frac{1-q}{n}\right)L^2\|x_k - x^*\|. \end{aligned}\tag{27}$$

Therefore, SAGD satisfies Assumption 2 with $\mathcal{S} = [n]$, $c_1 = (1-q)(1-1/n)$, $c_2 = 1 - c_1$, and $c_3 = 0$.

We consider the complexity of SAGD for $q = 1/n$ (the analysis for general q is similar). When $q = 1/n$, (27) implies

$$\mathbb{E}G(\phi^k) \leq \left(1 - \frac{1}{n}\right)\mathbb{E}G(\phi^{k-1}) + \frac{2L^2}{n}\mathbb{E}\|x_{k-1} - x^*\|^2, \forall k \geq 1.$$

The above display is almost the same as (24) for SAGA. We can thus use the same arguments to see the complexity of SAGD under $q = 1/n$ is $O(\max\{n, \kappa^2\}\log(1/\epsilon))$. When $\kappa^2 \geq n$, the complexity of SAGD with Catalyst is $O(\kappa\sqrt{n}\log(1/\epsilon))$.

6.6 HSAG

HSAG (Reddi et al., 2015) combines SVRG and SAGA. For a set $\mathcal{S} \subset [n]$ with cardinality $S \triangleq |\mathcal{S}|$, the proxies in HSAG are updated according to:

$$\phi_i^{k+1} = \begin{cases} \mathbb{I}(I_k = i)\, B_i(x_k) + \mathbb{I}(I_k \neq i)\, \phi_i^k, & i \in \mathcal{S}, \\ \mathbb{I}(m \mid (k+1))\, B_i(x_{k+1}) + \mathbb{I}(m \nmid (k+1))\, \phi_i^k, & i \in \mathcal{S}^c. \end{cases}$$

Then, Assumption 2.1 holds with $c_1 = 1 - 1/n$ and $c_2 = SL^2/n^2$, and Assumption 2.2 holds with $c_3 = (n-S)\,L^2/n$ and $m_i = m$ for all $i \geq 1$.

We choose $m = O\left(\max\{\kappa^2, n\mathbb{I}(S>0)\}\right)$ to obtain a complexity of

$$\pi_{\mathcal{M}(A,B)} = O\left(\max\{\kappa^2, n\mathbb{I}(S>0)\} + n - S\right)\log(1/\epsilon).$$

This expression recovers the complexity of SVRG when $S = 0$ and SAGA when $S = n$. Choose $\sigma^* = O(\kappa\,\mu/\sqrt{n})$ to minimize (17), so the complexity of HSAG with Catalyst is $O\left(\kappa\sqrt{n}\log\left(1/\epsilon\right)\right)$ for $S = O(n)$.

For asynchronous HSAG, Assumption 4 holds with $c_1 = 1 - 1/n$, $c_2 = 2SL^2/n^2$, $\mathcal{E}_1 = 2SL^2\gamma^2\tau^2M^2/n^2$, $c_3 = (n-S)\,L^2/n$, and $\mathcal{E}_2 = 4\tau M_{\phi,B}^2/n$.

6.7 SAGA+SVRG-rand

We build on the hybridization idea of HSAG and combine SVRG-rand and SAGA. SVRG-rand needs to compute the full operator with a positive, though usually small, probability in every iteration. The full operator evaluation is time-consuming for large n. Alternatively, SAGA achieves variance reduction by storing the entire proxy vector, which avoids the need for periodic full operator evaluations.

Let $\mathcal{S}_1 \subseteq [n]$ be the index set for the proxies following a SAGA-type update, so the proxies in $\mathcal{S}_2 = \mathcal{S}_1^c$ follow SVRG-rand-type updates. Then, the proxies are updated according to:

$$\phi_i^{k+1} = \begin{cases} \mathbb{I}\left(I_k = i\right)B_i(x_k) + \mathbb{I}\left(I_k \neq i\right)\phi_i^k & i \in \mathcal{S}_1, \\ \mathbb{I}(W_k \leq p_k)B_i(x_k) + \mathbb{I}(W_k > p_k)\phi_i^k & i \in \mathcal{S}_2. \end{cases} \tag{28}$$

Under the same assumptions on $\{p_k\}_{k\geq 1}$ as SVRG-rand, SAGA+SVRG-rand satisfies Assumption 2 with $\mathcal{S} = [n]$, $m_i = 1$ for all $i \geq 1$, $c_3 = 0$, $c_1 = \max\left\{1 - 1/n, 1 - \underline{p}\right\}$, and $c_2 = S_1L^2/n^2 + \overline{p}S_2L^2/n$.

The complexity of SAGA+SVRG-rand when $p_k = 1/n$ for all $k \geq 1$ is $O(\max\{n, \kappa^2\}\log(1/\epsilon))$, which is the same as SAGA. In view of this observation, the complexity of SAGA+SVRG-rand with Catalyst is $O(\kappa\sqrt{n}\log\left(1/\epsilon\right))$.

6.8 SARAH

SARAH (see Algorithm 4) is another extension of SVRG (Nguyen et al., 2017). SARAH is also epoch-based, but it uses a biased gradient estimator v_k. Unless k is a multiple of the epoch length m, we have

$$\mathbb{E}\left[v_k \middle| \mathcal{F}_k\right] = B(x_k) - B(x_{k-1}) + v_{k-1} \neq B(x_k).$$

Recall $\tilde{x}_k = x_{km}$ for $k \geq 0$ are the iterates at the beginning of each epoch. In Appendix E, we establish linear convergence of SARAH for solving Problem (1) when $A \equiv 0$ by showing that $\mathbb{E}\|B(\tilde{x}_k)\|^2$ contracts by a factor of $3/4$ after every epoch, when the epoch length is $m = O(\kappa^2)$. The overall complexity of SARAH is then $O\left((n+\kappa^2)\log(1/\epsilon)\right)$.

Algorithm 4 SARAH

Input:

An initial value $x_0 \in \mathbb{R}^d$, the total number of iterations T, and the length of epochs m. Constant step-size γ.

for each $k = 0, 1, \ldots, T$ **do**

  Randomly sample I_k uniformly from $[n]$

  $v_k = \begin{cases} \frac{1}{n}\sum_{i=1}^n B_i(x_k) & k \mid m \\ B_{I_k}(x_k) - B_{I_k}(x_{k-1}) + v_{k-1} & k \nmid m \end{cases}$

  $x_{k+1} = x_k - \gamma v_k$

end for

Output:

x_{T+1}

6.9 Summary

We summarize the complexities of the algorithms considered in this section in Table 2. Table 2 also compares the complexity for solving a monotone inclusion problem versus doing function minimization. Using the inequality $\frac{1}{2}(n+\kappa^2) \leq \max\{n, \kappa^2\} \leq n + \kappa^2$, we obtain $\pi_{\mathcal{M}(A,B)} = n + \kappa^2$ for SAGA (as well as several other examples in this section) for solving Problem (1). Since SVRG-rand and SAGA+SVRG-rand are newly proposed in this study for Problem (1), there are no existing results on their complexity for Problem (2).

We observe that in general the complexity of solving Problem (1) is greater than the complexity of solving Problem (2). A similar phenomenon is observed in Chen et al. (2017), which argues why the complexity of solving variational inequality problems is greater than for doing function minimization. Giselsson (2017) studies Douglas-Rachford Splitting, and also finds a difference between the convergence rate for solving monotone inclusion problems and doing function minimization. We suspect that part of this disparity in complexity is due to the special properties enjoyed by convex functions and function minimization (for example, coercivity) that do not easily extend to the more general problem classes covered by Problem (1).

Algorithm	Complexity of solving (1)	Complexity of solving (2)
GD	$O(n\kappa^2)\log(1/\epsilon)$	$O(n\kappa)\log(1/\epsilon)$ (Nesterov, 2014)
SVRG	$O(n+\kappa^2)\log(1/\epsilon)$	$O(n+\kappa)\log(1/\epsilon)$ (Johnson and Zhang, 2013)
SAGA	$O(n+\kappa^2)\log(1/\epsilon)$	$O(n+\kappa)\log(1/\epsilon)$ (Defazio et al., 2014)
SVRG-rand	$O(n+\kappa^2)\log(1/\epsilon)$	no results
SAGD	$O(n+\kappa^2)\log(1/\epsilon)$	$O(n+\kappa)\log(1/\epsilon)$ (Bibi et al., 2018)
HSAG	$O(n+\kappa^2)\log(1/\epsilon)$	$O(n+\kappa)\log(1/\epsilon)$ (Reddi et al., 2015)
SAGA+SVRG-rand	$O(n+\kappa^2)\log(1/\epsilon)$	no results
SARAH	$O(n+\kappa^2)\log(1/\epsilon)$	$O(n+\kappa)\log(1/\epsilon)$ (Nguyen et al., 2017)

Table 2: Comparison of complexity of different algorithms for solving monotone inclusions and for minimizing finite sums, respectively.

7. Numerical experiments

This section reports numerical experiments for a saddle-point problem (for policy evaluation) and a variational inequality problem (for a strongly monotone two-player game). Both of these applications fall outside the realm of function minimization.

We compare the performance of the following algorithms in this section:

- **SVRG**. The epoch length is set to be $m = 2n$ (Johnson and Zhang, 2013; Xiao and Zhang, 2014; Allen-Zhu and Yuan, 2016).
- **SAGA**. Following the discussion in Section 6, we use a constant step-size $\mu/(7L^2)$.
- **SVRG++**. An algorithm of Allen-Zhu and Yuan (2016) which falls in the category of SVRG with increasing epochs. The epoch length is initialized as $2n$ and it is doubled after every epoch (Allen-Zhu and Yuan, 2016).
- **SVRG-rand**. The proposed extension of SVRG described by (25). We let p_k be a small number for the first n iterations to avoid too many full operator evaluations in early iterations. After the first n iterations, p_k is assigned to decrease by half from $1/(2n)$ after every epoch to control the frequency of full updates in later iterations. We adopt the common strategy of automatically terminating the epoch if the current epoch is longer than $8n$ (Allen-Zhu and Yuan, 2016; Min et al., 2017).
- **Hybrid SAGA and SVRG-rand**. The proposed hybrid method with update rule given by (28) where the first $n/2$ proxies perform a SAGA-type update.
- **SAGD.** The probability of performing a GD update is $q = 1/(2n)$.
- **SARAH.** Following Nguyen et al. (2017), the epoch length is set as $2n$.

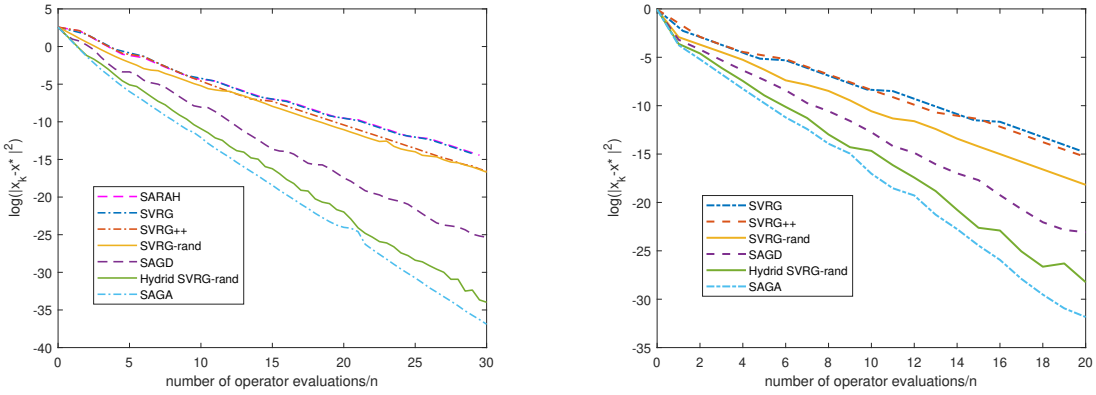
We run each algorithm ten times with the same constant step-size and report the mean performance. Following the tradition established in Xiao and Zhang (2014); Defazio et al. (2014); Allen-Zhu and Yuan (2016); Schmidt et al. (2017), we compare different algorithms based on the number of operator evaluations.

Policy evaluation Consider the saddle-point problem from Du et al. (2017) given by:

$$\min_{\theta\in\mathbb{R}^d}\max_{\omega\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^{n}\left(\omega^\top b_i - \omega^\top A_i\theta - \frac{1}{2}\omega^T C_i\omega + \frac{\lambda}{2}\|\theta\|^2\right), \tag{29}$$

where $\lambda > 0$, θ, ω, $b_i \in \mathbb{R}^d$, and A_i, $C_i \in \mathbb{R}^{d\times d}$. Problem (29) is closely related to the policy evaluation problem in dynamic programming (Dann et al., 2014; Du et al., 2017). In this experiment, we test the algorithms on the Boyan chain benchmark (Boyan, 2002).

Figure 1a shows the behaviour of the algorithms for solving Problem (29). The horizontal axis is the number of operator evaluations divided by n and the vertical axis is the logarithm of the distance to the optimum. From this figure, we can see that SAGA has the best performance, followed by hybrid SAGA and SVRG-rand. However, to achieve this superior performance, SAGA inherently requires more storage than either hybrid SAGA or SVRG-rand. We also observe that SAGD performs worse than either SAGA or SAGA+SVRG-rand,



(a) saddle-point problem (b) two-player game

Figure 1: Comparison of different algorithms on two problems

even though it has the same storage cost as SAGA. However, SAGD does outperform the four algorithms that do not incur storage cost (SARAH, SVRG, SVRG++, and SVRG-rand). Among these four algorithms that do not require storage cost, our proposed SVRG-rand performs best in both earlier and later iterations. SVRG++ has similar performance to SVRG in early iterations and better performance than SVRG in later iterations. SARAH performs almost the same as SVRG in both earlier and later iterations.

Strongly monotone games Next we consider a general finite sum strongly monotone two player game (see, for example, Tatarenko and Kamgarpour (2018)). Let A_1, $A_2 \subset \mathbb{R}^d$ be closed and convex action sets for each player, and let $A = A_1 \times A_2$ be the set of joint actions. Following Tatarenko and Kamgarpour (2019), we consider $a_1, a_2 \in \mathbb{R}_+$ satisfying $a_1 + a_2 \leq 1$.

Let $f_j : A \to \mathbb{R}$ be the cost function for each player $j = 1, 2$. We suppose that $f_j(a_j, a_{-j})$ is continuously differentiable in $a = (a_j, a_{-j})$ and μ-strongly convex in a_j with an L-Lipschitz gradient, for $j = 1, 2$. To motivate the finite sum structure of this problem, we recall that for fixed actions a_1 and a_2, the cost functions are often an expectation over a set of possible scenarios (Kattsoff, 1945). In this study, we take the specific cost functions

$$f_1(a_1, a_2) = \frac{1}{n}\sum_{i=1}^{n}\left(a_1^\top\left(b_{i,1} + A_{i,1}a_2\right) + \frac{1}{2}a_1^\top C_{i,1}a_1\right),$$

$$f_2(a_1, a_2) = \frac{1}{n}\sum_{i=1}^{n}\left(a_2^\top\left(b_{i,2} + A_{i,2}a_1\right) + \frac{1}{2}a_2^\top C_{i,2}a_2\right),$$

which correspond to using SAA based on historical data $\{b_{i,1}, A_{i,1}, C_{i,1}, b_{i,2}, A_{i,2}, C_{i,2}\}_{i\geq 1}$, and we assume the costs depend linearly on the complementary strategy.

Player j's best response problem given a_{-j} is $\min_{a_j \in A_j} f_j(a_j, a_{-j})$. The *game mapping* is defined as $M(a) \triangleq (\nabla_{a_1} f_1(a_1, a_2), \nabla_{a_2} f_2(a_1, a_2))$, where for our specific game we have

$$M(a) = \frac{1}{n}\sum_{i=1}^{n}\left(M_i \cdot (a_1, a_2)^\top + b_i\right)$$

where

$$M_i \triangleq \begin{bmatrix} C_{i,1} & A_{i,1} \\ A_{i,2} & C_{i,2} \end{bmatrix}, \forall i \in [n],$$

and $b_i \triangleq (b_{i,1}, b_{i,2})^\top$ for all $i \in [n]$. The corresponding VI formulation for a Nash equilibrium is to find $a^* \in A$ such that $M(a^*)^\top (a - a^*) \geq 0$ for all $a \in A$. This VI is equivalent to the monotone inclusion problem $0 \in M(a) + N_A(a)$.

Figure 1b shows the behaviour of the algorithms for computing an equilibrium of this game. Since A is nonzero in this example, we do not test SARAH. Their behaviour for solving the two-player game matches their behavior for the saddle-point problem. However, the gap in performance between SVRG-rand and SVRG is now larger (and still favors SVRG-rand).

By evaluating the performance of different variance-reduced algorithms in two different examples, we observe a trade-off between storage cost and convergence rate. Algorithms with storage cost generally perform better than algorithms without storage cost, and this observation is consistent with the literature (Nicolas et al., 2012; Balamurugan and Bach, 2016). These experiments also help to show that the performance of our two newly proposed algorithms is satisfactory. When no additional storage is allowed, we observe that the new algorithm SVRG-rand is faster than classical SVRG and SVRG++. If additional storage is allowed, then SAGA+SVRG-rand performs better than SAGD, and SAGA converges slightly faster than SAGA+SVRG-rand but at greater storage cost.

8. Conclusion and future work

We have presented a unifying framework for variance-reduced FB splitting for finding zeroes of the sum of two monotone operators, where one is the average of a large number of maximal strongly monotone and Lipschitz operators, and the other is a general maximal monotone operator. Our framework covers many popular variance-reduced algorithms for solving finite sum minimization problems as well as their extensions to monotone inclusion problems.

The basis of our technique is a Lyapunov-type argument which we use to establish linear convergence of a class of algorithms including GD, SVRG, SAGA, HSAG, and several others. These variance-reduced algorithms can be understood in some sense as iteration of near contraction operators subject to a disturbance, and that the effect of the disturbance decays to zero over time when appropriate conditions are met. Our technique also extends to the design of new algorithms which similarly enjoy a linear convergence rate in expectation as well as enjoy favorable numerical properties. We further show that our framework is compatible with Catalyst acceleration and asynchronous implementation.

This work is only the beginning of the analysis of variance-reduced algorithms for monotone inclusion problems. Unbiased operator estimates in the inner forward step played a fundamental role in our analysis. In future work, we will consider SAG and other biased variance-reduced algorithms, as well as other methods for acceleration.

Acknowledgments

We are grateful to the associate editor and two anonymous referees for their constructive comments and suggestions, which have helped improve the quality of the paper. Zhang and Ye were supported by the Natural Science Foundation of China Grant 72071138 and the Singapore MOE Tier 2 Grant R-266-000-143-112. Haskell was supported by the National University of Singapore Young Investigator Award "Practical considerations for large-scale competitive decision making".

References

Alekh Agarwal and Léon Bottou. A lower bound for the optimization of finite sums. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 78–86. JMLR.org, 2015.

Zeyuan Allen-Zhu and Yang Yuan. Improved svrg for non-strongly-convex or sum-of-non-convex objectives. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1080–1089. JMLR.org, 2016.

P. Balamurugan and Francis Bach. Stochastic variance reduction methods for saddle-point problems. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 1416–1424, USA, 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9.

Heinz H. Bauschke and Patrick L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Publishing Company, Incorporated, 1st edition, 2011. ISBN 1441994661, 9781441994660.

Adel Bibi, Alibek Sailanbayev, Bernard Ghanem, Robert Mansel Gower, and Peter Richtárik. Improving SAGA via a Probabilistic Interpolation with Gradient Descent. *arXiv e-prints*, art. arXiv:1806.05633, June 2018.

Justin A. Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2):233–246, Nov 2002. ISSN 1573-0565.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. ISBN 0521833787.

Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, November 2015. ISSN 1935-8237.

Yunmei Chen, Guanghui Lan, and Yuyuan Ouyang. Accelerated schemes for a class of variational inequalities. *Mathematical Programming*, 165(1):113–149, 2017.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995. ISSN 1573-0565.

Christoph Dann, Gerhard Neumann, and Jan Peters. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, 15:809–883, 2014.

A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1646–1654. Curran Associates, Inc., 2014.

Simon S. Du, Jianshu Chen, Lihong Li, Lin Xiao, and Dengyong Zhou. Stochastic variance reduction methods for policy evaluation. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1049–1058, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, CIKM '98, pages 148–155, New York, NY, USA, 1998. ACM. ISBN 1-58113-061-9.

Pontus Giselsson. Tight global linear convergence rate bounds for douglas-rachford splitting, 2017.

Patrick T. Harker and Jong-Shi Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: A survey of theory, algorithms and applications. *Mathematical Programming*, 48(1):161–220, Mar 1990. ISSN 1436-4646. doi: 10.1007/BF01582255. URL `https://doi.org/10.1007/BF01582255`.

Cho-Jui Hsieh, Hsiang-Fu Yu, and Inderjit S. Dhillon. Passcode: Parallel asynchronous stochastic dual co-ordinate descent. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2370–2379. JMLR.org, 2015.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc., 2013.

Louis O. Kattsoff. Theory of games and economic behavior. *Social Forces*, 24(2):245–246, 12 1945. ISSN 0037-7732. doi: 10.2307/2572550. URL `https://doi.org/10.2307/2572550`.

Jakub Konečný and Peter Richtárik. Semi-stochastic gradient descent methods. *Frontiers in Applied Mathematics and Statistics*, 3:9, 2017. ISSN 2297-4687.

Rémi Leblond, Fabian Pedregosa, and Simon Lacoste-Julien. ASAGA: Asynchronous Parallel SAGA. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 46–54, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, pages 3384–3392, Cambridge, MA, USA, 2015. MIT Press. URL `http://dl.acm.org/citation.cfm?id=2969442.2969617`.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. Catalyst Acceleration for First-order Convex Optimization: from Theory to Practice. *Journal of Machine Learning Research*, 18(1):7854–7907, April 2018. URL `https://hal.inria.fr/hal-01664934`. http://jmlr.org/papers/volume18/17-748/17-748.pdf.

Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, Aug 1989. ISSN 1436-4646.

Ji Liu, Stephen J. Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. *Journal of Machine Learning Research*, 16(1):285–322, January 2015. ISSN 1532-4435.

H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 27(4):2202–2229, 2017.

E. Min, Y. Zhao, J. Long, C. Wu, K. Li, and J. Yin. Svrg with adaptive epoch size. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2935–2942, May 2017.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer Publishing Company, Incorporated, 1 edition, 2014. ISBN 1461346916, 9781461346913.

Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A Novel Method for Machine Learning Problems Using Stochastic Recursive Gradient. *arXiv e-prints*, art. arXiv:1703.00102, February 2017.

L. Nicolas, M. Schmidt, and R Francis. A stochastic gradient method with an exponential convergence rate for finite training sets. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2663–2671. Curran Associates, Inc., 2012.

Feng Niu, Benjamin Recht, Christopher Re, and Stephen J. Wright. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 693–701, USA, 2011. Curran Associates Inc. ISBN 978-1-61839-599-3.

Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980. ISSN 00255718, 10886842.

Gregory B Passty. Ergodic convergence to a zero of the sum of monotone operators in hilbert space. *Journal of Mathematical Analysis and Applications*, 72(2):383 – 390, 1979. ISSN 0022-247X.

J. Reddi, A. Hefny, S. Sra, B. Poczos, and A. Smola. On variance reduction in stochastic gradient descent and its asynchronous variants. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2647–2655. Curran Associates, Inc., 2015.

H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 09 1951.

Lorenzo Rosasco, Silvia Villa, and Băng Công Vũ. A stochastic inertial forward–backward splitting algorithm for multivariate monotone inclusions. *Optimization*, 65(6):1293–1314, 2016.

E. Ryu and S. Boyd. A primer on monotone operator methods survey. *Applied and computational mathematics*, 15:3–43, 01 2016.

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, Mar 2017. ISSN 1436-4646.

T. Tatarenko and M. Kamgarpour. Learning generalized nash equilibria in a class of convex games. *IEEE Transactions on Automatic Control*, 64(4):1426–1439, 2019. doi: 10.1109/TAC.2018.2841319.

Tatiana Tatarenko and Maryam Kamgarpour. Learning generalized nash equilibria in a class of convex games. *IEEE Transactions on Automatic Control*, 64(4):1426–1439, 2018.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Appendix A. Proofs for Section 2 (forward-backward splitting)

A.1 Proof of Theorem 2 (linear convergence rate)

We have $x^* = (I+\gamma A)^{-1}(I-\gamma B)(x^*)$ for all $\gamma > 0$ since x^* is the unique solution of (1). Then, we have

$$\begin{aligned}\|x_{k+1} - x^*\|^2 &= \|(I+\gamma A)^{-1}(I-\gamma B)(x_k) - (I+\gamma A)^{-1}(I-\gamma B)(x^*)\|^2 \\ &\leq \|(I-\gamma B)(x_k) - (I-\gamma B)(x^*)\|^2 \\ &= \|x_k - x^*\|^2 - 2\,\gamma(B(x_k) - B(x^*))^\top (x_k - x^*) + \gamma^2 \|B(x_k) - B(x^*)\|^2 \\ &\leq (1 - 2\gamma\mu + \gamma^2 L^2)\|x_k - x^*\|^2,\end{aligned}$$

where the first inequality uses non-expansiveness of the resolvent $(I+\gamma A)^{-1}$, and the second inequality uses μ-strong monotonicity and L-Lipschitz continuity of B.

Appendix B. Proofs for Section 3 (randomized forward-backward splitting)

B.1 Proof of Lemma 4 (one-step expected error)

Using the shorthand $\mathcal{G}_k = \mathcal{G}\big(x_k,\, \phi^k, I_k\big)$, we have

$$\begin{aligned}
\|x_{k+1} - x^*\|^2 &= \|(I+\gamma_k A)^{-1}(x_k - \gamma_k \mathcal{G}_k) - (I+\gamma_k A)^{-1}(x^* - \gamma_k B(x^*))\|^2 \\
&\leq \|(x_k - \gamma_k \mathcal{G}_k) - (x^* - \gamma_k B(x^*))\|^2 \\
&= \|(x_k - x^*) - \gamma_k \left(B(x_k) - B(x^*)\right) - \gamma_k \left(\mathcal{G}_k - B(x_k)\right)\|^2 \\
&= \|x_k - x^*\|^2 + \gamma_k^2 \|B(x_k) - B(x^*)\|^2 + \gamma_k^2 \|\mathcal{G}_k - B(x_k)\|^2 \\
&\quad - 2\gamma_k (B(x_k) - B(x^*))^T (x_k - x^*) - 2\gamma_k (x_k - x^*)^T \left(\mathcal{G}_k - B(x_k)\right) \\
&\quad + 2\gamma_k^2 (B(x_k) - B(x^*))^T \left(\mathcal{G}_k - B(x_k)\right),
\end{aligned}$$

where the last equality is by expansion of the squared norm. Using unbiasedness of $\mathcal{G}_k$, we see that

$$\begin{aligned}
\mathbb{E}\left[\|x_{k+1} - x^*\|^2 \,|\mathcal{F}_k\right] \leq \|x_k - x^*\|^2 &- 2\gamma_k (B(x_k) - B(x^*))^T (x_k - x^*) \\
&+ \gamma_k^2 \|B(x_k) - B(x^*)\|^2 + \gamma_k^2 \mathbb{E}\left[\|\mathcal{G}_k - B(x_k)\|^2 | \mathcal{F}_k\right].
\end{aligned}$$

By strong monotonicity and Lipschitz continuity of B, we can bound the RHS of the above display with

$$\mathbb{E}\left[\|x_{k+1} - x^*\|^2 \,|\mathcal{F}_k\right] \leq (1 - 2\gamma_k \mu + \gamma_k^2 L^2) \|x_k - x^*\|^2 + \gamma_k^2 \mathbb{E}\left[\|\mathcal{G}_k - B(x_k)\|^2 | \mathcal{F}_k\right].$$

Taking expectations of both sides gives the desired conclusion.

B.2 Proof of Lemma 5 (bound on conditional variance)

Direct calculation shows that

$$\begin{aligned}
\mathbb{E}\left[\|\mathcal{G}\big(x_k,\, \phi^k, I_k\big) - B(x_k)\|^2 | \mathcal{F}_k\right] &= \mathbb{E}\left[\left\|B_{I_k}(x_k) - \phi^k_{I_k} - \mathbb{E}\left[B_{I_k} - \phi^k_{I_k} | \mathcal{F}_k\right]\right\|^2 \Big| \mathcal{F}_k\right] \\
&\leq \mathbb{E}\left[\left\|B_{I_k}(x_k) - \phi^k_{I_k}\right\|^2 \Big| \mathcal{F}_k\right],
\end{aligned}$$

where the inequality follows because the conditional variance of a random variable is not larger than its second moment. By adding and subtracting $B_{I_k}(x^*)$, we can further bound the last term of the above display with

$$\begin{aligned}
\mathbb{E}\left[\|B_{I_k}(x_k) - \phi^k_{I_k}\|^2 | \mathcal{F}_k\right] &= \mathbb{E}\left[\left\|\Big(B_{I_k}(x_k) - B_{I_k}(x^*)\Big) - \Big(\phi^k_{I_k} - B_{I_k}(x^*)\Big)\right\|^2 \Big| \mathcal{F}_k\right] \\
&\leq 2\,\mathbb{E}\left[\|(B_{I_k}(x_k) - B_{I_k}(x^*)\|^2 + \|\phi^k_{I_k} - B_{I_k}(x^*)\|^2 | \mathcal{F}_k\right] \\
&\leq 2\left(L^2 \|x_k - x^*\|^2 + \frac{1}{n}\sum_{i=1}^n \|\phi^k_i - B_i(x^*)\|^2\right),
\end{aligned}$$

by the triangle inequality. The first term on the RHS of the last inequality is due to L-Lipschitz continuity of each B_i. The second term on the RHS of the last inequality is the conditional expectation of $\|\phi_{I_k}^k - B_{I_k}(x^*)\|^2$ with respect to $\mathcal{F}_k$. Take full expectation of both sides of the above display to conclude.

B.3 Proof of Theorem 7 (linear convergence rate)

We first show that $\theta < 1$. For the choices of γ and ρ given in the theorem, θ is the maximum of ℓ_1 and ℓ_2 where

$$\ell_1 \triangleq 1 - 2\gamma\mu + 3\gamma^2 L^2 + c_2\gamma^{1.5}, \quad \ell_2 \triangleq 2\gamma^{0.5} + c_1.$$

Showing that $\theta < 1$ is achieved by showing that both ℓ_1 and ℓ_2 are strictly less than one. By decreasing the denominators of both terms inside the brackets of (12), we have

$$\gamma < \min\left\{ \left(\frac{2\mu}{\frac{3(1-c_1)L^2}{2} + c_2} \right)^2, \ \frac{(1-c_1)^2}{4} \right\}. \tag{30}$$

It is then readily seen that $\ell_2 < 1$ because $\gamma < (1-c_1)^2/4$. To show $\ell_1 < 1$, note that

$$3\gamma L^2 + c_2\gamma^{0.5} = \gamma^{0.5}\left(3\gamma^{0.5}L^2 + c_2\right) < \frac{2\mu}{\frac{3(1-c_1)L^2}{2} + c_2}\left(\frac{3(1-c_1)L^2}{2} + c_2\right) = 2\mu,$$

where we use the fact that γ is smaller than the first term in the bracket of (30) to get the first term in the last inequality, and the fact that γ is smaller than the second term in the bracket of (30) to get the second term in the last inequality. By expressing ℓ_1 as $1 - \gamma(2\mu - 3\gamma L^2 - c_2\gamma^{0.5})$, it is readily see that $\ell_1 < 1$ and so $\theta < 1$.

Next, we establish inequality (13). Based on Corollary 6 and Assumption 2, we have

$$\begin{aligned}
\mathbb{E}L_\rho\left(\tilde{x}_k, \tilde{\phi}^k\right) &= \mathbb{E}\|x_{S_k} - x^*\|^2 + \rho\mathbb{E}G\left(\phi^{S_k}\right) \\
&\leq \left(1 - 2\gamma\mu + 3\gamma^2L^2 + c_2\rho\right)\mathbb{E}\|x_{S_k-1} - x^*\|^2 \\
&\quad + \left(\frac{2\gamma^2}{\rho} + c_1\right)\rho\mathbb{E}G\left(\phi^{S_k-1}\right) + 2\gamma^2\mathbb{E}H\left(\phi^{S_k-1}\right) \\
&\leq \max\left\{1 - 2\gamma\mu + 3\gamma^2L^2 + c_2\rho, \ \frac{2\gamma^2}{\rho} + c_1\right\}\mathbb{E}L_\rho\left(x_{S_k-1}, \phi^{S_k-1}\right) \\
&\quad + 2\gamma^2\mathbb{E}H\left(\phi^{S_k-1}\right).
\end{aligned} \tag{31}$$

Using the definition of θ, the above display is equivalent to $\mathbb{E}L_\rho\left(\tilde{x}_k, \tilde{\phi}^k\right) \leq \theta\mathbb{E}L_\rho\left(x_{S_k-1}, \phi^{S_k-1}\right) + 2\gamma^2\mathbb{E}H\left(\phi^{S_k-1}\right)$. By recursively applying this inequality, we obtain

$$\begin{aligned}
\mathbb{E}L_\rho\left(\tilde{x}_k, \tilde{\phi}^k\right) &\leq \theta^{m_k}\mathbb{E}L_\rho\left(x_{S_{k-1}}, \phi^{S_{k-1}}\right) + 2\gamma^2\mathbb{E}\sum_{i=0}^{m_k-1}\theta^i H\left(\phi^{S_k-1-i}\right) \\
&\leq \left(\theta^{m_k} + 2\gamma^2 c_3\sum_{i=0}^{m_k-1}\theta^i\right)\mathbb{E}L_\rho\left(x_{S_{k-1}}, \phi^{S_{k-1}}\right),
\end{aligned}$$

where the last inequality is due to Assumption 2.2. Because $\theta < 1$ and $m_k \leq \bar{m}$ for all $k \geq 0$, the above display implies

$$\mathbb{E}L_\rho\left(x_{S_k}, \phi^{S_k}\right) \leq \left(\theta + 2\gamma^2\bar{m}c_3\right)\mathbb{E}L_\rho\left(x_{S_{k-1}}, \phi^{S_{k-1}}\right) = \lambda\mathbb{E}L_\rho\left(x_{S_{k-1}}, \phi^{S_{k-1}}\right).$$

We complete the proof by showing that $\lambda < 1$. Using θ as defined in (10) and λ as defined in (11), we may rewrite λ as

$$\lambda = \max\left\{1 - 2\gamma\mu + 3\gamma^2L^2 + c_2\rho + 2\gamma^2\bar{m}c_3, \frac{2\gamma^2}{\rho} + c_1 + 2\gamma^2\bar{m}c_3\right\}.$$

With $\rho = \gamma^{1.5}$, λ is the maximum of $\tilde{\ell}_1$ and $\tilde{\ell}_2$ where

$$\tilde{\ell}_1 \triangleq 1 - 2\gamma\mu + 3\gamma^2L^2 + c_2\gamma^{1.5} + 2\gamma^2\bar{m}c_3, \quad \tilde{\ell}_2 \triangleq 2\gamma^{0.5} + 2\gamma^2\bar{m}c_3 + c_1.$$

Now it suffices to show that both $\tilde{\ell}_1$ and $\tilde{\ell}_2$ are strictly less than one. Starting with $\tilde{\ell}_2$, we have

$$2\gamma^{0.5} + 2\bar{m}\gamma^2c_3 = \gamma^{0.5}(2 + 2\bar{m}\gamma^{1.5}c_3) < \frac{1 - c_1}{2 + 2\bar{m}(\frac{1-c_1}{2})^3c_3}\left(2 + 2\bar{m}\gamma^{1.5}c_3\right), \tag{32}$$

where the inequality follows because γ is less than the second term in the bracket of (12). Since $\gamma \leq (1 - c_1)^2/4$ by (30), the RHS of (32) is less than $1 - c_1$. From (32) we have that $2\gamma^{0.5} + 2\bar{m}\gamma^2c_3 < 1 - c_1$, and thus $\tilde{\ell}_2 < 1$. To show $\tilde{\ell}_1 < 1$, note that

$$\begin{aligned}
3\gamma L^2 + c_2\gamma^{0.5} + 2\gamma\bar{m}c_3 &= \gamma^{0.5}\left(3\gamma^{0.5}L^2 + 2\bar{m}c_3\gamma^{0.5} + c_2\right) \\
&< \frac{2\mu}{\frac{3(1-c_1)L^2}{2} + (1 - c_1)c_3\bar{m} + c_2}\left(3\gamma^{0.5}L^2 + 2\bar{m}c_3\gamma^{0.5} + c_2\right),
\end{aligned}$$

where the inequality is because γ is less than the first term in the bracket (12). Since $\gamma \leq (1 - c_1)^2/4$, the above display is less than 2μ. Finally, rewrite $\tilde{\ell}_1$ as $1 + \gamma(3\gamma L^2 + c_2\gamma^{0.5} + 2\gamma\bar{m}c_3 - 2\mu)$ to see that $\tilde{\ell}_1 < 1$.

Now we consider the distance to the optimal solution. We have that $\lambda \in [0, 1)$, so recursively applying inequality (13) and using Lipschitz continuity of B_i gives

$$\begin{aligned}
\mathbb{E}L_\rho\left(\tilde{x}_k, \tilde{\phi}^k\right) &\leq \lambda^k\mathbb{E}L_\rho\left(\tilde{x}_0, \tilde{\phi}^0\right) \\
&= \lambda^k\left(\|x_0 - x^*\|^2 + \frac{\rho}{n}\sum_{i\in\mathcal{S}}\|B_i(x_0) - B_i(x^*)\|^2\right) \\
&\leq \lambda^k\left(1 + \rho L^2\right)\|x_0 - x^*\|^2.
\end{aligned}$$

Finally, we use the definition of L_ρ to conclude that $\mathbb{E}\|\tilde{x}_k - x^*\|^2 \leq \mathbb{E}L_\rho\left(\tilde{x}_k, \tilde{\phi}^k\right) \leq \lambda^k\left(1 + \rho L^2\right)\|x_0 - x^*\|^2$.

Appendix C. Proofs for Section 4 (Catalyst)

C.1 Proof of Lemma 8 (one-step expected error)

By definition of $x^*(\bar{x})$, we have $A(x^*(\bar{x})) + B(x^*(\bar{x})) + \sigma x^*(\bar{x}) = \sigma \check{x}_k$ and thus $x^*(\bar{x}) = (A+B+\sigma I)^{-1}(\sigma \check{x}_k)$ for all $\sigma \geq 0$. Similarly, we have $x^* = (A+B+\sigma I)^{-1}(\sigma x^*)$ for all $\sigma \geq 0$. Using these two equalities we may bound $\mathbb{E}\|\check{x}_{k+1} - x^*(\bar{x})\|^2$ with

$$\begin{aligned}
&\mathbb{E}\|\check{x}_{k+1} - (A+B+\sigma I)^{-1}(\sigma \check{x}_k)\|^2 \\
\leq &\frac{1}{4(1+\sigma/\mu)^2}\mathbb{E}\|\check{x}_k - x^* - (A+B+\sigma I)^{-1}(\sigma \check{x}_k) + (A+B+\sigma I)^{-1}(\sigma x^*)\|^2 \\
= &\frac{1}{4(1+\sigma/\mu)^2}\mathbb{E}\|\check{x}_k - x^* - \left((A+B)\sigma^{-1}+I\right)^{-1}(\check{x}_k) + \left((A+B)\sigma^{-1}+I\right)^{-1}(x^*)\|^2 \\
= &\frac{1}{4(1+\sigma/\mu)^2}\mathbb{E}\| \left(I - \left((A+B)\,\sigma^{-1}+I\right)^{-1}\right)(\check{x}_k) - \left(I - \left((A+B)\,\sigma^{-1}+I\right)^{-1}\right)(x^*)\|^2,
\end{aligned} \tag{33}$$

where the second line follows by telescoping with x^*.

Now, for any maximal monotone operator C, $I - (I+C)^{-1} = (I + C^{-1})^{-1}$. To prove this statement, suppose $y + C(y) = x$. Then $x - (I+C)^{-1}(x) = x - y = C(y)$. On the other hand, $C(y) + C^{-1}(C(y)) = y + C(y) = x$, and so $(I + C^{-1})^{-1}(x) = C(y)$. Then $I - (I+C)^{-1} = (I+C^{-1})^{-1}$, and so (33) is equivalent to

$$\begin{aligned}
&\mathbb{E}\|\check{x}_{k+1} - (A+B+\sigma I)^{-1}(\sigma \check{x}_k)\|^2 \\
\leq &\frac{1}{4(1+\sigma/\mu)^2}\mathbb{E}\| \left(I + (A+B)^{-1}\,\sigma\right)^{-1}(\check{x}_k) - \left(I + (A+B)^{-1}\,\sigma\right)^{-1}(x^*)\|^2.
\end{aligned}$$

It is immediate $(A+B)^{-1}\,\sigma$ is monotone since the inverse of a monotone operator is monotone. In addition, $\left(I + (A+B)^{-1}\,\sigma\right)^{-1}$ is non-expansive since it is the resolvent of $(A+B)^{-1}\,\sigma$ and the resolvent of a monotone operator is always non-expansive (Ryu and Boyd, 2016). As a result, we obtain

$$\mathbb{E}\|\check{x}_{k+1} - (A+B+\sigma I)^{-1}(\sigma \check{x}_k)\|^2 \leq \frac{1}{4(1+\sigma/\mu)^2}\mathbb{E}\|\check{x}_k - x^*\|^2. \tag{34}$$

Thus, we can use Minkowski's inequality to see

$$\begin{aligned}
\left(\mathbb{E}\|\check{x}_{k+1} - x^*\|^2\right)^{1/2} \leq &\left(\mathbb{E}\|\check{x}_{k+1} - (A+B+\sigma I)^{-1}(\sigma \check{x}_k)\|^2\right)^{1/2} \\
&+ \left(\mathbb{E}\|\,(A+B+\sigma I)^{-1}(\sigma \check{x}_k) - x^*\|^2\right)^{1/2} \\
\leq &\frac{1}{2(1+\sigma/\mu)}\left(\mathbb{E}\|\check{x}_k - x^*\|^2\right)^{1/2} + \\
&+ \left(\mathbb{E}\|\left(\sigma^{-1}(A+B)+I\right)^{-1}(\check{x}_k) - \left(\sigma^{-1}(A+B)+I\right)^{-1}(x^*)\|^2\right)^{1/2},
\end{aligned}$$

where the first term on the RHS of the second inequality is by (34) and the second term on the RHS of the second inequality is by routine reformulation (using $(A+B+\sigma I)^{-1}(\sigma x) =$

$\left(\sigma^{-1}(A+B)+I\right)^{-1}(x)$ for all x). Since $\sigma^{-1}(A+B)$ is μ/σ-strongly monotone by Assumption 1, $\left(\sigma^{-1}(A+B)+I\right)^{-1}$ is $\frac{1}{1+\sigma^{-1}\mu}$-Lipschitz, and so we can rewrite the last inequality in the above display as

$$\begin{aligned}\left(\mathbb{E}\|\check{x}_{k+1}-x^*\|^2\right)^{1/2} \leq & \frac{1}{2(1+\sigma/\mu)}\left(\mathbb{E}\|\check{x}_k-x^*\|^2\right)^{1/2}+\frac{\sigma}{\sigma+\mu}\left(\mathbb{E}\|\check{x}_k-x^*\|^2\right)^{1/2} \\ = & \left(1-\frac{1}{2(1+\sigma/\mu)}\right)\left(\mathbb{E}\|\check{x}_k-x^*\|^2\right)^{1/2}.\end{aligned}$$

The desired result follows by squaring both sides.

Appendix D. Proofs for Section 5 (asynchronous implementation)

D.1 Proof of Lemma 10 (one-step expected error)

Use the definition of x_{k+1} to see that

$$\begin{aligned}\|x_{k+1}-x^*\|^2 &= \|x_k-x^*-\gamma\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k)\|^2 \\ &= \|x_k-x^*\|^2-2\gamma(\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k))^\top(x_k-x^*)+\gamma^2\|\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k)\|^2 \\ &= \|x_k-x^*\|^2-2\gamma(\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k))^\top(\hat{x}_k-x^*)+\gamma^2\|\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k)\|^2 \\ &\quad +2\gamma(\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k))^\top(\hat{x}_k-x_k).\end{aligned}$$

Here, the second equality is from expansion of the squared norm and the third equality is by adding and subtracting $\hat{x}_k$. Due to the unbiasedness of $\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k)$ and the strong monotonicity of B, we have

$$\mathbb{E}\left[(\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k))^\top(\hat{x}_k-x^*)|\mathcal{F}_k\right]=(B(\hat{x}_k)-B(x^*))^\top(\hat{x}_k-x^*)\geq\mu\|\hat{x}_k-x^*\|^2. \tag{35}$$

Use the triangle inequality to see

$$\frac{\mu}{2}\|x_k-x^*\|^2=\frac{\mu}{2}\|x_k-\hat{x}_k+\hat{x}_k-x^*\|^2\leq\mu\left(\|x_k-\hat{x}_k\|^2+\|\hat{x}_k-x^*\|^2\right),$$

and as a result we have

$$\mu\|\hat{x}_k-x^*\|^2\geq\frac{\mu}{2}\|x_k-x^*\|^2-\mu\|x_k-\hat{x}_k\|^2. \tag{36}$$

Combine (35) and (36) to get

$$\begin{aligned}\mathbb{E}\left[\|x_{k+1}-x^*\|^2|\mathcal{F}_k\right]\leq & (1-\gamma\mu)\|x_k-x^*\|^2+2\gamma\mathbb{E}\left[(\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k))^\top(\hat{x}_k-x_k)|\mathcal{F}_k\right] \\ & +2\gamma\mu\|\hat{x}_k-x_k\|^2+\gamma^2\mathbb{E}\left[\|\mathcal{G}(\hat{x}_k,\hat{\phi}^k,I_k)\|^2|\mathcal{F}_k\right],\end{aligned}$$

then take expectations of both sides of the above inequality to establish (20).

D.2 Proof of Lemma 11 (error bounds)

We have $\|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)\|^2 \leq M^2$ by the definition of M. Since $\hat{x}_k = x_{D(k)}$, we may use (19) to see that

$$\|x_k - \hat{x}_k\|^2 = \gamma^2 \|\mathcal{G}(\hat{x}_{D(k)}, \hat{\phi}^{D(k)}, I_{D(k)}) + \cdots + \mathcal{G}(\hat{x}_{k-1}, \hat{\phi}^{k-1}, I_{k-1})\|^2 \leq \gamma^2 \tau^2 M^2. \tag{37}$$

The inequality above is due to the fact that the number of terms inside the norm is not greater than τ from Assumption 3. To bound $(\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k))^\top (\hat{x}_k - x_k)$, we use (37) to see that

$$(\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k))^\top (\hat{x}_k - x_k) \leq \|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)\| \|\hat{x}_k - x_k\| \leq \gamma \tau M^2.$$

The first inequality above is from the Cauchy-Schwarz inequality, and the second is from (37) and the fact $\|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)\| \leq M$ for all $k \geq 0$.

We now bound $\mathbb{E}\|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)\|^2$. Use the fact that $B(x^*) = 0$ to see that

$$\begin{aligned}
\mathbb{E}\left[\|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)\|^2 | \mathcal{F}_k\right] &= \mathbb{E}\left[\|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k) - B(\hat{x}_k) + B(\hat{x}_k) - B(x^*)\|^2 | \mathcal{F}_k\right] \\
&= \mathbb{E}\left[\|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k) - B(\hat{x}_k)\|^2 | \mathcal{F}_k\right] + \|B(\hat{x}_k) - B(x^*)\|^2 \\
&\leq \mathbb{E}\left[\|B_{I_k}(\hat{x}_k) - \hat{\phi}^k_{I_k}\|^2 | \mathcal{F}_k\right] + L^2 \|\hat{x}_k - x^*\|^2,
\end{aligned}$$

where the second equality above is due to the unbiasedness of $\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)$. By adding and subtracting $B_{I_k}(x^*)$ in the first term of the last inequality, we can further bound the above display with

$$\begin{aligned}
\mathbb{E}\left[\|\mathcal{G}(\hat{x}_k, \hat{\phi}^k, I_k)\|^2 | \mathcal{F}_k\right] &\leq \mathbb{E}\left[\|B_{I_k}(\hat{x}_k) - B_{I_k}(x^*) + B_{I_k}(x^*) - \hat{\phi}^k_{I_k}\|^2 | \mathcal{F}_k\right] + L^2 \|\hat{x}_k - x^*\|^2 \\
&\leq 3L^2 \|\hat{x}_k - x^*\|^2 + 2\mathbb{E}\left[\|B_{I_k}(x^*) - \hat{\phi}^k_{I_k}\|^2 | \mathcal{F}_k\right] \\
&\leq 3L^2 (2\|\hat{x}_k - x_k\|^2 + 2\|x_k - x^*\|^2) \\
&\quad + 2\left(2\mathbb{E}\left[\|B_{I_k}(x^*) - \phi^k_{I_k}\|^2 | \mathcal{F}_k\right] + 2\mathbb{E}\left[\|\phi^k_{I_k} - \hat{\phi}^k_{I_k}\|^2 | \mathcal{F}_k\right]\right) \\
&= 6L^2 \|\hat{x}_k - x_k\|^2 + 6L^2 \|x_k - x^*\|^2 + \frac{4}{n}\sum_{i=1}^n \|\phi^k_i - B_i(x^*)\|^2 + \\
&\quad \frac{4}{n}\sum_{i=1}^n \|\phi^k_i - \hat{\phi}^k_i\|^2.
\end{aligned}$$

Here, the second and third inequalities are due to the triangle inequality. The last equality is by direct calculation of the conditional expectation. Use the fact that $\mathbb{E}\|x_k - \hat{x}_k\|^2 \leq \gamma^2 \tau^2 M^2$ and take expectations to conclude.

D.3 Proof of Theorem 13 (asynchronous convergence rates)

We first show that $\theta < 1$. With the choices of γ and ρ given in the theorem, we have that θ is the maximum of ℓ_1 and ℓ_2 where

$$\ell_1 \triangleq 1 - \gamma\mu + 6\gamma^2 L^2 + c_2 \gamma^{1.5}, \quad \ell_2 \triangleq 4\gamma^{0.5} + c_1.$$

It suffices to show that both ℓ_1 and ℓ_2 are strictly less than one. By decreasing the denominators of both terms inside the bracket of (22), we have

$$\gamma < \min\left\{\left(\frac{\mu}{\frac{3(1-c_1)L^2}{2}+c_2}\right)^2,\ \left(\frac{1-c_1}{4}\right)^2\right\}. \tag{38}$$

It is readily seen that $\ell_2 < 1$ because $\gamma < (1-c_1)^2/4$ by the above display. Similar to the proof of Theorem 7, we can use (38) to show $\mu - 6\gamma L^2 - c_2\gamma^{0.5} > 0$. By expressing ℓ_1 as $1-\gamma(\mu - 6\gamma L^2 - c_2\gamma^{0.5})$, we have $\ell_1 < 1$ and it follows that $\theta < 1$.

Next, we establish inequality (23). Based on Assumption 4 and Corollary 12, we have

$$\begin{aligned}
\mathbb{E}L_\rho\left(\tilde{x}_k, \tilde{\phi}^k\right) &= \mathbb{E}\left[\|x_{S_k} - x^*\|^2 + \rho G\left(\phi^{S_k}\right)\right] \\
&\le \left(1-\gamma\mu+6\gamma^2L^2+c_2\rho\right)\mathbb{E}\|x_{S_k-1}-x^*\|^2 + \left(\frac{4\gamma^2}{\rho}+c_1\right)\rho\mathbb{E}G\left(\phi^{S_k-1}\right) \\
&\quad + 4\gamma^2\mathbb{E}H\left(\phi^{S_k-1}\right) + \underbrace{4\gamma^2\mathcal{E}_2+\mathcal{E}_0+\rho\mathcal{E}_1}_{\mathcal{E}_3} \\
&\le \max\left\{1-\gamma\mu+6\gamma^2L^2+c_2\rho,\ \frac{4\gamma^2}{\rho}+c_1\right\}\mathbb{E}L_\rho\left(x_{S_k-1}, \phi^{S_k-1}\right) \\
&\quad + 4\gamma^2\mathbb{E}H(\phi^{S_k-1}) + \mathcal{E}_3.
\end{aligned}$$

Using the definition of θ, the above display is equivalent to

$$\mathbb{E}L_\rho\left(\tilde{x}_k, \tilde{\phi}^k\right) \le \theta\mathbb{E}L_\rho\left(x_{S_k-1}, \phi^{S_k-1}\right) + 4\gamma^2\mathbb{E}H\left(\phi^{S_k-1}\right) + \mathcal{E}_3.$$

By recursively applying the above inequality and using the facts that $\theta < 1$ and $m_k \le \bar{m}$ for all $k \ge 0$, we have

$$\begin{aligned}
\mathbb{E}L_\rho\left(x_{S_k}, \phi^{S_k}\right) &\le \left(\theta + 4\gamma^2\bar{m}c_3\right)\mathbb{E}L_\rho\left(x_{S_{k-1}}, \phi^{S_{k-1}}\right) + \sum_{i=0}^{m-1}\theta^i\mathcal{E}_3 \\
&\le \lambda\mathbb{E}L_\rho\left(x_{S_{k-1}}, \phi^{S_{k-1}}\right) + \frac{\mathcal{E}_3}{1-\theta}.
\end{aligned}$$

We finish the proof by showing that $\lambda < 1$. Using the definition of θ, we may rewrite λ as

$$\lambda = \max\left\{1-\gamma\mu+6\gamma^2L^2+c_2\rho+4\gamma^2\bar{m}c_3, \frac{4\gamma^2}{\rho}+c_1+4\gamma^2\bar{m}c_3\right\}.$$

With $\rho = \gamma^{1.5}$, λ is the maximum of $\tilde{\ell}_1$ and $\tilde{\ell}_2$ where

$$\tilde{\ell}_1 \triangleq 1-\gamma\mu+6\gamma^2L^2+c_2\gamma^{1.5}+4\gamma^2\bar{m}c_3, \quad \tilde{\ell}_2 \triangleq 4\gamma^{0.5}+4\gamma^2\bar{m}c_3+c_1.$$

It suffices to prove that both $\tilde{\ell}_1$ and $\tilde{\ell}_2$ are strictly less than one. We may rewrite $\tilde{\ell}_2$ as $c_1+4\gamma^{0.5}(1+\gamma^{1.5}\bar{m}c_3)$. Then, similar to the proof of Theorem 7, we can use the fact that γ

is less than the second term in the brackets of (22) and (38) to see that $\tilde{\ell}_2 < 1$. To establish $\tilde{\ell}_1 < 1$, note that

$$\begin{aligned} 6\gamma L^2 + c_2\gamma^{0.5} + 4\gamma\bar{m}c_3 &= \gamma^{0.5}\left(6\gamma^{0.5}L^2 + 4\bar{m}c_3\gamma^{0.5} + c_2\right) \\ &< \frac{\mu}{\frac{3(1-c_1)L^2}{2} + (1-c_1)c_3\bar{m} + c_2}\left(6\gamma^{0.5}L^2 + 4\bar{m}c_3\gamma^{0.5} + c_2\right), \end{aligned}$$

where the inequality follows because γ is less than the first term in the brackets of (22). Since $\gamma < (1-c_1)^2/16$, the above display is less than μ. Finally, rewrite $\tilde{\ell}_1$ as $1+\gamma(6\gamma L^2 + c_2\gamma^{0.5} + 4\gamma\bar{m}c_3 - \mu)$ to see that $\tilde{\ell}_1 < 1$.

Appendix E. Technical details for Section 6 (examples)

E.1 Technical details for SVRG

Basic complexity First define the constants $q \triangleq 1 - 2\gamma\mu + 3\gamma^2L^2$ and $p \triangleq 2\gamma^2L^2$. Recall $\tilde{x}_k = x_{km}$, so $\phi_i^t = B_i(\tilde{x}_k)$ for all $km \le t < (k+1)m$ and $i \in [n]$. We choose $\gamma = \mu/(3L^2)$ so that $q = 1 - 1/3\kappa^2$. Then, Corollary 6 implies

$$\mathbb{E}\|\tilde{x}_{k+1} - x^*\| = \mathbb{E}\|x_{(k+1)m} - x^*\|^2 \le p\mathbb{E}\|x_{(k+1)m-1} - x^*\|^2 + q\mathbb{E}\|\tilde{x}_k - x^*\|^2.$$

Recursively applying the above inequality shows that

$$\begin{aligned} \mathbb{E}\|\tilde{x}_{k+1} - x^*\|^2 \le& q^m\mathbb{E}\|x_{km} - x^*\|^2 + p(1 + q + \cdots + q^{m-1})\mathbb{E}\|\tilde{x}_k - x^*\|^2 \\ &\le \left(q^m + \frac{p}{1-q}\right)\mathbb{E}\|\tilde{x}_k - x^*\|^2, \end{aligned}$$

where the second inequality follows because $q = 1 - 1/3\kappa^2 < 1$. Furthermore, by routine calculation we have

$$q^m + \frac{p}{1-q} = \left(1 - \frac{1}{3\kappa^2}\right)^m + \frac{2}{3}.$$

We choose $m = \frac{\log\frac{1}{12}}{\log\left(1-\frac{1}{3\kappa^2}\right)}$ so that $q^m + \frac{p}{1-q} = \frac{3}{4}$, that is, the expected distance to x^* will contract by a factor of $3/4$ after every epoch. In large-scale problems, κ is large and so $m = O(\kappa^2)$ due to the fact that $\log(1+x) \approx x$ for small x. In this case, the total complexity is $O\left(n + \kappa^2\right)\log(1/\epsilon)$.

Catalyst complexity According to (17), the total complexity of SVRG with Catalyst acceleration is

$$O\left(\frac{\sigma+\mu}{\mu}\left(n + \left(\frac{L+\sigma}{\mu+\sigma}\right)^2\right)\log\left(\frac{2(\mu+\sigma)}{\mu}\right)\log\left(1/\epsilon\right)\right).$$

Letting $\lambda \triangleq \sigma/\mu$, the above term can be rewritten as

$$O\left((\lambda+1)\left(n + \left(\frac{\kappa+\lambda}{1+\lambda}\right)^2\right)\log\left(2(1+\lambda)\right)\log\left(\frac{1}{\epsilon}\right)\right).$$

We omit the logarithmic terms and only consider $O\left((\lambda+1)\left(n+\left(\frac{\kappa+\lambda}{1+\lambda}\right)^2\right)\right)$. We define $f(\lambda) \triangleq (\lambda+1)\left(n+\left(\frac{\kappa+\lambda}{1+\lambda}\right)^2\right)$ and optimize over $\lambda \geq 0$ to achieve the optimal complexity. It is straightforward to verify that $f(\lambda)$ attains its minimum at $\lambda^* = \sqrt{\frac{(\kappa-1)^2-2}{n+1}}$ where $f(\lambda^*) = 2\sqrt{(n+1)\left((\kappa-1)^2-2\right)} + 2\kappa - 2$, which is $O(\kappa\sqrt{n})$.

E.2 Technical details for SAGA

Basic complexity Recall from (31) that

$$\mathbb{E}L_\rho(x_k, \phi^k) \leq \max\left\{1 - 2\gamma\mu + 3\gamma^2L^2 + \frac{\rho L^2}{n}, \frac{2\gamma^2}{\rho} + 1 - \frac{1}{n}\right\}\mathbb{E}L_\rho(x_{k-1}, \phi^{k-1}),\ \forall k \geq 0. \tag{39}$$

By choosing $\rho = 4\gamma^2 n$ and $\gamma = \mu/(7L^2)$, the above display becomes

$$\mathbb{E}L_\rho\left(x_{k+1}, \phi^{k+1}\right) \leq \max\left\{1 - \frac{1}{7\kappa^2}, 1 - \frac{1}{2n}\right\} L_\rho\left(x_k, \phi^k\right).$$

Then, after

$$k = \max\left\{\frac{\log\left(\epsilon/L(x_0,\phi^0)\right)}{\log\left(1-\frac{1}{7\kappa^2}\right)}, \frac{\log\left(\epsilon/L(x_0,\phi^0)\right)}{\log\left(1-\frac{1}{2n}\right)}\right\} = O\left(\max\{n,\kappa^2\}\right)\log\left(\frac{1}{\epsilon}\right)$$

iterations, we have $\mathbb{E}L_\rho\left(x_k, \phi^k\right) \leq \epsilon$. We require two operator evaluations for each of the k iterations, and so the overall complexity is $O\left(\max\{n,\kappa^2\}\right)\log(1/\epsilon)$.

Catalyst complexity The total complexity of SAGA with Catalyst acceleration is

$$O\left(\frac{\sigma+\mu}{\mu}\max\left\{n, \left(\frac{L+\sigma}{\mu+\sigma}\right)^2\right\}\log\left(\frac{2(\mu+\sigma)}{\mu}\right)\log\left(\frac{1}{\epsilon}\right)\right).$$

Let $\lambda \triangleq \sigma/\mu$ and rewrite the above display as

$$O\left((\lambda+1)\max\left\{n, \left(\frac{\kappa+\lambda}{1+\lambda}\right)^2\right\}\log\left(2(\lambda+1)\right)\log\left(\frac{1}{\epsilon}\right)\right).$$

We minimize $f(\lambda) \triangleq (\lambda+1)\max\left\{n, \left(\frac{\kappa+\lambda}{1+\lambda}\right)^2\right\}$ over $\lambda \geq 0$ to obtain the optimal complexity of SAGA. For any $a, b \geq 0$ we have $(a+b)/2 \leq \max\{a,b\} \leq a+b$, and so it follows that

$$\frac{(\lambda+1)\left(n+\left(\frac{\kappa+\lambda}{1+\lambda}\right)^2\right)}{2} \leq f(\lambda) \leq (\lambda+1)\left(n+\left(\frac{\kappa+\lambda}{1+\lambda}\right)^2\right).$$

The RHS and LHS above both have optimal values with order $O(\kappa\sqrt{n})$ when $\lambda = \lambda^* = \sqrt{\frac{(\kappa-1)^2-2}{n+1}}$, and so $f(\lambda^*) = O(\kappa\sqrt{n})$.

Asynchronous implementation We now verify Assumption 4.1. It can be shown that

$$\mathbb{E}G(\phi^k) \leq \left(1 - \frac{1}{n}\right) \mathbb{E}G(\phi^{k-1}) + \frac{L^2}{n}\mathbb{E}\|\hat{x}_k - x^*\|^2, \forall k \geq 1.$$

By adding and subtracting x_{k-1} in the second term of the above inequality, we have

$$\begin{aligned}
\mathbb{E}G(\phi^k) &\leq \left(1 - \frac{1}{n}\right) \mathbb{E}G(\phi^{k-1}) + \frac{L^2}{n}\mathbb{E}\|\hat{x}_{k-1} - x_{k-1} + x_{k-1} - x^*\|^2 \\
&\leq \left(1 - \frac{1}{n}\right) \mathbb{E}G(\phi^{k-1}) + \frac{2L^2\mathbb{E}\|\hat{x}_{k-1} - x_{k-1}\|^2}{n} + \frac{2L^2\mathbb{E}\|x_{k-1} - x_{k-1}\|^2}{n} \\
&\leq \left(1 - \frac{1}{n}\right) \mathbb{E}G(\phi^{k-1}) + \frac{2L^2}{n}\mathbb{E}\|x_{k-1} - x_{k-1}\|^2 + \frac{2L^2\gamma^2\tau^2M^2}{n}.
\end{aligned}$$

The second inequality above follows from the triangle inequality, and the last inequality follows from Lemma 11.

E.3 Technical details for SVRG-rand

Verifying Assumption 2 We first verify that SVRG-rand satisfies Assumption 2 and has a linear convergence rate.

Theorem 14 *Suppose Assumption 1 holds and that $\underline{p} > 0$, and set γ and ρ as in Theorem 7. Then, the sequence $\{x_k\}_{k\geq 0}$ produced by SVRG-rand satisfies* (14).

Proof It suffices to verify Assumption 2 and then use Theorem 7. Let $\mathcal{S} = [n]$, then Assumption 2.2 holds automatically with $c_3 = 0$ and $m_i = 1$ for all $i \geq 1$. To verify Assumption 2.1, we observe that the proxy ϕ_i^k has a probability p_{k-1} of changing to $B_i(x_{k-1})$ and a probability $1-p_{k-1}$ of remaining as ϕ_i^{k-1} for all $i \in [n]$. Take conditional expectations to see that

$$\mathbb{E}\left[G(\phi^k)|\mathcal{F}_{k-1}\right] = \frac{p_{k-1}}{n}\sum_{i=1}^n \|B_i(x_{k-1}) - B_i(x^*)\|^2 + \frac{1-p_{k-1}}{n}\sum_{i=1}^n \|\phi_i^{k-1} - B_i(x^*)\|^2.$$

Then use Lipschitz continuity of each B_i and the definition of $\underline{p}$ to conclude that

$$\begin{aligned}
\mathbb{E}G(\phi^k) &\leq (1-p_{k-1})\mathbb{E}G(\phi^{k-1}) + p_{k-1}L^2\mathbb{E}\|x_{k-1} - x^*\|^2 \\
&\leq (1-\underline{p})\mathbb{E}G(\phi^{k-1}) + \overline{p}L^2\mathbb{E}\|x_{k-1} - x^*\|^2,
\end{aligned}$$

for all $k \geq 0$. Therefore, SVRG-rand satisfies Assumption 2 with $\mathcal{S} = [n]$, $m_i = 1$ for all $i \geq 1$, $c_3 = 0$, $c_1 = 1 - \underline{p}$, and $c_2 = \overline{p}L^2$. ■

Basic and Catalyst complexity When $\overline{p} = \underline{p} = 1/n$, Theorem 15 shows that (24) holds and so (39) also holds for SVRG-rand. We then use the same arguments as for SAGA to derive the complexity of SVRG-rand before and after Catalyst acceleration.

E.4 Technical details for HSAG

Verifying Assumption 2 We first verify Assumption 2.1. Note that $G(\phi^{k-1})$ changes only when I_{k-1} takes a value in $\mathcal{S}$. Taking conditional expectation shows that for all $k \geq 1$,

$$\begin{aligned}\mathbb{E}\left[G(\phi^k)|\mathcal{F}_{k-1}\right] &= \frac{1}{n}\sum_{i\notin\mathcal{S}} G(\phi^{k-1}) \\ &\quad + \frac{1}{n}\sum_{i\in\mathcal{S}}\left(G\left(\phi^{k-1}\right) + \frac{1}{n}\left(\|B_i(x_{k-1}) - B_i(x^*)\|^2 - \|\phi_i^{k-1} - B_i(x^*)\|^2\right)\right) \\ &= G\left(\phi^{k-1}\right) + \frac{1}{n^2}\sum_{i\in\mathcal{S}}\left(\|B_i(x_{k-1}) - B_i(x^*)\|^2 - \|\phi_i^{k-1} - B_i(x^*)\|^2\right) \\ &= \left(1-\frac{1}{n}\right)G\left(\phi^{k-1}\right) + \frac{1}{n^2}\sum_{i\in\mathcal{S}}\|B_i(x_{k-1}) - B_i(x^*)\|^2.\end{aligned}$$

Then, use Lipschitz continuity of each B_i to conclude that for all $k \geq 1$,

$$\mathbb{E}G(\phi^k) \leq \left(1-\frac{1}{n}\right)\mathbb{E}G(\phi^{k-1}) + \frac{SL^2}{n^2}\mathbb{E}\|x_{k-1} - x^*\|^2. \tag{40}$$

Next we check Assumption 2.2. For $i \notin \mathcal{S}$ and $mk \leq t < m(k+1)$, we have $\phi_i^t = B_i(x_{mk})$ and

$$H(\phi^t) = \frac{1}{n}\sum_{i\notin\mathcal{S}}\|B_i(x_{mk}) - B_i(x^*)\|^2 \leq \frac{(n-S)\,L^2}{n}\|x_{mk} - x^*\|^2 \leq \frac{(n-S)\,L^2}{n}L_\rho(x_{mk}, \phi^{mk}).$$

The first inequality is due to Lipschitz continuity of each B_i, and the second inequality is due to the definition of L_ρ. HSAG then satisfies Assumption 2 with $c_1 = 1 - 1/n$, $c_2 = SL^2/n$, $c_3 = (n-S)L^2/n$, and $m_i = m$ for all $i \geq 1$.

Basic complexity Define

$$q \triangleq \max\left\{1 - 2\gamma\mu + 3\gamma^2L^2 + \rho\frac{SL^2}{n^2}, \left(\frac{2\gamma^2}{\rho} + 1 - \frac{1}{n}\right)\mathbb{I}(S>0)\right\}, \quad p \triangleq 2\gamma^2\frac{(n-S)L^2}{n}. \tag{41}$$

When $S = 0$, HSAG becomes SVRG and the above constants q and p are the same as for SVRG. Then we have

$$\mathbb{E}L_\rho(\tilde{x}_{k+1}, \tilde{\phi}_{k+1}) \leq \left(q^m + p(1 + q + q^2 + \cdots + q^{m-1})\right)\mathbb{E}L_\rho(\tilde{x}_k, \tilde{\phi}_k). \tag{42}$$

Now we choose the step size $\gamma = \lambda\,\mu/L^2$ where

$$\lambda = \min\left\{\frac{\kappa}{\sqrt{6n}}\mathbb{I}(0<S<n) + \mathbb{I}(S=0) + \mathbb{I}(S=n), \frac{1}{3+4S/n}\right\}. \tag{43}$$

When $S = 0$, the step size is the same as our choice for SVRG; when $S = n$, it is the same as our choice for SAGA. With the above selections of λ, $\gamma = \lambda\,\mu/L^2$, and $\rho = 4\gamma^2 n$, we show $q < 1$ so that the summation in (42) is bounded. First, we see that

$$q = \max\left\{1 - \frac{\lambda\,(2 - (3+4S/n)\,\lambda)}{\kappa^2}, \left(1 - \frac{1}{2n}\right)\mathbb{I}(S>0)\right\}. \tag{44}$$

The second term inside the brackets is always less than one. By (43), $\lambda \leq \frac{1}{3+4S/n}$ and so the first term inside the brackets above is also less than one. Combine both cases to conclude that $q < 1$. Then, (42) implies that

$$\mathbb{E}L_\rho(\tilde{x}_{k+1}, \tilde{\phi}_{k+1}) \leq \left(q^m + \frac{p}{1-q}\right) \mathbb{E}L_\rho(\tilde{x}_k, \tilde{\phi}_k). \tag{45}$$

By the definition of q above and p in (41), we compute

$$\frac{p}{1-q} = \max\left\{\frac{2\lambda\left(\frac{n-S}{n}\right)}{2-(3+4\frac{S}{n})\lambda}, \left(\frac{4\lambda^2(n-S)}{\kappa^2}\right)\mathbb{I}(S>0)\right\}. \tag{46}$$

With λ given by (43), we claim that $\frac{p}{1-q} \leq \frac{2}{3}$. When $S = n$, we have $p/(1-q) = 0$; when $S = 0$, we have $\frac{p}{1-q} = \frac{2\lambda}{2-3\lambda}$. Since $\lambda \leq \frac{1}{3}$ by (43) and the mapping $\lambda \mapsto \frac{2\lambda}{2-3\lambda}$ is monotone, we have $\frac{p}{1-q} \leq \frac{2}{3}$.

Now we consider the case where $0 < S < n$ and show that both terms inside the brackets of (46) are less than $\frac{2}{3}$. In this case $\lambda = \min\left\{\frac{\kappa}{\sqrt{6n}}, \left(\frac{1}{3+4S/n}\right)\right\}$ from (43) and so $\lambda \leq \frac{1}{3+4S/n}$. Use the monotonicity of the mapping $\lambda \mapsto \frac{2\lambda\left(\frac{n-S}{n}\right)}{2-(3+4S/n)\lambda}$ to see that the first term inside the brackets of (46) is less than $\frac{2}{3}$. On the other hand, since $\lambda = \min\left\{\frac{\kappa}{\sqrt{6n}}, \left(\frac{1}{3+4S/n}\right)\right\}$, we also have $\lambda \leq \frac{\kappa}{\sqrt{6n}}$. Then, the second term inside the brackets of (46) can be bounded with

$$\frac{4\lambda^2(n-S)}{\kappa^2} \leq \frac{2(n-S)}{3n} < \frac{2}{3}.$$

Combining all of these cases, we have shown $\frac{p}{1-q} \leq \frac{2}{3}$ with the choice of λ in (43), $\gamma = \lambda\,\mu/L^2$, and $\rho = 4\gamma^2 n$. Recall q can be written as (44) with the choice of λ in (43), $\gamma = \lambda\mu/L^2$, and $\rho = 4\gamma^2 n$. By setting

$$m = \max\left\{\frac{\log\frac{1}{12}}{\log\left(1 - \frac{2\lambda-(3+4S/n)\lambda^2}{\kappa^2}\right)}, \frac{\log\frac{1}{12}}{\log\left(1-\frac{1}{2n}\right)}\mathbb{I}(S>0)\right\}, \tag{47}$$

we have $q^m \leq \frac{1}{12}$ and so (45) becomes

$$\mathbb{E}L_\rho(\tilde{x}_{k+1}, \tilde{\phi}_{k+1}) \leq \frac{3}{4}\mathbb{E}L_\rho(\tilde{x}_k, \tilde{\phi}_k),$$

since $\frac{p}{1-q} \leq \frac{2}{3}$. In other words, we have shown the expected value of the Lyapunov function contracts by $\frac{3}{4}$ after each epoch.

We require $O(\log(1/\epsilon))$ epochs to obtain an ϵ-solution. In each epoch, the full updates require $n - S$ operator evaluations due to the SVRG updates. In each inner iteration, we need two operator evaluations. Then, the total number of operator evaluations required to achieve an ϵ-solution is $O((m + n - S)\log(1/\epsilon))$. Next, we elaborate on the order of m. Using the fact $\log(1+x) \approx 1 + x$ for small x, we have

$$m = O\left(\max\left\{\frac{\kappa^2}{2\lambda-(3+4S/n)\lambda^2}, n\,\mathbb{I}(S>0)\right\}\right) \tag{48}$$

from (47). Recall λ is chosen according to (43). If $S = 0$ or $S = n$, then $\lambda = \frac{1}{3+4S/n}$ and $2\lambda - (3 + 4S/n)\lambda^2 = \frac{1}{3+4S/n} \in [\frac{1}{3}, \frac{1}{7}]$, so $m = O\left(\max\{\kappa^2, n\,\mathbb{I}(S > 0)\}\right)$. On the other hand, suppose $0 < S < n$ and the minimum in (43) is attained at $\frac{\kappa}{\sqrt{6n}}$. Then $\lambda = \frac{\kappa}{\sqrt{6n}}$ and $\frac{\kappa}{\sqrt{6n}} \leq \frac{1}{3+4\frac{S}{n}} \leq \frac{1}{3}$. Furthermore, $2\lambda - (3 + 4S/n)\lambda^2 = \lambda + \left(\lambda - (3 + 4S/n)\lambda^2\right) \geq \lambda$, where the inequality uses $\lambda \leq \frac{1}{3+4S/n}$. This implies $\frac{\kappa^2}{2\lambda-(3+4S/n)\lambda^2} \leq \frac{\kappa^2}{\lambda} = \kappa\sqrt{6n} \leq 2n$ due to the condition $\frac{\kappa}{\sqrt{6n}} \leq \frac{1}{3}$. Thus $m = O(n)$ from (48). When $0 < S < n$ and the minimum in (43) is attained at $\frac{1}{3+4S/n}$, we can argue in the same was as for $S = 0$ or $S = n$ to see that $m = O\left(\max\{\kappa^2, n\}\right)$. We then have $m = O\left(\max\{\kappa^2, n\mathbb{I}(S > 0)\}\right)$ and obtain a total complexity of

$$O\left(\max\{\kappa^2, n\mathbb{I}(S > 0)\} + n - S\right)\log(1/\epsilon).$$

Catalyst complexity It is readily seen that for all $S \geq 0$ we have

$$\max\{n, \kappa^2\} \leq \left(\max\{\kappa^2, n\mathbb{I}(S > 0)\} + n - S\right) \leq 2\max\{n, \kappa^2\}.$$

The LHS and RHS are both the same order of complexity as SAGA. We then use the same arguments as for SAGA to derive the complexity of HSAG with Catalyst.

Asynchronous implementation Similar to (40), it can be shown that

$$\mathbb{E}G(\phi^k) \leq \left(1 - \frac{1}{n}\right)\mathbb{E}G(\phi^{k-1}) + \frac{SL^2}{n^2}\mathbb{E}\|\hat{x}_{k-1} - x^*\|^2,\ \forall k \geq 1.$$

Furthermore,

$$\mathbb{E}G(\phi^k) \leq \left(1 - \frac{1}{n}\right)\mathbb{E}G(\phi^{k-1}) + \frac{2SL^2}{n^2}\mathbb{E}\|x_{k-1} - x^*\|^2 + \frac{2SL^2\gamma^2\tau^2M^2}{n^2},\ \forall k \geq 1.$$

Assumption 4.2 holds with $c_3 = (n - S)\,L^2/n$. In addition, we have $\sum_{i=1}^n \mathbb{E}\|\phi_i^k - \hat{\phi}_i^k\|^2/n \leq 4\tau M_{\phi,B}^2/n$.

E.5 Technical details for SAGA+SVRG-rand

Verifying Assumption 2

Theorem 15 *Suppose Assumption 1 holds and $\underline{p} > 0$. Choose γ and ρ as in Theorem 7, then SAGA+SVRG-rand has a linear convergence rate.*

Proof Define functions G_1 and G_2 via

$$G_1(\phi^k) \triangleq \frac{1}{n}\sum_{i\in\mathcal{S}_1}\|\phi_i^k - B_i(x^*)\|^2, \quad G_2(\phi^k) \triangleq \frac{1}{n}\sum_{i\in\mathcal{S}_2}\|\phi_i^k - B_i(x^*)\|^2,$$

and let $S_1 = |\mathcal{S}_1|$ and $S_2 = |\mathcal{S}_2|$. It suffices to verify Assumption 2. We have $\mathcal{S} = [n]$, and so $H(\phi^k) \equiv 0$ and $G(\phi^k) = G_1(\phi^k) + G_2(\phi^k)$. Assumption 2.2 holds automatically with $c_3 = 0$ and $m_i = 1$ for all $i \geq 1$. We then check Assumption 2.1. From (40), we have

$$\mathbb{E}G_1(\phi^k) \leq \left(1 - \frac{1}{n}\right)\mathbb{E}G_1(\phi^{k-1}) + \frac{S_1L^2}{n^2}\mathbb{E}\|x_{k-1} - x^*\|^2,\ \forall k \geq 1.$$

Similar to the proof of Theorem 14, it can be shown that

$$\mathbb{E}G_2(\phi^k) \leq \left(1-\underline{p}\right)\mathbb{E}G_2(\phi^{k-1}) + \frac{\overline{p}S_2L^2}{n}\mathbb{E}\|x_{k-1}-x^*\|^2,\ \forall k \geq 1.$$

Combine the above two inequalities to see that

$$\begin{aligned}
\mathbb{E}G(\phi^{k+1}) &= \mathbb{E}G_1(\phi^{k+1}) + \mathbb{E}G_2(\phi^{k+1}) \\
&\leq \left(1-\frac{1}{n}\right)\mathbb{E}G_1(\phi^k) + (1-\underline{p})\mathbb{E}G_2(\phi^k) + \left(\frac{S_1L^2}{n^2}+\frac{\overline{p}S_2L^2}{n}\right)\mathbb{E}\left\|x_k-x^*\right\|^2 \\
&\leq \max\left\{1-\frac{1}{n}, 1-\underline{p}\right\}\mathbb{E}\left(G_1(\phi^k)+G_2(\phi^k)\right) \\
&\quad + \left(\frac{S_1L^2}{n^2}+\frac{\overline{p}S_2L^2}{n}\right)\mathbb{E}\left\|x_k-x^*\right\|^2 \\
&= \max\left\{1-\frac{1}{n}, 1-\underline{p}\right\}\mathbb{E}G(\phi^k) + \left(\frac{S_1L^2}{n^2}+\frac{\overline{p}S_2L^2}{n}\right)\mathbb{E}\left\|x_k-x^*\right\|^2.
\end{aligned}$$

We then conclude that the hybrid algorithm satisfies Assumption 2 with $\mathcal{S}=[n]$, $m_i=1$ for all $i \geq 1$, $c_3=0$, $c_1=\max\left\{1-1/n, 1-\underline{p}\right\}$, and $c_2 = S_1L^2/n^2 + \overline{p}S_2L^2/n$. ■

Basic and Catalyst complexity When $\overline{p}=\underline{p}=1/n$, from Theorem 15 we have that (24) holds and so (39) also holds here. We then use the same arguments as for SAGA to derive the complexity of SAGA+SVRG-rand with Catalyst.

E.6 Technical details for SARAH

Similar to Nguyen et al. (2017), we obtain linear convergence of the sequence $\|v_k\|^2$ to zero in expectation with rate $1-1/\kappa^2$ within each epoch. However, Nguyen et al. (2017) relies on the co-coercivity of B_i and is restricted to convex function minimization, while we do not require co-coercivity. To simplify, for any non-negative integer $s \geq 0$, consider the sequences $\{y_t^s\}_{t=1}^m$ and $\{I_t^s\}_{t=1}^m$ where

$$y_t^s = x_{sm+t}, \quad I_t^s = I_{sm+t}, \quad 0 \leq t \leq m, \tag{49}$$

and $\{v_t^s\}_{t=0}^m$ is defined by

$$v_t^s = \begin{cases} v_{sm+t} & 0 \leq t \leq m-1, \\ B_{I_t^s}(y_t^s) - B_{I_t^s}(y_{t-1}^s) + v_{t-1}^s & t=m. \end{cases} \tag{50}$$

In the above definitions, s denotes the index of the current epoch. We first bound v_t^s in the following lemma.

Lemma 16 *Suppose Assumption 1 holds and let $\{v_t^s\}_{t=1}^m$ be defined by (50) for some $s \geq 0$. Then for all $1 \leq t \leq m$,*

$$\mathbb{E}\|v_t^s\|^2 \leq (1-2\gamma\mu+\gamma^2L^2)\mathbb{E}\|v_{t-1}^s\|^2.$$

Proof By definition of v_k, y_t^s, and I_t^s given in (49), we have

$$\begin{aligned}\|v_t^s\|^2 &= \|B_{I_t^s}(y_t^s) - B_{I_t^s}(y_{t-1}^s) + v_{t-1}^s\|^2 \\ &= \|v_{t-1}^s\|^2 + \|B_{I_t^s}(y_t^s) - B_{I_t^s}(y_{t-1}^s)\|^2 - 2(B_{I_t^s}(y_t^s) - B_{I_t^s}(y_{t-1}^s))^\top v_{k-1}^s \\ &= \|v_{t-1}^s\|^2 + \|B_{I_t^s}(y_t^s) - B_{I_t^s}(y_{t-1}^s)\|^2 - \frac{2}{\gamma}(B_{I_t^s}(y_t^s) - B_{I_t^s}(y_{k-1}^s))^\top (y_t^s - y_{t-1}^s),\end{aligned}$$

using $y_t^s = y_{t-1}^s - \gamma v_{t-1}^s$. Take conditional expectation over $\mathcal{F}_t$ (here, $\mathcal{F}_t$ contains information up to the tth iteration in the sth epoch) to see

$$\begin{aligned}\mathbb{E}\left[\|v_t^s\|^2 \middle| \mathcal{F}_t\right] &= \|v_{t-1}^s\|^2 + \frac{1}{n}\sum_{i=1}^n \|B_i(y_t^s) - B_i(y_{t-1}^s)\|^2 - \frac{2}{\gamma}(B(y_t^s) - B(y_{t-1}^s))^\top (y_t^s - y_{t-1}^s) \\ &\le \|v_{t-1}^s\|^2 + \left(L^2 - \frac{2\mu}{\gamma}\right)\|y_t^s - y_{t-1}^s\|^2 \\ &= (1 - 2\gamma\mu + \gamma^2 L^2)\|v_{t-1}^s\|^2,\end{aligned}$$

where the inequality is by Lipschitz continuity and strong monotonicity of B, and the last line is by $y_t^s = y_{t-1}^s - \gamma v_{t-1}^s$. Take full expectations to conclude. ■

Now we bound the expectation of $\|v_t^s - B(y_t^s)\|^2$ in the following lemma.

Lemma 17 *Suppose Assumption 1 holds and let $\{v_t^s\}_{t=1}^m$ be defined by (50) for some $s \ge 0$. Then, for all $1 \le t \le m$ we have*

$$\mathbb{E}\|v_t^s - B(y_t^s)\|^2 \le \gamma^2 L^2 \sum_{j=1}^t \mathbb{E}\|v_{j-1}^s\|^2.$$

Proof By telescoping with $B(y_{t-1}^s)$ and v_{t-1}^s, $\|v_t^s - B(y_t^s)\|^2$ can be written as

$$\begin{aligned}\|v_t^s - B(y_t^s)\|^2 &= \|\left[B(y_{t-1}^s) - v_{t-1}^s\right] - (v_t^s - v_{t-1}^s) + \left[B(y_t^s) - B(y_{t-1}^s)\right]\| \\ &= \|B(y_{t-1}) - v_{t-1}^s\|^2 + \|v_t^s - v_{t-1}^s\|^2 + \|B(y_t^s) - B(y_{t-1}^s)\|^2 \\ &\quad - 2(B(y_{t-1}^s) - v_{t-1}^s)^\top (v_t^s - v_{t-1}^s) + 2(B(y_{t-1}^s) - v_{t-1}^s)^\top (B(y_t^s) - B(y_{t-1}^s)) \\ &\quad - 2(v_t^s - v_{t-1}^s)^\top (B(y_t^s) - B(y_{t-1}^s)).\end{aligned}$$

We can compute the conditional expectation of both sides and use the fact that $\mathbb{E}[v_t^s - v_{t-1}^s|\mathcal{F}_t] = B(y_t^s) - B(y_{t-1}^s)$ to see

$$\begin{aligned}\mathbb{E}\left[\|v_t^s - B(y_t^s)\|^2 \middle| \mathcal{F}_t\right] =& \|B(y_{t-1}^s) - v_{t-1}^s\|^2 + \|v_t^s - v_{t-1}^s\|^2 + \|B(y_t^s) - B(y_{t-1}^s)\|^2 \\ &- 2(B(y_{t-1}^s) - v_{t-1}^s)^\top (B(y_t^s) - B(y_{t-1}^s)) \\ &+ 2(B(y_{t-1}^s) - v_{t-1}^s)^\top (B(y_t^s) - B(y_{t-1}^s)) \\ &- 2\|B(y_t^s) - B(y_{t-1}^s)\|^2 \\ =& \|B(y_{t-1}^s) - v_{t-1}^s\|^2 + \|v_t^s - v_{t-1}^s\|^2 - \|B(y_t^s) - B(y_{t-1}^s)\|^2.\end{aligned}$$

By taking full expectations we have

$$\mathbb{E}\|v_t^s - B(y_t^s)\|^2 - \mathbb{E}\|v_{t-1}^s - B(y_{t-1}^s)\|^2 = \mathbb{E}\|v_t^s - v_{t-1}^s\|^2 - \|\mathbb{E}B(y_t^s) - B(y_{t-1}^s)\|^2, \quad (51)$$

then summing over $j = 0, 1, \ldots, s$, we have

$$\begin{aligned}\mathbb{E}\|v_t^s - B(y_t^s)\|^2 &= \mathbb{E}\|v_0^s - B(y_0^s)\|^2 + \sum_{j=1}^{t} \left(\mathbb{E}\|v_j^s - B(y_j^s)\|^2 - \mathbb{E}\|v_{j-1}^s - B(y_{j-1}^s)\|^2\right) \\ &= \sum_{j=1}^{t} \left(\mathbb{E}\|v_j^s - v_{j-1}^s\|^2 - \mathbb{E}\|B(y_j^s) - B(y_{j-1}^s)\|^2\right),\end{aligned}$$

where we used (51) and the fact that $v_0^s = B(y_0^s)$. By definition, $v_j^s - v_{j-1}^s = B_{I_j^s}(y_j^s) - B_{I_j^s}(y_{j-1}^s)$ and so the above display implies

$$\mathbb{E}\|v_t^s - B(y_t^s)\|^2 \le \sum_{j=1}^{t} \mathbb{E}\|B_{I_j^s}(y_j^s) - B_{I_j^s}(y_{j-1}^s)\|^2 \le L^2\mathbb{E}\|y_j^s - y_{j-1}^s\|^2 = \gamma^2 L^2 \mathbb{E}\|v_{j-1}^s\|^2.$$

The first inequality uses the non-negativity of $\mathbb{E}\|B(y_j^s) - B(y_{j-1}^s)\|^2$, the second inequality is by Lipschitz continuity of B_i, and the last equality is by definition of $y_j^s = y_{j-1}^s - \gamma v_{j-1}^s$. ■

For the next result, define the expression $q(\gamma) \triangleq 1 - 2\gamma\mu + \gamma^2 L^2$. By the above lemma and recursive application of Lemma 16, we arrive at the following result.

Lemma 18 *Suppose Assumption 1 holds, choose $\gamma = \mu/(2L^2)$ and $m = \log_{1-\frac{3}{4\kappa^2}}(\frac{1}{24})$, and let $\{y_t^s\}_{t=0}^m$ be defined by (49) for some $s \ge 0$. Then, we have*

$$\mathbb{E}\|B(y_m^s)\|^2 \le \frac{3}{4}\mathbb{E}\|B(y_0^s)\|^2.$$

Proof When $\gamma = \mu/(2L^2)$, we have $q = 1 - 3/(4\kappa^2) < 1$. Then, Lemma 17 shows that

$$\mathbb{E}\|v_t^s - B(y_t^s)\|^2 \le \left(\gamma^2 L^2 \sum_{j=1}^{t} q^{j-1}\right) \mathbb{E}\|v_0^s\|^2 \le \frac{\gamma^2 L^2}{1-q}\mathbb{E}\|B(y_0^s)\|^2, \forall s \ge 1.$$

The first inequality is by recursively applying Lemma 16, and the second inequality uses $v_0^s = B(y_0^s)$ and $q < 1$. In view of the above display, the Cauchy-Schwartz inequality gives

$$\begin{aligned}\mathbb{E}\|B(y_m^s)\|^2 &\le 2\mathbb{E}\|v_m^s\|^2 + 2\mathbb{E}\|v_m^s - B(y_m^s)\|^2 \\ &\le 2q^m\mathbb{E}\|v_0^s\|^2 + \frac{2\gamma^2 L^2}{1-q}\mathbb{E}\|B(y_0^s)\|^2 \\ &= (2q^m + \frac{2\gamma^2 L^2}{1-q})\mathbb{E}\|B(y_0^s)\|^2,\end{aligned}$$

where the first term on the RHS of the second line is by Lemma 16. Recall $q = q(\lambda) = 1 - 2\gamma\mu + \gamma^2 L^2$, so when $\gamma = \mu/(2L^2)$ we have

$$2q^m + \frac{2\gamma^2 L^2}{1-q} = 2\left(1 - \frac{3}{4\kappa^2}\right)^m + \frac{2}{3},$$

by routine calculations. If $m = \log_{1-\frac{3}{4\kappa^2}}(\frac{1}{24})$, the RHS is $3/4$ and we have

$$\mathbb{E}\|B(y_m^s)\|^2 \leq \frac{3}{4}\mathbb{E}\|B(y_0^s)\|^2.$$

■

Recall that we used $\tilde{x}_k$ to denote x_{km}, so the above lemma is equivalent to

$$\mathbb{E}\|B(\tilde{x}_{k+1})\|^2 \leq \frac{3}{4}\mathbb{E}\|B(\tilde{x}_k)\|^2, \forall k \geq 0.$$

Thus we require $O(\log(1/\epsilon))$ epochs to obtain an $\tilde{x}_\epsilon^*$ such that $\mathbb{E}\|B(\tilde{x}_\epsilon^*)\|^2 \leq \epsilon$. The complexity within each epoch is $n + 2m = O(n + \kappa^2)$, and so the overall complexity is $O(n + \kappa^2)\log(1/\epsilon)$.

Appendix F. Online appendices

All the source codes can be found in our online appendices: https://github.com/xunzhang1229/Variance-Reduction-algorithms-monotone-inclusion.git