On Unbounded Delays in Asynchronous Parallel Fixed-Point Algorithms

Robert Hannah* and Wotao Yin†

Department of Mathematics, University of California, Los Angeles, CA 90095, USA

September 13, 2016

Abstract

The need for scalable numerical solutions has motivated the development of asynchronous parallel algorithms, where a set of nodes run in parallel with little or no synchronization, thus computing with delayed information. This paper studies the convergence of the asynchronous parallel algorithm ARock under *potentially unbounded delays*.

ARock is a general asynchronous algorithm that has many applications. It parallelizes fixed-point iterations by letting a set of nodes randomly choose solution coordinates and update them in an asynchronous parallel fashion. ARock takes some recent asynchronous coordinate descent algorithms as special cases and gives rise to new asynchronous operator-splitting algorithms. Existing analysis of ARock assumes the delays to be bounded, and uses this bound to set a step size that is important to both convergence and efficiency. Other work, though allowing unbounded delays, imposes strict conditions on the underlying fixed-point operator, resulting in limited applications.

In this paper, convergence is established under unbounded delays, which can be either stochastic or deterministic. The proposed step sizes are more practical and generally larger than those in the existing work. The step size adapts to the delay distribution or the current delay being experienced in the system. New Lyapunov functions, which are the key to analyzing asynchronous algorithms, are generated to obtain our results. A set of applicable optimization algorithms with large-scale applications are given, including machine learning and scientific computing algorithms.

1 Introduction

Today there is a great need for efficient algorithms to solve large-scale problems in machine learning, big data, network analysis, PDEs, cosmology, weather simulations, and other areas. The power of an individual core, after 30 years' exponential growth, stopped increasing in 2005. So serial algorithms written today may not run significantly faster in the future. Moving forward, CPUs will only become faster through the addition of more cores rather than more powerful cores (see [18, 19]). Therefore the only way to utilize more powerful processors is to exploit parallelism. The trade off is that parallel algorithms are difficult to analyze and implement. However, the rewards for success are great: breakthroughs in parallel computing have implications for many other scientific fields.

*Email: RobertHannah89@math.ucla.edu
†Email: WotaoYin@math.ucla.edu

1.1 Motivation and importance of asynchronous algorithms

The vast majority of parallel algorithms are *synchronous* algorithms. For instance the synchronous-parallel Gauss-Jacobi algorithm divides the problem space $\mathbb{R}^N$ into p coordinate blocks. At every iteration, these blocks are updated by a corresponding set of p processors, and each processor's update is communicated to every other processor. Synchronous algorithms are simpler to analyze and implement; however, they have major drawbacks, such as synchronization penalty. At each iteration, all processors must wait for the results of the slowest processor to be received.

1.1.1 Disadvantages of synchronous algorithms

Synchronous algorithms may become impractical at scale, or on a busy network. Network latency is a major problem and bottleneck for parallel algorithms. Over a 20-25 year period on a wide range of systems, latency has improved by a factor of $20 - 40$ whereas CPU speeds have improved by a factor of 1000 [16]. This means that synchronizing at every step can be extremely expensive, and the divergence between processing speeds and latency will make this problem worse over time.

Moreover, these modest improvements in latency refer to the hardware's maximum performance. Latency and bandwidth are much worse in large data centers, which are typically very congested: Spikes in traffic can cause latency to increase temporarily by a factor of 20 [16]. Congestion also causes packet loss: Some data may fail to reach all parties, and must be sent again. If any computing node in a synchronous-parallel system experiences congestion or packet-loss, the entire system must wait for that one node. In addition, dedicated access to computing nodes often cannot be guaranteed. Nodes may suddenly start being used by another user, temporarily go offline, etc. causing further unpredictable delays. The more processors that are used, the higher the likelihood that one will experience problems, and that all others will have to wait. What is needed is a more flexible framework for parallel optimization: One that is resilient to latency, unpredictable and congested networks, packet loss, and other practical issues.

1.1.2 Advantages of asynchronous algorithms

A node in an asynchronous algorithm, instead of waiting to receive results from all other nodes, simply computes its next update using the most recent information it has received. Using outdated information will still result in convergence if the asynchronous algorithm is properly designed.

Latency, congestion, and random delays will no longer cripple the system, because processors can make progress without waiting on the results of the slowest processor. Asynchronous algorithms are resilient to packet-loss, unexpected drains on computing power, the loss of a node, and many other common problems on large congested networks. The speed of asynchronous algorithms is more related to the aggregate computing power and bandwidth of the system, rather than the speed of the slowest processor.

In addition, the algorithm discussed in this paper dynamically balances load with random coordinate block assignment: Processors take on as much work as they are currently able to, and no workload tuning is required.

There is, however, a trade-off: Using outdated information means the error decreases less per iteration. However more iterations can occurs per second because of vastly reduced synchronization penalty. Promising empirical obtained in [13] suggest that this trade-off is a favorable one.

1.2 Fixed-point algorithms

In this paper we consider convergence of ARock [13] under possibly unbounded asynchronous delays. ARock is a very general asynchronous-parallel, fixed-point algorithm in which a shared solution vector x^k is updated by a collection of processors.

The fixed-point framework is used because it is general, and many problems and algorithms can be written in the fixed-point form. Take a **nonexpansive** operator $T : \mathbb{H} \to \mathbb{H}$ (i.e. an operator with Lipschitz constant $L \leq 1$). The aim is to find a **fixed-point** of this operator: That is, a point $x^* \in \mathbb{H}$ such that $Tx^* = x^*$. For example, smooth minimization of a convex function $f : \mathbb{H} \to \mathbb{R}$ with L-Lipschitz gradient ∇f is equivalent to finding a fixed point of the nonexpansive operator $T = I - \frac{2}{L}\nabla f$, where I is the identity. The set of fixed points of an operator T is denoted $\mathrm{Fix}(T)$

A common fixed-point algorithm is the Krasnosel'skiĭ-Mann (KM) algorithm. Examples of KM iteration include: gradient descent, the proximal point algorithm, Douglass Rachford, forward-backward, ADMM, and Tseng splitting. Each is simply KM with a different fixed-point operator T.

Definition 1. Krasnosel'skiĭ-Mann algorithm. Let $\epsilon > 0$, and η^k be a series of step lengths in $(\epsilon, 1-\epsilon)$. Let be T a nonexpansive operator with at least one fixed point, and

$$S = I - T. \tag{1.1}$$

The KM Algorithm is defined by the following KM iteration:

$$\begin{aligned} x^{k+1} &= x^k - \eta^k S(x^k) \\ &= \big(\eta^k T + \big(1-\eta^k\big)I\big)(x^k). \end{aligned} \tag{1.2}$$

ARock is modelled on KM iteration, and can be thought of as asynchronous block KM iteration.

1.3 The ARock algorithm

Take a space $\mathbb{H}$ on which to solve an optimization problem. $\mathbb{H}$ can be the real space $\mathbb{R}^N$ or a separable Hilbert space. Break this space into m orthogonal subspaces: $\mathbb{H} = \mathbb{H}_1 \times \ldots \times \mathbb{H}_m$ so that vectors $x \in \mathbb{H}$ can be written as $(x_1, x_2, \ldots, x_m)$ where each x_i is x's component in subspace $\mathbb{H}_i$. Take a nonexpansive operator $T : \mathbb{H} \to \mathbb{H}$. Let $S = I - T$ and $Sx = (S_1 x, \ldots, S_m x)$ where $S_j x$ denotes the j'th block of $S(x)$.

Convention: *Superscripts* will denote the iteration number of a sequence of points $x^0, x^1, x^2, \ldots$. *Subscripts* will denote different blocks of a vector or operator, e.g., $x = (x_1, x_2, \ldots, x_m)$ and $Sx = (S_1 x, \ldots, S_m x)$. For instance, x_l^k is the lth block of iterate x^k. $S_l x^k$ is the lth block of $S(x^k)$.

Definition 2. The ARock Algorithm. Let η^k be a series of step lengths. Let be T a nonexpansive operator with at least one fixed point x^*, and $S = I - T$. Take a starting point $x^0 \in \mathbb{H}$. Then the ARock algorithm [13] is defined via the iteration:

$$\text{for } i = 1, \ldots, m, \quad x_i^{k+1} \leftarrow \begin{cases} x_i^k - \eta^k S_i(\hat{x}^k), & i = i(k), \\ x_i^k, & i \neq i(k), \end{cases} \tag{1.3}$$

where the **delayed iterate** $\hat{x}^k$ represents a possibly outdated version of the iteration vector x^k and the **block index sequence** $i(k)$ specifies which block of x^k is being updated to produce the next iterate x^{k+1}.

The ARock algorithm resembles KM iteration. However we use a delayed iterate $\hat{x}^k$ because of asynchronicity.

We now precisely define the block sequence $i(k)$, and the delayed iterate $\hat{x}^k$, that we will use in this paper.

1.3.1 Block sequence

Assumption 1. IID block sequence. The sequence in which blocks of the solution vector are updated, $i(k)$, is a series of uniform IID random variables that takes values $1, 2, \ldots, m$ each with probability $1/m$.

A uniform distribution is *not* strictly necessary, but is simpler. This assumption will hold if we allow all nodes to randomly update any block chosen in a uniform IID fashion. However this may result in bad data locality[1]. An alternative is to assign the m computing nodes one of m blocks each, and assume that the times taken to compute updates follow IID Poisson processes. Future work may involve weakening this assumption, perhaps extending the result beyond Poisson distributions.

1.3.2 Delayed iterates

Let $\vec{j} = (j_1, \ldots, j_m) \in \mathbb{N}^m$ be a vector, and $x^0, x^1, x^2, \ldots$ a series of iterates. Let $k \in \mathbb{N}$ be the iteration number. We find it convenient to define:

$$x^{k-\vec{j}} = \left(x_1^{k-j_1}, x_2^{k-j_2}, \ldots, x_m^{k-j_m}\right). \tag{1.4}$$

We define a series of **delay vectors** $\vec{j}(0), \vec{j}(1), \vec{j}(2), \ldots$ in $\mathbb{N}^m$, corresponding to $x^0, x^1, x^2, \ldots$ respectively. The components of these delay vectors are as follows:

$$\vec{j}(k) = (j(k,1), j(k,2), \ldots, j(k,m)). \tag{1.5}$$

The **current delay** is defined as[2]:

$$j(k) = \max_i \{j(k,i)\}. \tag{1.6}$$

Using this, we define the delayed iterate.

Definition 3. Delayed iterate. The delayed iterate $\hat{x}^k$ is defined as[3]:

$$\hat{x}^k = x^{k-\vec{j}(k)}, \text{ or equivalently,} \tag{1.7}$$

$$\hat{x}^k = \left(\hat{x}_1^k, \hat{x}_2^k, \ldots, \hat{x}_m^k\right) = \left(x_1^{k-j(k,1)}, x_2^{k-j(k,2)}, \ldots, x_m^{k-j(k,m)}\right). \tag{1.8}$$

Recall that asynchronous algorithms do not wait to receive results from all other nodes, but simply perform their updates with the most recent information they have available. Therefore processors may not necessarily have the most up-to-date information on x^k, but instead have a delayed iterate $\hat{x}^k$. Every block of $\hat{x}^k$ is outdated by a different amount: $j(k,i)$ denotes how many iterates out of date block i is at step k. Block 1 may be up to date, so $j(k,1) = 0$. Block 5 may be 17 iterations behind, so $j(k,5) = 17$.

Clearly this is a very general model: There is a series of delay vectors $\vec{j}(0), \vec{j}(1), \vec{j}(2), \ldots$ that represents how old the information that a computing node has access to is. How these delay vectors are determined depends on the model of asynchronicity chosen. We consider two possibilities in this paper: stochastic and deterministic delays (see Sections 1.4.1 and 1.4.2 respectively).

[1] That is, implementing the algorithm in this way may require a lot of data movement. This is because every single time a node makes an update, it must be sent the data for the entire block that it is updating.

[2] **Note:** The lack of the vector symbol distinguishes the current delay from the delay vector.

[3] **Stronger asynchronicity:** It is possible to have more general asynchronicity, where different components of the *same block*, $x_l \in \mathbb{H}_l$, have different ages. This leads to similar results, and a similar proof, but the current setup was chosen for simplicity.

1.4 New results and contributions

The contributions of this paper are two-fold. First, we prove the convergence of ARock under unbounded delays that are either stochastic or deterministic. This is achieved by constructing and analyzing Lyapunov functions. The second contribution of this paper is to describe and demonstrate general techniques for constructing Lyapunov functions, which appears to be the key to analyzing the convergence of asynchronous algorithms, and many other types of algorithms.

We leave coding and numerical tests to our future work because they involve engineering issues that are beyond the scope of this work. For example, the current delay, which affects the step size, can be obtained by many different methods. Our ongoing work such as [10] will develop codes and numerical results.

In rest of this subsection, we present these convergence results, but not in their most general forms. A more complete description of these results in all their generality is given in Sections 2 and 3.

1.4.1 Stochastic unbounded delay

The first result is the convergence of ARock under stochastic, potentially unbounded delays. First we precisely define the assumptions on the delay:

Definition 4. Evenly old delays. We say that delays are "evenly old" if there exists some constant B such that, with probability 1, we have $|j(k,i) - j(k,l)| \leq B$ for all $k \in \mathbb{N}$ and $1 \leq l \leq m$.

Delays can be arbitrarily large, but the ages of the various block are similar if they are evenly old. Clearly, if we have **bounded delay** (that is, with probability 1 we have $j(k) \leq \tau$ for some τ), this implies the evenly old property with constant $B = \tau$.

Assumption 2. Stochastic unbounded delays. The sequence of delay vectors $\vec{j}(0), \vec{j}(1), \ldots$ are IID, and independent of the block sequence $i(0), i(1), \ldots$. In addition, they are evenly old.

Hence there exists a function $p : \mathbb{N}^m \to [0,1]$ such that, for all $k \in \mathbb{N}$, the probability that $\vec{j}(k)$ equals some vector $\vec{v}$ is given by

$$\mathbb{P}\left[\vec{j}(k) = \vec{v}\right] = p(\vec{v}). \tag{1.9}$$

Define

$$P_l = \mathbb{P}[j(k) \geq l]. \tag{1.10}$$

Theorem 1. Convergence under stochastic unbounded delays. *Assume that the block sequence $i(k)$ is a uniform IID block sequence (Assumption 1) and that the delays vector $\vec{j}(k)$ are an evenly old, IID sequence that is independent of the block sequence (Assumption 2). Let the step size be $\eta^k = cH$ for an arbitrary fixed*[4] *$c \in (0,1)$, and H given below. Then the iterates of ARock converge weakly to a solution with probability 1 if either of the following holds:*

1. $\sum_{l=1}^{\infty} (lP_l)^{1/2} < \infty$, *and* $H = \left(1 + \frac{1}{\sqrt{m}} \sum_{l=1}^{\infty} P_l^{1/2} \left(l^{1/2} + l^{-1/2}\right)\right)^{-1}$.
2. $\sum_{l=1}^{\infty} P_l^{1/2} l < \infty$, *and* $H = \left(1 + \frac{2}{\sqrt{m}} \sum_{l=1}^{\infty} P_l^{1/2}\right)^{-1}$.

[4] By "arbitrary fixed" we mean that the constant c can be any number in $(0,1)$, so long as that number does not change. However it is possible to relax this.

Convergence under unbounded delays in this setting has only been proven under very strong assumptions (See Section 1.5 for a discussion of existing results). Additionally, this result improves on the step size criterion of ARock and other similar algorithms if we are willing to assume stochastic delays (e.g. [13, 11, 12]). So for instance, there may be a scenario where the maximum delay τ is very high, but delays near that size rarely occur. Theorem 1 implies a much larger allowable time step than prior work, leading to faster convergence.

Even if the assumption of independent IID delays does not hold in practice, the preceding step size gives a useful heuristic to use given an empirical distribution of delays: For instance, the step size should be $\eta^k \sim 1/\sqrt{m}$ for large m.

1.4.2 Deterministic unbounded delay

The second result of this paper proves convergence of ARock and related algorithms under deterministic unbounded delays. In order to achieve convergence, it is necessary to use a step size η^k that is a decreasing function of the current delay $j(k)$ (whereas in Theorem 1, a constant step size was sufficient). Also convergence is only on a family of subsequences.

Assumption 3. Deterministic unbounded delays. The sequence of delay vectors $\vec{j}(0), \vec{j}(1), \vec{j}(2), \ldots$ is an arbitrary sequence in $\mathbb{N}^m$, independent of $i(k)$, with $\liminf j(k) < \infty$.

Definition 5. Convergence on subsequences of bounded delay. Let $x^0, x^1, x^2, \ldots$ be a sequence of iterates and $\vec{j}(0), \vec{j}(1), \vec{j}(2), \ldots$ a corresponding sequence of delay vectors, with $\liminf j(k) < \infty$. Let Q_T be the subsequence of $x^0, x^1, x^2, \ldots$ where those iterates x^k with current delay $j(k) > T$ are removed[5]. We say that x^k converges to x^* on subsequences of bounded delay if x^k converges to x^* on every subsequence Q_T for $T \geq \liminf j(k)$[6].

Theorem 2. Convergence under deterministic unbounded delays. *Assume that the block sequence $i(k)$ is a sequence of uniform IID random variables (Assumption 1) and that the sequence of delay vectors $\vec{j}(0), \vec{j}(1), \vec{j}(2), \ldots$ is an arbitrary sequence in $\mathbb{N}^m$, independent of $i(k)$, with $\liminf j(k) < \infty$ (Assumption 3). Pick arbitrary, fixed $c \in (0, 1)$ and $R > 1$. Let the step size be*

$$\eta^k = c\left(1 + \frac{R^{(j(k)-\frac{1}{2})}}{\sqrt{m}(R-1)}\right)^{-1}. \tag{1.11}$$

Then with probability 1, the iterates of ARock weakly converge to a solution x^* on subsequences of bounded delay Q_T (Definition 5), where x^* does not depend on the bound T as long as $T \geq \liminf j(k)$.

This step size rule assumes a worst case scenario. In practice it can used if it was necessary to be certain that the algorithm converges. Even if network conditions are very unfavorable, the algorithm with the step size (1.11) makes some progress at every step. This result could also be used in the bounded delay regime when the bound τ is not known in advance. In previous results, τ is needed in advance to calculate the correct step size. Theorem 2 provides a rule adaptive to the current delay. When the delays are bounded (but possibly unknown to us), Theorem 2 implies weak convergence of the full sequence with probability 1, not merely on subsequences of bounded delay.

[5] Q_T represent subsequences of bounded delay.

[6] $T \geq \liminf j(k)$ ensures that Q_T is an infinite subsequence.

1.5 Related work

Asynchronous algorithms were first proposed by Chazan and Miranker in [6] to solve linear systems. Since then, asynchronous algorithms have been applied to many fields including nonlinear systems, differential equations, consensus problems, and optimization.

Until relatively recently, authors assumed a deterministic sequence of block updates: $i(1), i(2), \ldots$ with very little restriction. However, this imposes stronger restrictions on the problem. The delays $\vec{j}(k)$ are usually also assumed to be deterministic, but this appears to be relatively less restrictive. In [5], the authors describe two basic classes of deterministic asynchronous scenarios that appeared in the literature.

Definition 6. Totally asynchronous iteration. Every block, x_i, is updated infinitely many times. Information from iteration k (i.e. the components of x^k) is only used a finite number of times.

Total asynchronicity is a very weak condition that leads to convergence results with limited applicability (though there do exist applications to linear problems and strictly convex network flow problems [5, 21]). For instance, asynchronous linear iteration $x \mapsto Ax + b$ will only converge in general if the largest eigenvalue of $|A|$ (the matrix obtained by taking an absolute value of every entry) is strictly less than 1 ([6, 5]).

Definition 7. Partially asynchronous iteration. There exists an integer B such that every component, x_i, is updated at least once every B steps; and the information used by the processors cannot be older than B steps (bounded delay).

Partially asynchronous algorithms have better convergence properties. For instance, from [20]:

Theorem 3. *For strongly convex f with ∇f Lipschitz, there is a step size γ_1 such that for any step size $0 < \gamma < \gamma_1$, asynchronous gradient descent with partial asynchronicity converges at least linearly to a minimum, with rate $\mathcal{O}\Big((1 - c\gamma)^k\Big)$ for some constant c.*

However, the formulas for c or γ_1 are complicated, and the authors did not include them. These constants are also very tiny, because one needs to assume the worst-case scenario. The maximum delay B needs to be known in advance to determine the step size.

Stochastic asynchronous algorithms began to appear recently, a popular example being "Hogwild!" [15]. These algorithms always assume a bounded delay ($j(k, i) \leq \tau$ for all k and i), and that the sequence of blocks $i(k)$ is chosen independently and identically with $\mathbb{P}[i(k) = j] = p_j$ for fixed nonzero probabilities p_j. In [12], the authors prove function-value convergence for asynchronous stochastic coordinate descent. Under the assumption that the time step exponentially decays in τ in a certain way, they prove $\mathcal{O}(1/k)$ convergence for f convex with ∇f Lipschitz, and linear convergence when f is also strongly convex. This was extended in [11] to composite objective functions. In [3], the authors prove similar results for an asynchronous stochastic linear solver, using similar ideas. However point convergence ($x^k \to x^*$) is not attained for the non-strongly convex case in these papers[7]. The work presented in this paper generalizes and strengthens results from these recent papers on stochastic asynchronous algorithms.

There are recent unbounded delay results in the stochastic unconstrained convex optimization setting [9, 17, 1]. It is hard to compare results from a different optimization setting. However we note the following: We obtain point convergence ($x^k \rightharpoonup x^*$) rather than function-value convergence ($f\big(x^k\big) \to f(x^*)$) for convex f that is not necessarily strongly convex. The deterministic unbounded delay criterion in Theorem 2 is

[7] In the non strongly-convex case, point convergence is strongly than function-value convergence. In the strongly-convex case, they are equivalent.

weaker than all other delay assumptions. The step size in these papers converges to 0 as $k \to \infty$, which is an inevitable part of the problem setting. This makes asynchronicity error less of a problem. Nonetheless, in this paper, we are able to prove convergence in our setting with a step size rule that is only a function of the delay distribution despite unbounded delays (Theorem 1). The step size rule is invariant in k, and does not converge to 0. Theorem 2 features a step size that adapts to current delay conditions, once again invariant in k, which is cited as a key advantage of [17].

1.5.1 Technical discussion of deterministic unbounded delays

Theorem 2 can be seen as a halfway point between convergence of partially asynchronous and totally asynchronous algorithms. The only other convergence results for deterministic unbounded delays that the authors are aware of is for totally asynchronous algorithms, where only a small class of algorithms or problems converge. However making a slightly stronger assumption about the delays than in the totally asynchronous regime allows us to derive a stronger convergence result with a wider range of applications.

1.6 Structure of the paper

The remainder of the paper is organized as follows. Sections 2 and 3 give the convergence proofs for the stochastic and deterministic cases, respectively. Section 4 describes a set of optimization applications of our work.

2 Proof of Convergence for Stochastic Unbounded Delays

This section proves Theorem 4 below, which is a more general version of Theorem 1 from the introduction. Theorem 4 involves a sequence of arbitrary parameters $\epsilon_1, \epsilon_2, \ldots$ that appear naturally in our analysis. The values of these parameters can be chosen situationally to obtain different result. In Section 2.5, we select (i) the values that give the weakest conditions on delays, and (ii) the values that give the largest allowable timestep to obtain the two parts of Theorem 1 from the introduction.

Definition 8. Summable sequence. Let $a = (a_1, a_2, \ldots)$ $(a_i \in \mathbb{R}, \forall i)$ be a sequence. a is said to be **summable** or "in ℓ^1" if its ℓ^1 norm is finite, that is,

$$\|a\|_{\ell^1} = \sum_{i=1}^{\infty} |a_i| < \infty.$$

Theorem 4. Convergence under stochastic delays. *Consider ARock under the following conditions:*

1. The block sequence $i(k)$ is a uniform IID block sequence (Assumption 1).

2. The sequence of delay vectors $\vec{j}(k)$ is an evenly old, IID sequence that is independent of the sequence $i(k)$ (Assumption 2).

3. Let $\epsilon_1, \epsilon_2, \ldots \in (0, \infty)$ be an arbitrary sequence of parameters such that $\sum_{i=1}^{\infty} \frac{1}{\epsilon_i} < \infty$ and $\sum_{l=1}^{\infty} \epsilon_l P_l l < \infty$ for $P_l = \mathbb{P}[j(k) \geq l]$ (Assumption 4).

4. The step size is chosen as $\eta^k = cH$ for an arbitrary fixed $c \in (0,1)$ and $H = \left(1 + \frac{1}{m}\sum_{l=1}^{\infty} \epsilon_l P_l + \left\|\frac{1}{\epsilon_i}\right\|_{\ell^1}\right)^{-1}$.

Then with probability 1, the sequence of ARock iterates converges weakly to a solution.

This theorem is proven in section 2.4.3 after we build up a series of results throughout this section. This section is written in a way that attempts to explain the logic and intuition behind the approach taken. A general strategy for constructing Lyapunov functions is presented in Section 2.6. In Section 2.7, we discuss how to modify the proof for the simpler case of bounded delay.

2.1 Proof outline

Both convergence proofs rely on the following convergence criterion for fixed-point algorithms (see [4]):

Proposition 9. Convergence of nonexpansive fixed-point iterations. *Let T be a nonexpansive operator with at least one fixed point. If we have the following:*
(1) Norm convergence:[8] $\left\|x^k - x^*\right\|$ converges for every $x^* \in \mathrm{Fix}(T)$, and
(2) Fixed-point-residual (FPR) strong convergence:[9] $\|Tx^k - x^k\| \to 0$,
then x^k weakly converges to some $x^* \in \mathrm{Fix}(T)$ [10].

Proposition 9 is the basis of our convergence proofs in this paper, as well the proof of convergence of KM iteration. Toward applying Proposition 9, we study the following:

1. **Building a Lyapunov function:**[11] It turns out to be more natural to look at the Lyapunov function:

$$\underbrace{\xi^k}_{\text{Total error}} = \underbrace{\left\|x^k - x^*\right\|^2}_{\text{Classical error}} + \underbrace{\frac{1}{m}\sum_{i=1}^{\infty} c_i \left\|x^{k+1-i} - x^{k-i}\right\|^2}_{\text{Asynchronicity error}} \tag{2.1}$$

   rather than the classical error $\left\|x^k - x^*\right\|^2$ alone. Here, we let $x^n = x^0$ form $n < 0$. We cannot ensure that $\mathbb{E}\left[\left\|x^{k+1} - x^*\right\|^2\right] < \left\|x^k - x^*\right\|^2$ due to asynchronicity, and generally some kind of monotonicity result is needed to prove convergence. However adding what we might call the **asynchronicity error**, we regain this monotonicity of expectation, which leads to a viable proof.

2. **Martingale convergence theory**: This allows us to prove *norm convergence* and *FPR strong convergence* using results on the above Lyapunov function, which will complete the proof. Martingale theory is what allowed the authors in [13] to prove that x^k converges to a solution for minimization of a convex function with Lipschitz gradient, and not just that the function value converged to the optimal value.

2.2 Preliminary results

Recall that stochastic unbounded delays are analyzed under Assumptions 1 and 2. Define, for $k = 0, 1, \ldots,$ the filtration

$$\mathcal{F}^k = \sigma\Big(x^0, x^1, \ldots, x^k, \vec{j}(0), \vec{j}(1), \ldots, \vec{j}(k)\Big), \tag{2.2}$$

[8] We call this property *norm convergence*. The *distance* of x^k to each fixed-point x^* is what is converging (in general to a nonzero value) and not x^k itself. This property does not appear to have been given a name in the literature, although it is an important property in convergence proofs.
[9] The *fixed-point residual* (FPR) at x is defined as $(T - I)(x)$
[10] Weak convergence is the same as regular convergence in $\mathbb{R}^N$, but differs in a general Hilbert space.
[11] Technically this is not a Lyapunov function, but it resembles one.

which represents the history of iterates and delays up to the present step k. Let x^* be any solution, and set $x^* = 0$ with no loss in generality, to make some notation more compact. This can be achieved by translating the origin of the coordinate system to x^*. Thence, $\|x^k\|$ is the distance from the solution[12]:

$$\mathbb{E}\Big[\|x^{k+1}\|^2|\mathcal{F}^k\Big] = \mathbb{E}\Big[\|x^k - \eta^k S_{i(k)}\hat{x}^k\|^2|\mathcal{F}^k\Big] \tag{2.3}$$

$$= \|x^k\|^2 + \mathbb{E}\Big[-2\eta^k\langle x^k, S_{i(k)}\hat{x}^k\rangle + (\eta^k)^2\|S_{i(k)}\hat{x}^k\|^2|\mathcal{F}^k\Big], \tag{2.4}$$

where the expectation is taken over only the block index $i(k)$ only. Since the step size η^k is chosen independently of $i(k)$ and Assumptions 1 and 2 hold, we obtain

$$\mathbb{E}\Big[\|x^{k+1}\|^2|\mathcal{F}^k\Big] = \|x^k\|^2 \underbrace{-2\frac{\eta^k}{m}\langle x^k, S\hat{x}^k\rangle}_{\text{cross term}} + \frac{(\eta^k)^2}{m}\|S\hat{x}^k\|^2. \tag{2.5}$$

2.2.1 A fundamental inequality

We start with a fundamental inequality, which is the starting point for analyzing convergence.

Proposition 10. Fundamental inequality. *Under Assumptions 1 and 2, for $j(k)$ defined in (1.6), and an arbitrary sequence $\epsilon_1, \epsilon_2, \ldots \in (0,\infty)$, the ARock iterates obey the following inequality:*

$$\begin{aligned}\mathbb{E}\Big[\|x^{k+1} - x^*\|^2|\mathcal{F}^k\Big] \le \|x^k - x^*\|^2 &+ \frac{1}{m}\sum_{i=1}^{j(k)}\epsilon_i\|x^{k+1-i} - x^{k-i}\|^2 \\ &- \frac{\eta^k}{m}\|S\hat{x}^k\|^2\left(1 - \eta^k\left(1 + \sum_{i=1}^{j(k)}\frac{1}{\epsilon_i}\right)\right).\end{aligned} \tag{2.6}$$

The ϵ_i sequence is tunable. In [13], they are set to a constant value. However we eventually set them so that $1/\epsilon_i$ is summable, which is fundamental to the convergence proof for unbounded delays.

Proof. Let us start with the cross term in (2.5). Since T is nonexpansive, $\frac{1}{2}S$ is firmly nonexpansive (FNE)[13]. Hence,

$$\begin{aligned}-\frac{2\eta^k}{m}\langle S\hat{x}^k, x^k\rangle &= -\frac{2\eta^k}{m}\left(\langle S\hat{x}^k, \hat{x}^k\rangle + \langle S\hat{x}^k, x^k - \hat{x}^k\rangle\right) \\ &= -\frac{2\eta^k}{m}\left(2\left\langle\frac{1}{2}S\hat{x}^k, \hat{x}^k\right\rangle + \langle S\hat{x}^k, x^k - \hat{x}^k\rangle\right) \\ \left(\frac{1}{2}S \text{ is FNE}\right) &\le -\frac{2\eta^k}{m}\left(2\left\|\frac{1}{2}S\hat{x}^k\right\|^2 + \langle S\hat{x}^k, x^k - \hat{x}^k\rangle\right)\end{aligned}$$

[12] We will use an abuse of notation in this paper. We equate $S_i(x) \in \mathbb{H}_i$ (the components of $S(x)$ in the ith block) and $(0,\ldots,0,S_i(x),0,\ldots,0) \in \mathbb{H}_1 \times \ldots \times \mathbb{H}_m$ (the projection of $S(x)$ to the i'th subspace). Hence we can write the ARock iteration more compactly as $x^{k+1} = x^k - \eta^k S_{i(k)}\hat{x}^k$.

[13] A firmly nonexpansive (FNE) operator $Q : \mathbb{H} \to \mathbb{H}$ is an operator that can be written as $Q = \frac{1}{2}I + \frac{1}{2}R$, where R is nonexpansive. Equivalently, FNE operators satisfy $\langle Qy - Qx, y - x\rangle \ge \|Qy - Qx\|^2,\ \forall x, y \in \mathbb{H}$.

$$
\begin{aligned}
&= -\frac{\eta^k}{m}\left\|S\hat{x}^k\right\|^2 - \frac{2\eta^k}{m}\left\langle S\hat{x}^k, x^k - \hat{x}^k\right\rangle \\
\text{(break into coordinate blocks)} &= \sum_{l=1}^{m}\left(-\frac{\eta^k}{m}\left\|S_l\hat{x}^k\right\|^2 - \frac{2\eta^k}{m}\left\langle S_l\hat{x}^k, x_l^k - x_l^{k-j(k,l)}\right\rangle\right).
\end{aligned}
$$

Take block l. We turn the inner product into a telescoping sum:

$$
\begin{aligned}
&-\frac{\eta^k}{m}\left\|S_l\hat{x}^k\right\|^2 - \frac{2\eta^k}{m}\left\langle S_l\hat{x}^k, x_l^k - x_l^{k-j(k,l)}\right\rangle \\
&= -\frac{\eta^k}{m}\left\|S_l\hat{x}^k\right\|^2 - \frac{2\eta^k}{m}\left(\sum_{i=1}^{j(k,l)}\left\langle S_l\hat{x}^k, x_l^{k+1-i} - x_l^{k-i}\right\rangle\right) \\
\text{(Cauchy-Schwarz)} &\leq -\frac{\eta^k}{m}\left\|S_l\hat{x}^k\right\|^2 + \frac{2\eta^k}{m}\left(\sum_{i=1}^{j(k,l)}\frac{1}{2}\left(\left\|S_l\hat{x}^k\right\|^2\frac{\eta^k}{\epsilon_i} + \frac{\epsilon_i}{\eta^k}\left\|x_l^{k+1-i} - x_l^{k-i}\right\|^2\right)\right) \\
&\leq -\frac{\eta^k}{m}\left\|S_l\hat{x}^k\right\|^2 + \frac{\eta^k}{m}\left(\sum_{i=1}^{j(k)}\left(\left\|S_l\hat{x}^k\right\|^2\frac{\eta^k}{\epsilon_i} + \frac{\epsilon_i}{\eta^k}\left\|x_l^{k+1-i} - x_l^{k-i}\right\|^2\right)\right) \\
&= \frac{\eta^k}{m}\left\|S_l\hat{x}^k\right\|^2\left(\eta^k\left(\sum_{i=1}^{j(k)}\frac{1}{\epsilon_i}\right) - 1\right) + \frac{1}{m}\sum_{i=1}^{j(k)}\epsilon_i\left\|x_l^{k+1-i} - x_l^{k-i}\right\|^2.
\end{aligned}
$$

Adding all the components back together, we have:

$$
-2\frac{\eta^k}{m}\left\langle x^k, S\hat{x}^k\right\rangle + \frac{\left(\eta^k\right)^2}{m}\left\|S\hat{x}^k\right\|^2 \leq \frac{\eta^k}{m}\left\|S\hat{x}^k\right\|^2\left(\eta^k\left(1 + \left(\sum_{i=1}^{j(k)}\frac{1}{\epsilon_i}\right)\right) - 1\right) + \frac{1}{m}\sum_{i=1}^{j(k)}\epsilon_i\left\|x^{k+1-i} - x^{k-i}\right\|^2.
$$

Hence the proposition follows by adding $\left\|x^k\right\|^2$ to each side, and using (2.5). □

2.3 Building a Lyapunov function

In this section we demonstrate how to construct a Lyapunov function from (2.6) to prove convergence.

When calculating $\mathbb{E}\left[\left\|x^{k+1} - x^*\right\|^2 \middle| \mathcal{F}^k\right]$, notice that we obtained some difference terms of the form $\left\|x^{k+1-i} - x^{k-i}\right\|^2$. These difference terms are not easy to compare with $\left\|x^k - x^*\right\|^2$, and hence we cannot immediately say anything about the growth of the error. Instead of just considering $\left\|x^k - x^*\right\|^2$, we consider a Lyapunov function ξ^k defined as follows:

Definition 11. The Lyapunov function. Let $x^0, x^1, x^2, \ldots$ be a sequence of points in $\mathbb{H}$, and let $c_1, c_2, c_3, \ldots$ be a sequence of parameters in $[0, \infty)$. Set $x^n = x^0$ for all $n < 0$. We define the Lyapunov function:

$$
\xi^k = \left\|x^k - x^*\right\|^2 + \frac{1}{m}\sum_{i=1}^{\infty}c_i\left\|x^{k+1-i} - x^{k-i}\right\|^2. \tag{2.7}
$$

This is simply a linear combination of all the terms found when calculating the expectation of the original error. It is similar to, but different from, that used in [13]. When we calculate $\mathbb{E}\big[\xi^{k+1}\big|\mathcal{F}^k\big]$, we hope to only have terms similar to the terms found in ξ^k: that is, only terms like $\left\|x^k - x^*\right\|^2$ and $\left\|x^{k+1-i} - x^{k-i}\right\|$, and not some third species of terms. If this is the case, then we may carefully chose the coefficients $c_1, c_2, \ldots$ so that we may compare ξ^k and ξ^{k+1} in a meaningful way. Information about how *some kind of error* grows is essential to convergence proofs.

2.3.1 Analysis of the Lyapunov function

We now analyze the conditional expectation of the Lyapunov function defined in (2.7).

Lemma 12. Branch point lemma. *Take arbitrary $\epsilon_1, \epsilon_2, \ldots \in (0, \infty)$. Under Assumptions 1 and 2, the ARock iterates and ξ^k defined in (2.7) satisfy the following inequality:*

$$\begin{aligned}\mathbb{E}\big[\xi^{k+1}\big|\mathcal{F}^k\big] \leq \left\|x^k\right\|^2 &+ \frac{1}{m}\left(\sum_{i=1}^{j(k)} \epsilon_i \left\|x^{k+1-i} - x^{k-i}\right\|^2 + \sum_{i=1}^{\infty} c_{i+1}\left\|x^{k+1-i} - x^{k-i}\right\|^2\right) \\ &- \frac{\eta^k}{m}\left\|Sx^{k-\vec{j}(k)}\right\|^2\left(1 - \eta^k\left(1 + \frac{c_1}{m} + \sum_{i=1}^{j(k)} \frac{1}{\epsilon_i}\right)\right).\end{aligned} \tag{2.8}$$

Proof. Calculate the expectation:

$$\mathbb{E}\big[\xi^{k+1}\big|\mathcal{F}^k\big] = \mathbb{E}\Big[\left\|x^{k+1}\right\|^2\big|\mathcal{F}^k\Big] + \frac{c_1}{m}\mathbb{E}\Big[\left\|x^{k+1} - x^k\right\|^2\big|\mathcal{F}^k\Big] + \frac{1}{m}\sum_{i=1}^{\infty} c_{i+1}\left\|x^{k+1-i} - x^{k-i}\right\|^2. \tag{2.9}$$

The second term yields (by the definition of ARock iteration (1.3), and taking expectation over $i(k)$)

$$\mathbb{E}\Big[\left\|x^{k+1} - x^k\right\|^2\big|\mathcal{F}^k\Big] = \frac{\left(\eta^k\right)^2}{m}\left\|S(x^{k-j(k)})\right\|^2. \tag{2.10}$$

Then,

$$\begin{aligned}\mathbb{E}\big[\xi^{k+1}\big|\mathcal{F}^k\big] &= \underbrace{\mathbb{E}\Big[\left\|x^{k+1}\right\|^2\big|\mathcal{F}^k\Big]}_{A} + \underbrace{\frac{c_1}{m}\mathbb{E}\Big[\left\|x^{k+1} - x^k\right\|^2\big|\mathcal{F}^k\Big]}_{B} + \underbrace{\frac{1}{m}\sum_{i=1}^{\infty} c_{i+1}\left\|x^{k+1-i} - x^{k-i}\right\|^2}_{C} \\ &\leq \underbrace{\left\|x^k\right\|^2 + \frac{\eta^k}{m}\left\|Sx^{k-\vec{j}(k)}\right\|^2\left(\eta^k\left(1 + \sum_{i=1}^{j(k)}\frac{1}{\epsilon_i}\right) - 1\right) + \frac{1}{m}\sum_{i=1}^{j}\epsilon_i\left\|x^{k+1-i} - x^{k-i}\right\|^2}_{A} \quad \text{(by (2.6))} \\ &\quad + \underbrace{\frac{c_1}{m}\left(\frac{\left(\eta^k\right)^2}{m}\left\|Sx^{k-\vec{j}(k)}\right\|^2\right)}_{B} \quad \text{(by (2.10))}\end{aligned}$$

$$
\begin{aligned}
&\underbrace{+\frac{1}{m}\sum_{i=1}^{\infty} c_{i+1}\left\|x^{k+1-i}-x^{k-i}\right\|^2}_{C} \\
&= \left\|x^k\right\|^2 + \frac{1}{m}\left(\sum_{i=1}^{j(k)} \epsilon_i \left\|x^{k+1-i}-x^{k-i}\right\|^2 + \sum_{i=1}^{\infty} c_{i+1}\left\|x^{k+1-i}-x^{k-i}\right\|^2\right) \\
&\quad - \frac{\eta^k}{m}\left\|Sx^{k-\vec{j}(k)}\right\|^2\left(1-\eta^k\left(1+\frac{c_1}{m}+\sum_{i=1}^{j(k)}\frac{1}{\epsilon_i}\right)\right).
\end{aligned}
$$

□

Define

$$
\mathcal{G}^k = \sigma\big(x^0, x^1, \ldots, x^k\big), \tag{2.11}
$$

which represents the history of iterates $x^0, x^1, x^2, \ldots$. In the proposition below, we derive the natural choice of parameters of the Lyapunov function that allow a meaningful comparison between $\mathbb{E}\big[\xi^{k+1}\big|\, G^k\big]$ and ξ^k. With this choice, we obtain

$$
\mathbb{E}\big[\xi^{k+1}\big|\mathcal{G}^k\big] \leq \xi^k - (\text{descent terms}),
$$

which strongly resembles norm convergence: one of the convergence conditions in Proposition 9.

We first make some assumptions on the parameters. The necessity of these assumptions will become clear in the proof of Lemma 13.

Assumption 4. Coefficient summability conditions. Let $\epsilon_1, \epsilon_2, \ldots \in (0, \infty)$ and let $c_i = \sum_{l=i}^{\infty} \epsilon_l P_l$. These sequences also satisfy the summability conditions:

$$
\sum_{i=1}^{\infty} \frac{1}{\epsilon_i} < \infty, \tag{2.12}
$$

$$
\sum_{i=1}^{\infty} c_i < \infty. \tag{2.13}
$$

Lemma 13. Descent lemma for stochastic delays. *Consider the Lyapunov function ξ^k defined in (2.7). Let Assumptions 1, 2, and 4 hold. Let $H = \left(1 + \frac{c_1}{m} + \left\|\frac{1}{\epsilon_i}\right\|_{\ell^1}\right)^{-1}$. Then, ARock yields the following inequality for step size η^k:*

$$
\mathbb{E}\big[\xi^{k+1}\big|\mathcal{G}^k\big] \leq \xi^k - \big(1-\eta^k/H\big)\frac{\eta^k}{m}\sum_{\vec{j}\in\mathbb{N}^m} p\Big(\vec{j}\Big)\left\|Sx^{k-\vec{j}}\right\|^2.
$$

Proof. From the previous lemma (12) and (2.12), we have:

$$\begin{aligned}\mathbb{E}\big[\xi^{k+1}\big|\mathcal{F}^k\big] \leq \|x^k\|^2 + \frac{1}{m}\left(\sum_{i=1}^{j(k)} \epsilon_i \|x^{k+1-i} - x^{k-i}\|^2 + \sum_{i=1}^{\infty} c_{i+1}\|x^{k+1-i} - x^{k-i}\|^2\right) \\ - \frac{\eta^k}{m}\left\|Sx^{k-\vec{j}(k)}\right\|^2 \left(1 - \eta^k \underbrace{\left(1 + \frac{c_1}{m} + \left\|\frac{1}{\epsilon_i}\right\|_{\ell^1}\right)}_{1/H}\right).\end{aligned} \tag{2.14}$$

Let $p_j = \mathbb{P}[j(k) = j]$. Now take expectations over delays (via taking expectation with respect to $\mathcal{G}^k$ instead of $\mathcal{F}^k$).

$$\begin{aligned}\mathbb{E}\big[\xi^{k+1}\big|\mathcal{G}^k\big] &\leq \|x^k\|^2 + \frac{1}{m}\left(\sum_{j=1}^{\infty} p_j \sum_{i=1}^{j} \epsilon_i \|x^{k+1-i} - x^{k-i}\|^2 + \sum_{i=1}^{\infty} c_{i+1}\|x^{k+1-i} - x^{k-i}\|^2\right) \\ &\quad - \big(1 - \eta^k/H\big)\frac{\eta^k}{m}\sum_{\vec{j}\in\mathbb{N}^m} p\Big(\vec{j}\Big)\left\|Sx^{k-\vec{j}}\right\|^2 \\ &= \|x^k\|^2 + \frac{1}{m}\left(\sum_{i=1}^{\infty}\left(\sum_{j=i}^{\infty} p_j\right)\epsilon_i \|x^{k+1-i} - x^{k-i}\|^2 + \sum_{i=1}^{\infty} c_{i+1}\|x^{k+1-i} - x^{k-i}\|^2\right) \\ &\quad - \big(1 - \eta^k/H\big)\frac{\eta^k}{m}\sum_{\vec{j}\in\mathbb{N}^m} p\Big(\vec{j}\Big)\left\|Sx^{k-\vec{j}}\right\|^2 \\ &= \|x^k\|^2 + \frac{1}{m}\left(\sum_{i=1}^{\infty}(\epsilon_i P_i + c_{i+1})\|x^{k+1-i} - x^{k-i}\|^2\right) - \big(1 - \eta^k/H\big)\frac{\eta^k}{m}\sum_{\vec{j}\in\mathbb{N}^m} p\Big(\vec{j}\Big)\left\|Sx^{k-\vec{j}}\right\|^2.\end{aligned}$$

Let $\eta^k \leq H$ to eliminate the last term. Ideally $\mathbb{E}\big[\xi^{k+1}\big|\mathcal{G}^k\big] \leq \xi^k$, which can be achieved with:

$$\|x^k\|^2 + \frac{1}{m}\left(\sum_{i=1}^{\infty}(\epsilon_i P_i + c_{i+1})\|x^{k+1-i} - x^{k-i}\|^2\right) \leq \|x^k\|^2 + \frac{1}{m}\sum_{i=1}^{\infty} c_i\|x^{k+1-i} - x^{k-i}\|^2.$$

The obvious choice of coefficients is then given by $c_{i+1} + P_i\epsilon_i = c_i$. However this doesn't uniquely determine the coefficients. We assume that $c_i \to 0$ as i goes to ∞ to ensure that any bounded sequence has a corresponding Lyapunov function that is finite. Hence:

$$c_i = \sum_{l=i}^{\infty} \epsilon_i P_i.$$

This recovers the coefficient formula from Assumption 4. With this choice of coefficients, we have our result. □

2.4 Convergence proof

Now that we have built a Lyapunov function and obtained Lemma 13, we can prove convergence.

Lemma 14. *Let Assumptions 1, 2, and 4 hold. Use step size $\eta^k = cH$ for some arbitrary fixed $c \in (0,1)$, and H given in Lemma 13. Then with probability 1, ξ^k converges, and in addition,*

$$\sum_{k=0}^{\infty} \sum_{\vec{j} \in \mathbb{N}^m} p\left(\vec{j}\right) \left\| S x^{k-\vec{j}} \right\|^2 < \infty. \tag{2.15}$$

The proof of this lemma relies on the following:

Theorem 5. Supermartingale convergence theorem [7]. *Let α^k, θ^k and γ^k be positive sequences adapted to $\mathcal{F}^k$, and let γ^k be summable with probably 1. If*

$$\mathbb{E}\left[\alpha^{k+1} | \mathcal{F}^k\right] + \theta^k \leq \alpha^k + \gamma^k,$$

then with probability 1, α^k converges to a $[0,\infty)$-valued random variable, and $\sum_{k=1}^{\infty} \theta^k < \infty$.

We now prove Lemma 14.

Proof. Apply Theorem 5 with $\alpha^k = \xi^k$, $\gamma^k = 0$, and $\theta^k = \left(1 - \eta^k/H\right) \frac{\eta^k}{m} \sum_{\vec{j} \in \mathbb{N}^m} p\left(\vec{j}\right) \left\| S x^{k-\vec{j}} \right\|^2$. We immediately obtain our result by noting that $\left(1 - \eta^k/H\right) \frac{\eta^k}{m}$ is a constant. □

2.4.1 Norm convergence

Now is the point where the "evenly old" assumption about the delays made in Assumptions 2 becomes important, and it is hard to see a way to weaken it. First a lemma on convolutions is necessary.

Lemma 15. Convolution lemma ([2], Proposition 1.3.2). *Define the convolution of sequences $a = (\ldots, a_{-2}, a_{-1}, a_0, a_1, a_2, \ldots)$ and $b = (\ldots, b_{-2}, b_{-1}, b_0, b_1, b_2, \ldots)$ as the sequence defined by the formula[14]:*

$$(a * b)(k) = \sum_{i=-\infty}^{\infty} a_i b_{k-i}. \tag{2.16}$$

Let a_i be in ℓ^1, and let b be bounded with $b_i \to 0$ as $i \to \infty$. Then the convolution $(a * b)(k) \to 0$ as $k \to \infty$.

Proposition 16. Norm convergence. *Let Assumptions 1, 2, and 4 hold. Then with probability 1, $\left\| x^k - x^* \right\|$ converges for all $x^* \in \mathrm{Fix}(T)$.*

Proof. We first prove that with probability 1, $\frac{1}{m} \sum_{i=1}^{\infty} c_i \left\| x^{k+1-i} - x^{k-i} \right\|^2 \to 0$.

1. P_l is summable. Since the sequence $\frac{1}{\epsilon_i}$ is summable, $\epsilon_i \to \infty$, and thus $\inf_{i \in \mathbb{N}} \epsilon_i > 0$. Hence

$$\sum_{l=1}^{\infty} P_l \leq \frac{1}{\inf_{i \in \mathbb{N}} \epsilon_i} \sum_{l=1}^{\infty} \epsilon_l P_l = \frac{1}{\inf_{i \in \mathbb{N}} \epsilon_i} c_1 < \infty$$

[14] The convolution is not always well-defined, because the sum may not be convergent for all k. However in this lemma, it is well-defined.

2. $k - j(k) \to \infty$**.** (That is, the components of iterate x^k are used only a finite number of times).

$$\mathbb{P}[k - j(k) \leq k_0] = P_{k-k_0}$$
$$\sum_{k=k_0}^{\infty} \mathbb{P}[k - j(k) \leq k_0] = \sum_{k=k_0}^{\infty} P_{k-k_0} < \infty$$

Therefore by the Borel-Cantelli lemma, $k - j(k) \leq k_0$ happens only a finite number of times with probability 1. Hence with probability 1, this is true for all $k_0 \in \mathbb{N}$, which implies that $k - j(k) \to \infty$.

3. $Sx^{k+\vec{t}} \to 0$ **for all delay feasible "patterns"** $\vec{t}$**.** We assume without loss in generality that none of the delay vectors attained $(\vec{j}(0), \vec{j}(1), \ldots)$ has probability 0 (since this occurs with probability 1). Let $\vec{t}(k) \triangleq j(k)(1, \ldots, 1) - \vec{j}(k)$. $j(k)$ is the age of the oldest block in $x^{k-\vec{j}(k)}$, whereas $\vec{t}(k) \in \{0, 1, \ldots, B\}^m$ represent the "pattern" of the rest of the delay. We call a vector $\vec{t} \in \{0, 1, \ldots, B\}^m$ **feasible** if it occurs with nonzero probability. Take (2.15), and group the sum into feasible patterns and we obtain:

$$\sum_{k=0}^{\infty} \left\| Sx^{k+\vec{t}} \right\|^2 < \infty,$$
$$\implies \left\| Sx^{k+\vec{t}} \right\| \to 0, \tag{2.17}$$

for each feasible $\vec{t}$.

4. Delayed fixed-point residual $\left\| Sx^{k-\vec{j}(k)} \right\| \to 0$. Observe that

$$\left\| Sx^{k-\vec{j}(k)} \right\| = \left\| Sx^{(k-j(k))+\vec{t}(k)} \right\|.$$

Let $A(k, \vec{t}) = \left\| Sx^{(k-j(k))+\vec{t}} \right\|$ (this is a family of sequences indexed by $\vec{t}$). By equation (2.17), and the fact that $k - j(k) \to \infty$, we have $A(k, \vec{t}) \to 0$ for any *fixed* t. Notice that $\left\| Sx^{k-\vec{j}(k)} \right\| = A(k, \vec{t}(k))$. At every step, $A(k, \vec{t}(k))$ selects one from a finite family of sequences, all of which converge to 0. Since there are only a finite number of these sequences, $A(k, \vec{t}(k)) \to 0$ and hence $\left\| Sx^{k-\vec{j}(k)} \right\| \to 0$[15].

5. Difference sum converges to 0**.**

$$\begin{aligned}
&\frac{1}{m} \sum_{i=1}^{\infty} c_i \left\| x^{k+1-i} - x^{k-i} \right\|^2 \\
&\leq \frac{c^2 H^2}{m} \sum_{i=1}^{\infty} c_i \left\| Sx^{(k-i)-\vec{j}(k-i)} \right\|^2 \\
&= \frac{c^2 H^2}{m} \left((0, \ldots, 0, c_1, c_2, \ldots) * \left(\ldots, \left\| Sx^{(i-1)-\vec{j}(i-1)} \right\|^2, \left\| Sx^{(i)-\vec{j}(i)} \right\|^2, \left\| Sx^{(i+1)-\vec{j}(i+1)} \right\|^2, \ldots \right) \right)(k)
\end{aligned}$$

This expression is the convolution of an ℓ^1 sequence (Assumption 4), and a bounded sequence that converges to 0 as $i \to \infty$ (by part 4 of this proof) respectively. Therefore by Lemma 15, $\frac{1}{m} \sum_{i=1}^{\infty} c_i \left\| x^{k+1-i} - x^{k-i} \right\|^2 \to 0$.

[15] If you select from an *infinite* number of sequences converging to 0, this may not be true. E.g. consider $B(k, i) = \delta_{k-i}$, where $\delta_0 = 1$ and $\delta_l = 0$ for all $l \neq 0$. For fixed i, $B(k, i) \to 0$. However $B(k, k) = 1$ for all k, and hence never converges to 0.

6. Norm convergence. Because ξ^k converges a.s. and $\frac{1}{m}\sum_{i=1}^{\infty} c_i \left\|x^{k+1-i} - x^{k-i}\right\|^2 \to 0$ a.s., we have that for any particular x^*, $\left\|x^k - x^*\right\|$ converges with probability 1. Because the space is *separable*, this implies that with probability 1, $\left\|x^k - x^*\right\|$ converges for **all** $x^* \in \mathrm{Fix}(T)$, which is subtly different (See [7], Proposition 2.3 (iii) for a proof of this fact.). □

2.4.2 Fixed-point-residual strong convergence

Proposition 17. FPR strong convergence. *Under the conditions of Proposition 16,* $\left\|Sx^k\right\| \to 0$ *with probability 1.*

Proof. From equation (2.15), we have that $\left\|Sx^{k+\vec{t}}\right\| \to 0$ for some feasible $\vec{t}$ (clearly there must be at least one feasible $\vec{t}$). Recall that m is the number of blocks, and B is the maximum difference in age between blocks. We have

$$\begin{aligned}
\left\|Sx^k\right\| &\leq \left\|Sx^{k+\vec{t}} - Sx^k\right\| + \left\|Sx^{k+\vec{t}}\right\| \\
&\leq 2\left\|x^{k+\vec{t}} - x^k\right\| + \left\|Sx^{k+\vec{t}}\right\| \\
\text{(triangle inequality)} &\leq 2\sum_{i=1}^{m}\left\|x_i^{k+t_i} - x_i^k\right\| + \left\|Sx^{k+\vec{t}}\right\| \\
&\leq 2\sum_{i=1}^{m}\sum_{l=1}^{t_i}\left\|x_i^{k+l} - x_i^{k-1+l}\right\| + \left\|Sx^{k+\vec{t}}\right\| \\
\text{(since } \vec{t} \in \{0,1,\ldots,B\}^m\text{)} &\leq 2m\sum_{l=1}^{B}\left\|x_i^{k+l} - x_i^{k-1+l}\right\| + \left\|Sx^{k+\vec{t}}\right\| \to 0,
\end{aligned}$$

since $\left\|x^{k+1} - x^k\right\| \to 0$ and $\left\|Sx^{k+\vec{t}}\right\| \to 0$ (from parts 5 and 3 of the proof of Proposition 16 respectively). □

2.4.3 Proof of Theorem 4

Proof. Norm convergence is proven in Proposition 16. The FPR strong convergence criterion is proven in Proposition 17. Having satisfied the conditions of Proposition 9, we conclude that the sequence of ARock iterates converges to a solution with probability 1. Hence we have proven Theorem 4. □

2.5 Parameter choice

Choosing different parameters $\epsilon_1, \epsilon_2, \ldots$ lead to different convergence results. We featured two possibilities in Theorem 1 (though there are obviously others). We need both $\frac{1}{\epsilon_i} \in \ell^1$ and $\sum_{l=1}^{\infty} c_l = \sum_{l=1}^{\infty} \epsilon_l P_l l < \infty$ for convergence under step size $\eta^k = cH = c\Big(1 + \frac{1}{m}\sum_{l=1}^{\infty}\epsilon_l P_l + \left\|\frac{1}{\epsilon_i}\right\|_{\ell^1}\Big)^{-1}$.

1. If we wish to have the weakest restriction on our distribution of delays, let $\epsilon_l = m^{-1/2}P_l^{-1/2}l^{-1/2}$. This leads to the convergence condition $\sum_{l=1}^{\infty} P_l^{1/2} l^{1/2} < \infty$ for step size $\eta^k = c\Big(1 + \frac{1}{\sqrt{m}}\sum_{l=1}^{\infty} P_l^{1/2}\big(l^{1/2} + l^{-1/2}\big)\Big)^{-1}$.

2. If we wish to have the largest allowable step size (at the expense of a strong condition on the delay distribution), let $\epsilon_l = m^{-1/2} P_l^{-1/2}$. This leads to the convergence condition $\sum_{l=1}^{\infty} P_l^{1/2} l^1 < \infty$ for step size $\eta^k = c\Big(1 + \frac{2}{\sqrt{m}} \sum_{l=1}^{\infty} P_l^{1/2}\Big)^{-1}$.

2.6 General strategy

The general strategy for building Lyapunov functions is as follows. This has wide applicability in optimization, and not just asynchronous algorithms.

General Strategy:

1. Let ξ^k initially be the classical error $\left\|x^{k+1}\right\|^2$. We will adaptively change ξ, until we have a useful Lyapunov function. Calculate the expectation of the classical error $\mathbb{E}\Big[\left\|x^{k+1}\right\|^2 \big| \mathcal{F}^k\Big]$ and take inequalities (See section 2.2 where we obtained Proposition 10, the fundamental inequality.).

2. If this produces residual terms (in our case $\left\|x^{k+1-i} - x^{k-i}\right\|^2$) that we cannot eliminate, add a general linear combination of these terms to ξ^k. In this case, we add $\frac{1}{m} \sum_{i=1}^{\infty} c_i \left\|x^{k+1-i} - x^{k-i}\right\|^2$ to obtain the Lyapunov function in 11.

3. Repeat steps 1 and 2 until we gain "closure". I.e. The *positive terms* in the inequality for $\mathbb{E}\big[\xi^{k+1} \big| \mathcal{F}^k\big]$ are the same as the terms found in ξ^k (In our case, we only needed to do this once.).

4. *Negative terms* are not problematic because they serve to decrease the expectation of ξ^{k+1}. They should not be eliminated because they can give useful information. In our case,
$$-\frac{\eta^k}{m} \left\|S x^{k-\vec{j}(k)}\right\|^2 \left(1 - \eta^k \left(1 + \frac{c_1}{m} + \sum_{i=1}^{j(k)} \frac{1}{\epsilon_i}\right)\right) \tag{2.18}$$
was a negative term (see Lemma 13). This term was critical in the proof of the norm convergence and FPR strong convergence criterion in Section 2.4. See Lemma 14, Propositions 16, and 17.).

5. Vary the coefficients of the Lyapunov function to enable a useful comparison between $\mathbb{E}\big[\xi^{k+1} \big| \mathcal{F}^k\big]$ and ξ^k (See Lemma 13 where the coefficient formula in Assumption 4 was derived).

Which inequalities to take and which residual terms to create is a matter of trial and error. Some choices lead to dead ends, whereas others lead to a viable proof.

2.7 Bounded delay

Our main focus is on unbounded delay, because convergence under unbounded delay is a new result. It is easy, though, to modify this section's proof for the case of bounded delay, which results in a much simpler proof. Let $\epsilon_1, \ldots, \epsilon_\tau, \in (0, \infty)$ be a series of parameters, let $c \in (0, 1)$, let the step size be $\eta^k = c\Big(1 + \sum_{l=1}^{\tau} \Big(\frac{1}{m} \epsilon_l P_l + \frac{1}{\epsilon_l}\Big)\Big)^{-1}$. Then we have convergence with probability 1. The proof uses the following metric instead of an infinite sum:
$$\xi^k = \left\|x^k - x^*\right\|^2 + \frac{1}{m} \sum_{i=1}^{\tau} c_i \left\|x^{k+1-i} - x^{k-i}\right\|^2, \quad \text{for } c_i = \sum_{l=i}^{\tau} \epsilon_l P_l.$$

3 Proof of Convergence for Unbounded Deterministic Delays

Proving convergence for deterministic delays leads to a slightly weaker convergence result. This is likely because deterministic unbounded delay is a very general condition. Below is our most general result:

Theorem 6. Convergence under deterministic delays. *Consider ARock under the following conditions:*

1. The block sequence $i(k)$ is a sequence of uniform IID random variables (Assumption 1).

2. The sequence of delay vectors $\vec{j}(0), \vec{j}(1), \vec{j}(2), \ldots$ is an arbitrary sequence in $\mathbb{N}^m$, independent of $i(k)$, with $\liminf j(k) < \infty$ (Assumption 3).

3. Let $\epsilon_1, \epsilon_2, \ldots \in (0, \infty)$ be an arbitrary sequence of parameters such that $\sum_{l=1}^{\infty} \epsilon_l < \infty$.

4. The step size is set to $\eta^k = cH_{j(k)}$ for some arbitrary fixed $c \in (0,1)$ and $H_j = \left(1 + \frac{1}{m}\|\epsilon_i\|_{\ell^1} + \sum_{i=1}^{j} \frac{1}{\epsilon_i}\right)^{-1}$.

Then with probability 1, the sequence of ARock iterates converges weakly to a solution on subsequences of bounded delay (Definition 5).

This theorem is proven in Section 3.2.3. Similar to Theorem 4, there is a sequence of parameters $\epsilon_1, \epsilon_2, \ldots$. However in the case of deterministic delays, there is no "best" way to chose ϵ_i's unless stronger assumptions are made on the delays. It is impossible to optimize the parameters to uniformly ensure the maximum allowable timestep, since optimizing for a current delay of $j = n$ can only come at the expense of decreasing the allowable step size for other values $m \neq n$. We set these parameters to a convenient, simple choice in Section 3.3 to obtain Theorem 2 presented in the introduction.

Remark 1. Bounded delay. We can obtain a bounded-delay version of Theorem 6 by truncating the metric to the first τ terms as in Section 2.7 and setting $\epsilon_{\tau+1}, \epsilon_{\tau+2}, \ldots = 0$. Using the step size $\eta^k = c\left(1 + \sum_{i=1}^{j}\left(\frac{1}{m}\epsilon_l + \frac{1}{\epsilon_i}\right)\right)^{-1}$ results in convergence with probability 1.

3.1 Building a Lyapunov function

We build a Lyapunov function in a similar way to before. Our starting point is the Branch Point Lemma 12. Recall that $\mathcal{F}^k = \sigma\left(x^0, x^1, \ldots, x^k, \vec{j}(0), \vec{j}(1), \ldots, \vec{j}(k)\right)$, and let the Lyapunov function ξ^k be defined as before in equation (2.7). First though, it is necessary to make an assumption on the coefficients of the Lyapunov function. The necessity of this assumption will become clear in the proof of Lemma 18.

Assumption 5. Coefficient formula. Let $\epsilon_1, \epsilon_2, \ldots \in (0, \infty)$ be an arbitrary sequence of parameters such that $\sum_{l=1}^{\infty} \epsilon_l < \infty$. The coefficients of the Lyapunov function in equation (2.7) are given by $c_i = \sum_{l=i}^{\infty} \epsilon_l$.

3.1.1 Analysis of the metric

Lemma 18. Descent lemma for deterministic delays. *Consider the Lyapunov function ξ^k defined in (2.7). Let Assumption 1, 3, and 5 hold. Define*

$$H_j = \left(1 + \frac{c_1}{m} + \sum_{i=1}^{j} \frac{1}{\epsilon_i}\right)^{-1}. \tag{3.1}$$

Then ARock yields the following inequality for step size η^k:

$$\mathbb{E}\left[\xi^{k+1} \middle| \mathcal{F}^k\right] \leq \xi^k - \frac{\eta^k}{m}\left\|Sx^{k-\vec{j}(k)}\right\|^2 \left(1 - \left(\eta^k / H_{j(k)}\right)\right). \tag{3.2}$$

Proof. Start from the Branch Point Lemma (12):

$$\begin{aligned}\mathbb{E}\left[\xi^{k+1}\middle|\mathcal{F}^k\right] &\leq \left\|x^k\right\|^2 + \frac{1}{m}\left(\sum_{i=1}^{j(k)}\epsilon_i\left\|x^{k+1-i}-x^{k-i}\right\|^2 + \sum_{i=1}^{\infty}c_{i+1}\left\|x^{k+1-i}-x^{k-i}\right\|^2\right)\\ &\quad - \frac{\eta^k}{m}\left\|Sx^{k-\vec{j}(k)}\right\|^2\left(1-\eta^k\left(1+\frac{c_1}{m}+\sum_{i=1}^{j(k)}\frac{1}{\epsilon_i}\right)\right)\\ &\leq \left\|x^k\right\|^2 + \frac{1}{m}\left(\sum_{i=1}^{\infty}(\epsilon_i+c_{i+1})\left\|x^{k+1-i}-x^{k-i}\right\|^2\right)\\ &\quad - \frac{\eta^k}{m}\left\|Sx^{k-\vec{j}(k)}\right\|^2\left(1-\left(\eta^k/H_{j(k)}\right)\right).\end{aligned}$$

First assume $\eta^k/H_{j(k)} \leq 1$, to eliminate the last term. Ideally we have $\mathbb{E}\left[\xi^{k+1}\middle|\mathcal{F}^k\right] \leq \xi^k$, which can be achieved with:

$$\left\|x^k\right\|^2 + \frac{1}{m}\left(\sum_{i=1}^{\infty}(c_{i+1}+\epsilon_i)\left\|x^{k+1-i}-x^{k-i}\right\|^2\right) \leq \left\|x^k\right\|^2 + \frac{1}{m}\sum_{i=1}^{\infty}c_i\left\|x^{k+1-i}-x^{k-i}\right\|^2.$$

Using a similar argument to the one used in the proof of Lemma 13, we obtain the coefficient formula:

$$c_i = \sum_{l=i}^{\infty}\epsilon_l.$$

With this choice of coefficients, Lemma 18 is proven. □

3.2 Convergence proof

Now that we have built the Lyapunov function, and obtained Lemma 18, it is possible to prove convergence.

Lemma 19. *Consider the Lyapunov function ξ^k defined in (2.7). Let Assumption 1, 3, and 5 hold. Define H_j via equation (3.1). Let the step size $\eta^k = cH_{j(k)}$ for an arbitrary fixed $c \in (0,1)$. Then with probability 1, ξ^k converges, and we have:*

$$\sum_{k=1}^{\infty}H_{j(k)}\left\|Sx^{k-\vec{j}(k)}\right\|^2 < \infty, \tag{3.3}$$

$$\sum_{k=1}^{\infty}\left\|x^{k+1}-x^k\right\|^2 < \infty. \tag{3.4}$$

Hence $H_{j(k)}\left\|Sx^{k-\vec{j}(k)}\right\|^2 \to 0$ and $\left\|x^{k+1}-x^k\right\| \to 0$.

Proof. Now $\left\|x^{k+1}-x^k\right\| \leq cH_{j(k)}\left\|Sx^{k-\vec{j}(k)}\right\|$ (see Definition 2), and $H_{j(k)} \leq 1$. Hence:

$$\sum_{k=1}^{\infty}\left\|x^{k+1}-x^k\right\|^2 \leq \sum_{k=1}^{\infty}c^2H_{j(k)}^2\left\|Sx^{k-\vec{j}(k)}\right\|^2$$

$$\leq \sum_{k=1}^{\infty} H_{j(k)} \left\| S x^{k-\vec{j}(k)} \right\|^2.$$

Clearly then, equation (3.3) will imply all parts of this lemma (since any summable sequence converges to 0).

Use the Supermartingale Convergence Theorem (Theorem 5) on Lemma 18 with $\alpha^k = \xi^k$, $\gamma^k = 0$, and $\theta^k = \frac{\eta^k}{m} \left\| S x^{k-\vec{j}(k)} \right\|^2 \left(1 - \left(\eta^k / H_{j(k)}\right)\right)$. This implies that ξ^k converges with probability 1, and we have:

$$\sum_{k=1}^{\infty} \frac{c H_{j(k)}}{m} \left\| S x^{k-\vec{j}(k)} \right\|^2 (1-c) < \infty,$$
$$\implies \sum_{k=1}^{\infty} H_{j(k)} \left\| S x^{k-\vec{j}(k)} \right\|^2 < \infty.$$

This proves the lemma. □

3.2.1 Norm convergence

Lemma 20. *Assume the conditions of Lemma 19. Then with probability* 1, $\left\| x^k - x^* \right\|$ *converges for all* $x^* \in \mathrm{Fix}(T)$.

Proof. **1) Difference sum converges to** 0**:**

$$\frac{1}{m} \sum_{i=1}^{\infty} c_i \left\| x^{k+1-i} - x^{k-i} \right\|^2$$
$$= \left((0, \ldots, 0, c_1, c_2, \ldots) * \left(\ldots, \left\| x^{(i-1)+1} - x^{i-1} \right\|^2, \left\| x^{i+1} - x^i \right\|^2, \left\| x^{(i+1)+1} - x^{(i+1)} \right\|^2, \ldots \right) \right)(k)$$

Hence the difference sum is the convolution of a bounded sequence that converges to 0 as $i \to \infty$ (by Assumption 5), and an ℓ^1 sequence (by Lemma 19), respectively. Notice the reversal of roles from Proposition 16. Therefore, by Lemma 15, the difference sum converges to 0 with probability 1.

2) Norm Convergence: Therefore for any particular $x^* \in \mathrm{Fix}(T)$, with probability 1, $\left\| x^k - x^* \right\|$ converges. As argued before in the proof of Proposition 16, because the space is *separable*, this implies that with probability 1, $\left\| x^k - x^* \right\|$ converges for **all** $x^* \in \mathrm{Fix}(T)$. □

3.2.2 Fixed-point-residual strong convergence on subsequences of bounded delay

Lemma 21. FPR strong convergence. *Let the conditions of Lemma 19 hold. Let* $T \geq \liminf j(k)$. *Let* $Q_T \subset \mathbb{N}$ *be the subsequence of indices,* k*, on which the current delay,* $j(k)$*, is less than or equal to* T *(see Definition 5). On this subsequence, we have* $\left\| S x^k \right\| \to 0$.

Proof. **1) Delayed fixed-point residual** $\left\| S x^{k-\vec{j}(k)} \right\| \to 0$ **on** Q_T**.** The starting point is (3.3) from Lemma 19:

$$\sum_{k=1}^{\infty} H_{j(k)} \left\| S x^{k-\vec{j}(k)} \right\|^2 < \infty,$$

Consider the subsequence $Q_T \subset \mathbb{N}$. On this subsequence, the above becomes:

$$\begin{aligned}\infty &> \sum_{k\in Q_T} H_{j(k)} \left\| Sx^{k-\vec{j}(k)} \right\|^2 \\ &\geq \sum_{k\in Q_T} H_T \left\| Sx^{k-\vec{j}(k)} \right\|^2 \qquad \text{(since } H_j \text{ is decreasing in } j\text{).}\end{aligned}$$

Hence $\infty > \sum_{k\in Q_T} \left\| Sx^{k-\vec{j}(k)} \right\|^2$. So $\left\| Sx^{k-\vec{j}(k)} \right\| \to 0$ on Q_T.

2) Fixed-point residual strong convergence.

$$\begin{aligned}\left\| Sx^k \right\| &\leq \left\| Sx^k - Sx^{k-\vec{j}(k)} \right\| + \left\| Sx^{k-\vec{j}(k)} \right\| \\ &\leq 2\left\| x^k - x^{k-\vec{j}(k)} \right\| + \left\| Sx^{k-\vec{j}(k)} \right\| \\ &\leq 2\sum_{l=1}^{m} \left\| x_l^k - x_l^{k-j(k,l)} \right\| + \left\| Sx^{k-\vec{j}(k)} \right\| \\ &\leq 2\sum_{l=1}^{m} \sum_{i=1}^{j(k,l)} \left\| x_l^{k+1-i} - x_l^{k-i} \right\| + \left\| Sx^{k-\vec{j}(k)} \right\| \\ &\leq 2m\Big(\left\| x^k - x^{k-1} \right\| + \ldots + \left\| x^{k-(T+1)} - x^{k-T} \right\| \Big) + \left\| Sx^{k-j(k)} \right\| \to 0.\end{aligned}$$

The last line converges to 0 because $\left\| x^k - x^{k-1} \right\| \to 0$ and $\left\| Sx^{k-j(k)} \right\| \to 0$. Hence $\left\| Sx^k \right\| \to 0$ on Q_T. □

3.2.3 Proof of Theorem 6

Proof. Norm convergence was proven in Lemma 20. FPR strong convergence on subsequences of bounded delay was proven in Lemma 21. Having satisfied the conditions of Proposition 9, we conclude that the sequence of ARock iterates converges to a solution with probability 1 on subsequence of bounded delay. □

3.3 Parameter choice

The parameters $\epsilon_1, \epsilon_2, \ldots$ are arbitrary. However, for the purposes of simplicity and demonstration, ϵ_i was set to $\sqrt{m}R^{l-\frac{1}{2}}$ to obtain Theorem 2 in the introduction, from the more general Theorem 6.

4 Applications

Our results apply to the fixed-point algorithm ARock, which encompasses a wide range of algorithms, including gradient descent, proximal point, Douglas-Rachford (and Peaceman-Rachford), forward-backward, ADMM, etc. We discuss some of these applications in this section so the interested reader will see how our results can be applied.

In order for ARock to be applicable, the optimization problem needs to be able to be formulated as a fixed-point problem of a nonexpansive operator T. For ARock to be practical however, it needs to be possible to efficiently parallelize the corresponding serial iteration. For example, consider gradient

descent on a convex function f with L-Lipschitz gradient. The corresponding ARock algorithm is given by $x_{i_k}^{k+1} = x_{i_k}^k - \frac{\eta^k}{L}\nabla_{i_k} f(\hat{x}^k)$, where $\nabla_{i_k} f$ is the i_k'th block of the full gradient ∇f and $x_i^{k+1} = x_i^k$ for all $i \neq i_k$. x^{k+1} and x^k only differ on the i_kth component. If it is not significantly easier to calculate $\nabla_{i_k} f$ than to calculate the full gradient, then there is no advantage to a parallel algorithm in this case. However ARock is practical for a wide variety of algorithms and applications; see the paper [14] for the structures of operators that give rise to parallelizable ARock algorithms.

4.1 Fixed-point algorithms

Below is a table of popular and general algorithms that can be viewed as fixed-point algorithms, and therefore can potentially be made asynchronous parallel using ARock and our results.

In the table, gradients such as $\nabla f, \nabla g, \nabla h$ are assumed to be Lipschitz continuous with constants $L_{\nabla f}, L_{\nabla g}, L_{\nabla h}$, respectively. The resolvent operator $J_{\gamma\partial f} = (I + \gamma\partial f)^{-1}$ is equivalent to the proximal operator of f, that is,

$$J_{\gamma\partial f} = (I + \gamma\partial f)(y) = \arg\min_x f(x) + \frac{1}{2\gamma}\|x - y\|^2, \quad \forall y.$$

If an algorithm does not use the gradient of a function, the function can represent a possibly nonsmooth function or a constraint (through the indicator function.) For example, the constraint $x \in C$ where C is nonempty closed convex set is equivalent the minimization objective function $f(x) = \iota_C(x)$, which equals 0 if $x \in C$ and ∞ otherwise; for this indicator function, ∂f is the normal cone of C, and $J_{\gamma\partial f}(y)$ equals the projection of x onto C. The reflective resolvent open $R_{\gamma\partial f}$ is defined as $2J_{\gamma\partial f} - I$. When C is a hyperplane and $f = \iota_C$, for any $\gamma > 0$, $R_{\gamma\partial f}(y)$ returns the reflection of y through C. Also, it is assumed that $\partial(f + g) = \partial f + \partial g$ and $\partial(F + G) = \partial F + \partial G$ in the table.

Optimization problem	**Algorithm**	**Nonexpansive fixed-point operator T**	**Assumptions**
$\min\ f(x)$	Gradient descent	$I - \gamma\nabla f$	$\gamma \in (0, \frac{2}{L_{\nabla f}}]$
$\min\ f(x)$	Proximal point	$J_{\gamma\partial f}$	$\gamma > 0$
$\min f(x) + g(x)$	Forward backward	$J_{\gamma\partial f} \circ (I - \gamma\nabla g)$	$\gamma \in (0, \frac{2}{L_{\nabla g}}]$
$\min\{g(x) : x \in C\}$	Projected gradient	$\mathrm{Proj}_C \circ (I - \gamma\nabla g)$	$\gamma \in (0, \frac{2}{L_{\nabla g}}]$
$\min f(x) + g(x)$	Peaceman-Rachford	$R_{\gamma\partial f} \circ R_{\gamma\partial g}$	$\gamma > 0$
$\min \sum_{i=1}^d f_i(x)$	Parallel Peaceman-Rachford	$(\frac{2}{d}\mathbf{1}\mathbf{1}^T - I) \circ R_{\gamma\partial\mathbf{f}}$ where $\mathbf{f} = [f_1; \ldots; f_d] : \mathbb{H}^d \to \mathbb{R}^d$	$\gamma > 0$
$\min f(x) + g(x)$	Douglas-Rachford	$\frac{1}{2}I + \frac{1}{2}R_{\gamma\partial f} \circ R_{\gamma\partial g}$	$\gamma > 0$
$\min f(x) + g(x) + h(x)$	Davis-Yin	$I - J_{\gamma\partial g} + J_{\gamma\partial f} \circ (2J_{\gamma\partial g} - I - \gamma\nabla h \circ J_{\gamma\partial g})$	$\gamma \in (0, \frac{2}{L_{\nabla h}}]$
$\min\{f(x) + g(z) : Ax + Bz = b\}$	ADMM	$\frac{1}{2}I + \frac{1}{2}R_{\gamma\partial F} \circ R_{\gamma\partial G}$, where $F(y) := f^*(A^T y)$, $G(y) := g^*(B^T y) - b^T y$	$\gamma > 0$

Explanation: Columns 1 and 2 contain the optimization problem and the applicable algorithm. Column 3 gives the nonexpansive fixed-point operator T corresponding to the algorithm. A fixed point of T is a solution to the optimization problem. When you apply the KM iteration to T, you obtain the algorithm in column 2. Column 4 contains assumptions for convergence. The derivations of the algorithms and operators,

as well as the proof of nonexpansiveness, are out of the scope of this paper. We refer the interested reader to [4, 8].

As an example derivation, consider gradient descent on a proper closed convex function f with $L_{\nabla f}$-Lipschitz gradient. By the Baillon-Haddad theorem, $\frac{1}{L_{\nabla f}}\nabla f$ is firmly nonexpansive, and therefore $T = I - \frac{2}{L_{\nabla f}}\nabla f$ is a nonexpansive operator. In addition we have:

$$\begin{aligned} x^* \in \mathrm{Fix}(T) &\iff \left(I - \frac{2}{L_{\nabla f}}\nabla f\right)(x^*) = x^* \\ &\iff \frac{2}{L_{\nabla f}}\nabla f(x^*) = 0 \\ &\iff x^* \text{ is a minimizer of } f. \end{aligned}$$

Therefore the corresponding fixed point problem is equivalent to the function minimization problem. Applying KM iteration to this T with step size η^k yields:

$$x^{k+1} \quad = \quad x^k - \frac{2\eta^k}{L_{\nabla f}}\nabla f(x^k),$$

which is the gradient descent algorithm. We make the above asynchronous parallel to obtain the corresponding ARock iteration:

$$\begin{aligned} x_{i_k}^{k+1} &= x_{i_k}^k - \frac{2\eta^k}{L_{\nabla f}}\nabla_{i_k} f\big(\hat{x}^k\big), \\ x_j^{k+1} &= x_j^k, \qquad\qquad j \neq i_k, \end{aligned}$$

to which the results of this paper apply.

4.2 Applications

ARock and results of this paper can be applied to a wide variety of computational problems that appear in machine learning, scientific computing, etc. We present a small sample of applications (more applications are found in [13, 14]). Below is a summary:

Convex Optimization Problem	**Setup**	**ARock Iteration**
Smooth minimization: min $f(x)$	∇f is L-Lipschitz, $\nabla f = \begin{pmatrix} \nabla f_1 \\ \vdots \\ \nabla f_m \end{pmatrix}$	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \frac{2\eta^k}{L} \nabla f_{i_k}(\hat{x}^k)$
Constrained minimization: min $f(x)$ subject to $\ell \leq x \leq u$	same as above	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \eta^k \left(\hat{x}_{i_k}^k - \mathrm{Proj}_{[\ell_{i_k}, u_{i_k}]} \big(\hat{x}_{i_k}^k - \frac{2}{L} \nabla f_{i_k}(\hat{x}^k) \big) \right)$
Composite minimization (ERM model): min $f(x) + g(x)$	same as above, plus $g(x) = \sum_{i=1}^m g_i(x_i)$	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \eta^k \left(\hat{x}_{i_k}^k - \mathrm{prox}_{\frac{2}{L} g_i} \big(\hat{x}_{i_k}^k - \frac{2}{L} \nabla f_{i_k}(\hat{x}^k) \big) \right)$
Kernel SVM: $\min_s \frac{1}{2} s^T Q s - e^T s$ subject to $\sum_i y_i s_i = 0$, $0 \leq s_i \leq C,\ \forall i$	training set $\{x_i, y_i\}$, $y_i \in \{\pm 1\}$, kernel $k(\cdot,\cdot)$, $Q_{ij} = y_i y_j k(x_i, x_j)$, applies Davis-Yin	See the last equation in [14, Section 5.2.1], and apply it with damping η^k
Linear System: Solve $Ax = b$	A is symmetric positive definite, $\begin{pmatrix} - A_1 - \\ \vdots \\ - A_m - \end{pmatrix} x = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}$	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \left(\frac{2\eta^k}{M} \right) \big(A_{i_k} \hat{x}^k + b_{i_k} \big)$
Linear System: Solve $Ax = b$	$A = D + R$ where D is diagonal, M off-diagonal, $\rho\big(-D^{-1}R\big) \leq 1$	$x_{i_k}^{k+1} \leftarrow x_{i_k}^k - \eta^k \big(\big(I + D^{-1} M \big) \hat{x}^k - D^{-1} b \big)_{i_k}$

In the table, $\mathrm{Proj}_{[\ell_{i_k}, u_{i_k}]}$ projects a scalar to the interval $[\ell_{i_k}, u_{i_k}]$, and the operator $\mathrm{prox}_{\gamma g_i}(y)$, $\gamma > 0$, returns the minimization solution to the proximal subproblem

$$\underset{x_i}{\text{minimize}}\, g_i(x_i) + \frac{1}{2\gamma} \|x_i - y\|_2^2.$$

Each iteration in Column 3 of the table is obtained by substituting the operator $S = I - T$ in the ARock iteration $x_{i_k}^{k+1} \leftarrow x_{i_i}^k - \eta^k S_{i_k}(\hat{x}^k)$ using the appropriate fixed-point operator T for each application.

References

[1] A. Agarwal and J. C. Duchi. "Distributed Delayed Stochastic Optimization". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., 2011, pp. 873–881.

[2] W. Arendt et al. *Vector-Valued Laplace Transforms and Cauchy Problems: Second Edition*. Springer Science & Business Media, Apr. 5, 2011. 540 pp.

[3] H. Avron, A. Druinsky, and A. Gupta. "Revisiting Asynchronous Linear Solvers: Provable Convergence Rate through Randomization". In: *Parallel and Distributed Processing Symposium, 2014 IEEE 28th International*. Parallel and Distributed Processing Symposium, 2014 IEEE 28th International. May 2014, pp. 198–207.

[4] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Science & Business Media, Apr. 19, 2011. 470 pp.

[5] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[6] D. Chazan and W. Miranker. "Chaotic Relaxation". In: *Linear Algebra and its Applications* 2.2 (Apr. 1, 1969), pp. 199–222.

[7] P. Combettes and J. Pesquet. "Stochastic Quasi-Fejér Block-Coordinate Fixed Point Iterations with Random Sweeping". In: *SIAM Journal on Optimization* 25.2 (Jan. 1, 2015), pp. 1221–1248.

[8] D. Davis and W. Yin. "A Three-Operator Splitting Scheme and Its Optimization Applications". In: (Apr. 4, 2015). arXiv: 1504.01032 [math].

[9] J. C. Duchi, S. Chaturapruek, and C. Ré. "Asynchronous Stochastic Convex Optimization". In: (Aug. 4, 2015). arXiv: 1508.00882 [math, stat].

[10] B. Edmunds, Z. Peng, and W. Yin. "TMAC: A Toolbox of Modern Async-Parallel, Coordinate, Splitting, and Stochastic Methods". In: (June 14, 2016). arXiv: 1606.04551 [math].

[11] J. Liu and S. Wright. "Asynchronous Stochastic Coordinate Descent: Parallelism and Convergence Properties". In: *SIAM Journal on Optimization* 25.1 (Jan. 1, 2015), pp. 351–376.

[12] J. Liu et al. "An Asynchronous Parallel Stochastic Coordinate Descent Algorithm". In: *J. Mach. Learn. Res.* 16.1 (Jan. 2015), pp. 285–322.

[13] Z. Peng et al. "ARock: An Algorithmic Framework for Asynchronous Parallel Coordinate Updates". In: (June 8, 2015). arXiv: 1506.02396 [cs, math, stat].

[14] Z. Peng et al. "Coordinate Friendly Structures, Algorithms and Applications". In: *Annals of Mathematical Sciences and Applications* 1.1 (2016), pp. 57–119. arXiv: 1601.00863.

[15] B. Recht et al. "Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent". In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor et al. Curran Associates, Inc., 2011, pp. 693–701.

[16] S. M. Rumble et al. "It's Time for Low Latency". In: *Proceedings of the 13th USENIX Workshop on Hot Topics in Operating Systems*. 2011, pp. 11–15.

[17] S. Sra et al. "Adadelay: Delay Adaptive Distributed Stochastic Convex Optimization". In: (Aug. 20, 2015). arXiv: 1508.05003 [cs, math, stat].

[18] H. Sutter. "The Free Lunch Is over: A Fundamental Turn toward Concurrency in Software". In: *Dr. Dobb's journal* 30.3 (2005), pp. 202–210.

[19] H. Sutter. *Welcome to the Jungle*. 2011-11-08T14:34:45+00:00. URL: https://herbsutter.com/welcome-to-the-jungle/.

[20] P. Tseng. "On the Rate of Convergence of a Partially Asynchronous Gradient Projection Algorithm". In: *SIAM Journal on Optimization* 1.4 (Nov. 1, 1991), pp. 603–619.

[21] P. Tseng, D. Bertsekas, and J. Tsitsiklis. "Partially Asynchronous, Parallel Algorithms for Network Flow and Other Problems". In: *SIAM Journal on Control and Optimization* 28.3 (Mar. 1, 1990), pp. 678–710.