

Asynchronous Distributed Optimization with Heterogeneous Regularizations and Normalizations

Stefan Hochhaus^{*,†} and Matthew T. Hale^{*}

Abstract

As multi-agent networks grow in size and scale, they become increasingly difficult to synchronize, though agents must work together even when generating and sharing different information at different times. Targeting such cases, this paper presents an asynchronous optimization framework in which the time between successive communications and computations is unknown and unspecified for each agent. Agents' updates are carried out in blocks, with each agent updating only a small subset of all decision variables. To provide robustness to asynchrony, each agent uses an independently chosen Tikhonov regularization. Convergence is measured with respect to a weighted block-maximum norm in which convergence of agents' blocks can be measured in different p -norms and weighted differently to heterogeneously normalize problems. Asymptotic convergence is shown and convergence rates are derived explicitly in terms of a problem's parameters, with only mild restrictions imposed upon them. Simulation results are provided to verify the theoretical developments made.

I. INTRODUCTION

Distributed optimization techniques have been applied in many areas ranging from sensor networks [1], [2], [3] and communications [4], [5], to robotics [6] and smart power grids [7]. With this diversity in applications, there have emerged correspondingly diverse problem formulations which address a wide variety of practical considerations. As multi-agent systems become increasingly complex, a key practical consideration is the ability to tightly couple agents and the timing of their behaviors. Often, perfect synchrony among agents' communications and computations is difficult or impossible because closely coupling all agents in large networks is also difficult or impossible. Instead, one must sometimes utilize information that is asynchronously generated and shared. This paper examines how to do so in a distributed optimization setting.

There is a significant existing literature on distributed optimization, including a large corpus of work on asynchronous optimization. One common approach is to assume that delays in communications and computations are bounded, and this approach is used for example in [8], [9], [10], [11], [12], [13], [14], [15], [16], and the delay bound parameter explicitly appears in convergence rates in [8], [11], [12], [14], [16]. However, in some cases, delay bounds cannot be enforced. For example, agents with mutually interfering communications may be unable to ensure that delay lengths stay below a certain threshold because delays are outside their control. Similarly, agents facing anti-access/area-denial (A2AD) measures may be unable to predict when transmissions will be received or even

^{*}The authors are with the Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL, USA. Email: smhochhaus@ufl.edu, matthewhale@ufl.edu.

[†] Corresponding author.

measure delay lengths at all. As a result, some works have addressed asynchronous optimization with unbounded delays. Early work in this area includes [17], as well as [18], which gives a textbook-level treatment and simplified proof of the main results in [17].

Work in [17] was expanded upon in [19], where it was shown that a fixed Tikhonov regularization implies the existence of the nested sets required in [17] for asymptotic convergence. However, developments in [19] require every agent to apply the same regularization, which can be difficult to enforce and verify in practice, especially in large decentralized networks. Moreover, convergence in [19] is measured with respect to the same un-weighted norm for all agents. There is a wide variety of statistical and machine learning problems which must be normalized due to disparate numerical scales across potentially many orders of magnitude [20], and which may require measuring convergence of different components in different norms. While such problems are commonly solved using distributed optimization techniques, they are not accounted for by the work in [19]. Therefore, a fundamentally new approach is required to account for heterogeneous regularizations and normalizations in the setting of distributed optimization.

In this paper we develop an asynchronous optimization framework to address this gap. In particular, we examine set-constrained optimization problems with potentially non-separable cost functions, and we allow agents' communications and computations to be arbitrarily asynchronous, subject only to mild assumptions. Agents are permitted to independently choose regularization parameters with no restrictions on the disparity between them. Under these conditions, agents' convergence is measured with respect to a weighted block-maximum norm which allows for heterogeneous normalizations of agents' distance to an optimum in order to accommodate problems with different numerical scales. Convergence rates are developed in terms of agents' communications and computations without specifying when they must occur. The framework developed in this paper uses a block-based update scheme in which each agent updates only a subset of all decision variables in a problem in order to provide a scalable update law for large convex programs. The contributions of this paper therefore consist of a scalable optimization framework that accommodates heterogeneous regularizations and normalizations, together with its convergence rate.

The rest of the paper is organized as follows. Section II defines the optimization problems to be solved and regularizations used. Next, Section III defines the block-based multi-agent update law, and Section IV proves its convergence and derives its convergence rate. After that, Section V presents simulation results and Section VI provides concluding remarks.

II. TIKHONOV REGULARIZATION AND PROBLEM STATEMENT

In this section we describe the class of problems to be solved and the assumptions imposed upon problem data. We then introduce heterogeneous regularizations and the need for heterogeneous normalizations. Then we give a formal problem statement that is the focus of the remainder of the paper.

We consider convex optimization problems spread across teams of agents. In particular, we consider teams comprised of N agents, where agents are indexed over $i \in [N] := \{1, \dots, N\}$. Agent i has a decision variable $x_i \in \mathbb{R}^{n_i}$, $n_i \in \mathbb{N}$, which we refer to as its state, and we allow for $n_i \neq n_j$ when $i \neq j$. The state x_i is subject to the set constraint $x_i \in X_i \subset \mathbb{R}^{n_i}$, which can represent, e.g., that a mobile robot must stay in a given area. We

make the following assumption about each X_i .

Assumption 1: For all $i \in [N]$, the set $X_i \subset \mathbb{R}^{n_i}$ is non-empty, compact, and convex. \triangle

Towards making a formal problem statement, we aggregate agents' set constraints by defining $X := X_1 \times \dots \times X_N$, and Assumption 1 ensures that X is also non-empty, compact, and convex. We further define the ensemble state as $x := (x_1^T, \dots, x_N^T)^T \in X \subset \mathbb{R}^n$, where $n = \sum_{i \in [N]} n_i$. We consider problems in which each agent has a local objective function f_i to minimize, which can represent, e.g., a mobile robot's desire to minimize its distance to a target location; only agent i needs to know f_i . The agents are also collectively subject to a coupling cost c , which can represent the cost of communication congestion in a network, and we allow for c to be non-separable. We then make the following assumption about the functions f_i and c .

Assumption 2: The functions $f_i, i \in [N]$, and c are convex and C^2 (twice continuously differentiable) in x_i and x , respectively. \triangle

In particular, ∇f is Lipschitz and we denote its Lipschitz constant by L . The sum of these costs then gives the aggregate cost function

$$f(x) := c(x) + \sum_{i \in [N]} f_i(x_i),$$

and the agents will jointly minimize f . For simplicity of the forthcoming analysis, we assume that f has a unique minimizer. To endow f with an inherent robustness to asynchrony, we will regularize it before agents start optimizing. In particular, we regularize f on a per-agent basis, where agent i uses the regularization parameter $\alpha_i > 0$ and where we allow $\alpha_i \neq \alpha_j$ when $i \neq j$. Regularizing f makes it strongly convex, and this will be shown to provide robustness to asynchrony below. The regularized form of f is denoted f_A , and is defined as

$$f_A(x) := f(x) + \frac{1}{2}x^T A x,$$

where $A = \text{diag}(\alpha_1 I_{n_1}, \dots, \alpha_N I_{n_N})$, and where I_{n_i} is the $n_i \times n_i$ identity matrix.

In some optimization settings, some decision variables evolve at drastically different numerical scales [20]. To more meaningfully evaluate the convergence of agents with respect to one another, it would be useful to normalize each agent's distance to an optimum to prevent the error of one agent dominating the convergence analysis. Allowing heterogeneous normalizations would therefore give a more useful estimate of the distance to an optimum, and this should be accounted for by our framework. Moreover, each agent may wish to evaluate the convergence of its own state using a particular p -norm. Therefore, our framework should accommodate agents measuring the distance to an optimum in different norms. Bearing these criteria in mind, we now state the problem that is the focus of the rest of the paper.

Problem 1: For a team of N agents,

$$\underset{x \in X}{\text{minimize}} \quad f_A(x)$$

while measuring convergence with heterogeneous normalization constants and norms across the agents. \diamond

Section III specifies the structure of the asynchronous communications and computations used to solve Problem 1.

III. BLOCK-BASED MULTI-AGENT UPDATE LAW

To define the exact update law for each agent's state, we must describe what information is stored and how agents communicate. Each agent will store a vector containing its own states and those of agents it communicates with. Each agent only updates its own states within the vector it stores onboard. States stored onboard agent i which correspond to other agents' states are only updated when those agents send their states to agent i . This type of block-based update can be used to capture, for example, when an agent does not have the information required to update other agents' states, or when it is desirable to parallelize updates to reduce each agent's computational burden.

Formally, we will denote agent i 's full vector of states by x^i . Agent i 's own states in this vector are then denoted by x_i^i . The current values stored onboard agent i for agent j are denoted by x_j^i . At timestep k , agent i 's full state vector is denoted $x^i(k)$, with its own states denoted $x_i^i(k)$ and those of agent j denoted $x_j^i(k)$. At any single timestep, agent i may or may not update its states due to asynchrony in agents' computations, and the times of these updates must be accounted for. We define the set K^i to be the collection of time indices k at which agent i updates x_i^i ; agent i does not compute an update for time indices $k \notin K^i$. Using this notation, agent i 's update law can be written as

$$x_i^i(k+1) = \begin{cases} x_i^i(k) - \gamma \nabla_i f_A(x^i(k)) & k \in K^i \\ x_i^i(k) & k \notin K^i \end{cases},$$

where agent i uses stepsize $\gamma > 0$, which will be bounded below. Here $\nabla_i f_A := \frac{\partial f_A}{\partial x_i}$ is the gradient of the regularized cost function with respect to x_i . The significance of agent i 's choice of regularization parameter can be seen by expanding $\nabla_i f_A(x^i(k))$ as $\nabla_i f_A(x^i(k)) = \nabla_i f(x^i(k)) + \alpha_i x_i^i(k)$, where $\alpha_i > 0$ is set by agent i alone.

In order to account for communication delays we use $\tau_j^i(k)$ to denote the time at which the value of $x_j^i(k)$ was originally computed by agent j . For example, if agent j computes a state update at time k_a and immediately transmits it to agent i , then agent i may receive this state update at time $k_b > k_a$ due to communication delays. Then τ_j^i is defined so that $\tau_j^i(k_b) = k_a$, the time at which agent j originally computed the update just received by agent i . Concerning K^i and $\tau_j^i(k)$, we have the following assumption.

Assumption 3: For all $i \in [N]$, the set K^i is infinite. Moreover, for all $i \in [N]$ and $j \in [N] \setminus \{i\}$, if $\{k_d\}_{d \in \mathbb{N}}$ is a sequence in K^i tending to infinity, then

$$\lim_{d \rightarrow \infty} \tau_j^i(k_d) = \infty.$$

△

Assumption 3 is quite mild in that it simply requires that no agent ever permanently stop updating and sharing information. For $i \neq j$, the sets K^i and K^j need not have any relationship because agents' updates are asynchronous. The entire update law for all agents can then be written as follows.

Algorithm 1: For all $i \in [N]$ and $j \in [N] \setminus \{i\}$, execute

$$\begin{aligned} x_i^i(k+1) &= \begin{cases} x_i^i(k) - \gamma \nabla_i f_A(x^i(k)) & k \in K^i \\ x_i^i(k) & k \notin K^i \end{cases} \\ x_j^i(k+1) &= \begin{cases} x_j^j(\tau_j^i(k+1)) & \text{i receives j's state at k+1} \\ x_j^i(k) & \text{otherwise} \end{cases} . \end{aligned}$$

◇

In Algorithm 1 we see that x_j^i changes only when agent i receives a transmission from agent j ; otherwise it remains constant. Agent i can therefore reuse old values of agents j 's state many times and can reuse different agents' states different numbers of times. Showing convergence of this update law must take these delays into account, and that is the subject of the next section.

IV. CONVERGENCE OF ASYNCHRONOUS OPTIMIZATION

In this section we prove the convergence of the multi-agent block update law in Algorithm 1. We first define the block-maximum norm used to measure convergence and then define a collection of nested sets that will be used to show asymptotic convergence of all agents adapted from the approach in [19]. Then a convergence rate is developed using parameters from these sets.

A. Block-Maximum Norms

We begin by analyzing the convergence of the optimization algorithm using block maximum norms similar to those defined in [17], [18], and [19], and we do so to accommodate the need for heterogeneous normalizations and norms in Problem 1. Due to asynchrony in agents' communications, we will generally have $x^i(k) \neq x^j(k)$ for all agents i and j and all timesteps k . We will refer to x_i^i as the i^{th} block of x^i and x_j^i as the j^{th} block of x^i . With these blocks defined we next define the block-maximum norm that will be used to measure convergence below.

Definition 1: Let $x \in \mathbb{R}^n$ consist of N blocks, with $x_i \in \mathbb{R}^{n_i}$ being the i^{th} block. The i^{th} block is weighted by some normalization constant $w_i \geq 1$ and is measured in the p_i -norm for some $p_i \in [1, \infty]$. The norm of the full vector x is defined as the maximum norm of any single block, i.e.,

$$\|x\|_{\max} := \max_{i \in [N]} \frac{\|x_i\|_{p_i}}{w_i}.$$

△

The following lemma allows us to upper-bound the induced block-maximum matrix norm by the Euclidian matrix norm, which will be used below in our convergence analysis. In this lemma, we use the notion of a block of an $n \times n$ matrix. Given a matrix $B \in \mathbb{R}^{n \times n}$, where $n = \sum_{i=1}^N n_i$, the i^{th} block of B , denoted $B^{[i]}$, is the $n_i \times n$ matrix formed by rows of B with indices $\sum_{k=1}^{i-1} n_k + 1$ through $\sum_{k=1}^i n_k$. We then have the following result.

Lemma 1: Let $p_{\min} := \min_{i \in [N]} p_i$ and let $w_{\min} = \min_{i \in [N]} w_i$. Then for all $B \in \mathbb{R}^{n \times n}$,

$$\|B\|_{\max} \leq \begin{cases} n^{(p_{\min}^{-1} - \frac{1}{2})} w_{\min}^{-1} \|B\|_2 & p_{\min} < 2 \\ \frac{1}{w_{\min}} \|B\|_2 & p_{\min} \geq 2 \end{cases} .$$

Proof: For $B^{[i]}$ the i^{th} block of B and any $x \in \mathbb{R}^n$, by definition we have

$$\frac{\|B^{[i]}x\|_{p_i}}{w_i} = \frac{1}{w_i} \left(\sum_{k=1}^{n_i} \left| \sum_{j=1}^n B_{k,j}^{[i]} x_j \right|^{p_i} \right)^{\frac{1}{p_i}}. \quad (1)$$

From the definition of a p -norm, the right side of Equation (1) will always be non-negative. Thus summing the right-hand side over every block results in

$$\frac{\|B^{[i]}x\|_{p_i}}{w_i} \leq \sum_{i=1}^N \left(\frac{1}{w_i} \left(\sum_{k=1}^{n_i} \left| \sum_{j=1}^n B_{k,j}^{[i]} x_j \right|^{p_i} \right)^{\frac{1}{p_i}} \right).$$

Next, recalling that $\|x\|_q \leq \|x\|_r$ for all vectors $x \in \mathbb{R}^n$ and all $q \geq r > 0$, we find that

$$\begin{aligned} \frac{\|B^{[i]}x\|_{p_i}}{w_i} &\leq \frac{1}{w_{\min}} \sum_{i=1}^N \left(\sum_{k=1}^{n_i} \left| \sum_{j=1}^n B_{k,j}^{[i]} x_j \right|^{p_i} \right)^{\frac{1}{p_i}} \\ &\leq \frac{1}{w_{\min}} \sum_{i=1}^N \left(\sum_{k=1}^{n_i} \left| \sum_{j=1}^n B_{k,j}^{[i]} x_j \right|^{p_{\min}} \right)^{\frac{1}{p_{\min}}}. \end{aligned}$$

This then allows us to express the sum over all rows of B via

$$\frac{\|B^{[i]}x\|_{p_i}}{w_i} \leq \frac{1}{w_{\min}} \left(\sum_{l=1}^n \left| \sum_{j=1}^n B_{l,j} x_j \right|^{p_{\min}} \right)^{\frac{1}{p_{\min}}}.$$

If $p_{\min} \geq 2$, then $\|B^{[i]}x\|_{p_i} \leq \|B^{[i]}x\|_2$ for all p_i . If $p_{\min} < 2$, we recall that $\|x\|_l \leq \|x\|_{p_{\min}} \leq n^{(p_{\min}^{-1} - l^{-1})} \|x\|_l$, which follows from Hölder's inequality for $0 < p_{\min} < l$, and observe that $\|B^{[i]}x\|_{p_i} \leq \|B^{[i]}x\|_{p_{\min}} \leq n^{(p_{\min}^{-1} - \frac{1}{2})} \|Bx\|_2$. Combining these inequalities we find that

$$\frac{\|B^{[i]}x\|_{p_i}}{w_i} \leq \begin{cases} n^{(p_{\min}^{-1} - \frac{1}{2})} w_{\min}^{-1} \|Bx\|_2 & p_{\min} < 2 \\ \frac{1}{w_{\min}} \|Bx\|_2 & p_{\min} \geq 2 \end{cases}$$

for all i . Thus the weighted block maximum norm of Bx for any $x \in \mathbb{R}^n$ can be bounded as

$$\begin{aligned} \|Bx\|_{\max} &= \max_{i \in [N]} \frac{\|B^{[i]}x\|_{p_i}}{w_i} \\ &\leq \begin{cases} n^{(p_{\min}^{-1} - \frac{1}{2})} w_{\min}^{-1} \|Bx\|_2 & p_{\min} < 2 \\ \frac{1}{w_{\min}} \|Bx\|_2 & p_{\min} \geq 2 \end{cases}, \end{aligned}$$

and the lemma follows by taking the supremum over all unit vectors x . ■

B. Convergence Via Nested Sets

We now begin the convergence analysis for the block-based update law in Algorithm 1 where agents are asynchronously optimizing. In order for this system to converge using the communications described in the previous section, we construct a sequence of sets, $\{X(s)\}_{s \in \mathbb{N}}$, based on work in [17] and [18]. Below we use the notation $\hat{x}_A := \arg \min_{x \in X} f_A(x)$ to specify the minimizer of the regularized cost function f_A . We state the conditions

imposed upon these sets as an assumption, and this assumption will be shown below to be satisfied using the heterogeneous regularization applied by A .

Assumption 4: The sets $\{X(s)\}_{s \in \mathbb{N}}$ satisfy:

- 1) $\cdots \subset X(s+1) \subset X(s) \subset \cdots \subset X$
- 2) $\lim_{s \rightarrow \infty} X(s) = \{\hat{x}_A\}$
- 3) $X_i(s) \subset X_i$ for all $i \in [N]$ and $s \in \mathbb{N}$ such that $X(s) = X_1(s) \times \cdots \times X_N(s)$
- 4) $\theta_i(y) \in X_i(s+1)$, where $\theta_i(y) := y_i - \gamma \nabla_i f_A(y)$ for all $y \in X(s)$ and $i \in [N]$. \triangle

Assumptions 4.1 and 4.2 together show that these sets are nested as they converge to the minimum \hat{x}_A . Assumption 4.3 allows for the blocks to be updated independently by the agents, and Assumption 4.4 ensures that state updates always progress down the chain of nested sets such that only forward progress toward \hat{x}_A is made. It is shown in [17] and [18] that the existence of such a sequence of sets implies asymptotic convergence of the asynchronous update law in Algorithm 1, and we therefore use this construction to show asymptotic convergence in this paper. Defining the Lipschitz constant of $\nabla_i f_A$ as L_i , we further define $L_{\max} := \max_{i \in [N]} L_i$, and then define the constant

$$q = \max \left\{ \max_{i \in [N]} |1 - \gamma \alpha_i|, \max_{i \in [N]} |1 - \gamma L_i| \right\}.$$

Letting $\gamma \in \left(0, \frac{2}{L_{\max}}\right)$ and $\alpha \in (0, L_{\max})$, we find $q \in (0, 1)$; a proof for this can be seen in [21]. We then proceed to define D_o as

$$D_o := \max_{i \in [N]} \|x^i(0) - \hat{x}_A\|_{\max},$$

which is the worst-performing block onboard any agent with respect to distance to \hat{x}_A at timestep 0. We then define the sequence of sets $\{X(s)\}_{s \in \mathbb{N}}$ as

$$X(s) = \{y \in X : \|y - \hat{x}_A\|_{\max} \leq q^s D_o\}, \quad (2)$$

and this construction is shown in the following theorem to satisfy Assumption 4, thereby ensuring asymptotic convergence of Algorithm 1.

Theorem 1: The collection of sets $\{X(s)\}_{s \in \mathbb{N}}$ as defined in Equation (2) satisfies Assumption 4.

Proof: For Assumption 4.1 we see that

$$X(s+1) = \{y \in X : \|y - \hat{x}_A\|_{\max} \leq q^{s+1} D_o\}.$$

Since $q \in (0, 1)$, we have $q^{s+1} < q^s$, which results in $\|y - \hat{x}_A\|_{\max} \leq q^{s+1} D_o < q^s D_o$. Then $y \in X(s+1)$ implies $y \in X(s)$ and $X(s+1) \subset X(s) \subset X$, as desired.

From Assumption 4.2 we find

$$\begin{aligned} \lim_{s \rightarrow \infty} X(s) &= \lim_{s \rightarrow \infty} \{y \in X : \|y - \hat{x}_A\|_{\max} \leq q^s D_o\} \\ &= \{y \in X : \|y - \hat{x}_A\|_{\max} \leq 0\} \\ &= \{\hat{x}_A\}, \end{aligned}$$

and Assumption 4.2 is therefore satisfied. The structure of the weighted block-maximum norm then allows us to see that $\|y - \hat{x}_A\|_{\max} \leq q^s D_o$ if and only if $\frac{1}{w_i} \|y_i - \hat{x}_{A,i}\|_{p_i} \leq q^s D_o$ for all $i \in [N]$. It then follows that

$$X_i(s) = \left\{ y_i \in X_i : \frac{1}{w_i} \|y_i - \hat{x}_{A,i}\|_{p_i} \leq q^s D_o \right\},$$

which shows $X(s) = X_1(s) \times \cdots \times X_N(s)$, thus satisfying Assumption 4.3.

In order to show Assumption 4.4 is satisfied we recall the following exact expansion of ∇f_A :

$$\begin{aligned} \nabla f_A(y) - \nabla f_A(\hat{x}_A) &= \int_0^1 \nabla^2 f_A(\hat{x}_A + \tau(y - \hat{x}_A))(y - \hat{x}_A) d\tau \\ &= \left(\int_0^1 \nabla^2 f_A(\hat{x}_A + \tau(y - \hat{x}_A)) d\tau \right) \cdot (y - \hat{x}_A) \\ &=: H(y)(y - \hat{x}_A), \end{aligned} \tag{3}$$

where we have defined

$$H(y) = \int_0^1 \nabla^2 f_A(\hat{x}_A + \tau(y - \hat{x}_A)) d\tau.$$

We then see that for $y \in X(s)$,

$$\begin{aligned} \frac{\|\theta_i(y) - \hat{x}_{A,i}\|_{p_i}}{w_i} &= \frac{1}{w_i} \|y_i - \gamma \nabla_i f_A(y) - \hat{x}_{A,i} + \gamma \nabla_i f_A(\hat{x}_A)\|_{p_i} \\ &\leq \max_{i \in [N]} \frac{1}{w_i} \|y_i - \gamma \nabla_i f_A(y) - \hat{x}_{A,i} + \gamma \nabla_i f_A(\hat{x}_A)\|_{p_i} \\ &= \|y - \hat{x}_A - \gamma \nabla f_A(y) + \gamma \nabla f_A(\hat{x}_A)\|_{\max} \\ &= \|y - \hat{x}_A - \gamma (\nabla f_A(y) - \nabla f_A(\hat{x}_A))\|_{\max} \\ &= \|y - \hat{x}_A - \gamma H(y)(y - \hat{x}_A)\|_{\max} \\ &\leq \|I - \gamma H(y)\|_{\max} \|y - \hat{x}_A\|_{\max} \\ &\leq \begin{cases} \frac{n^{(p_{\min} - \frac{1}{2})}}{w_{\min}} \|I - \gamma H(y)\|_2 \|y - \hat{x}_A\|_{\max} & p_{\min} < 2 \\ \frac{1}{w_{\min}} \|I - \gamma H(y)\|_2 \|y - \hat{x}_A\|_{\max} & p_{\min} \geq 2 \end{cases}, \end{aligned}$$

where we have used Equation (3) in the fourth equality and Lemma 1 in the third inequality. We then define the vector $\nabla f_A = (\nabla_1 f_A, \dots, \nabla_N f_A)^T$ which has a Lipschitz constant of $M = \sqrt{\sum_{i=1}^N L_i^2}$. It then follows from the definition of f_A that $A \preceq H(\cdot) \preceq MI$, which implies that the eigenvalues of $H(\cdot)$ are bounded below by the smallest diagonal entry of A and above by M . Since $H(y)$ is a symmetric matrix it follows that

$$\begin{aligned} \|I - \gamma H(y)\|_2 &= \max \{ |\lambda_{\min}(I - \gamma H(y))|, |\lambda_{\max}(I - \gamma H(y))| \} \\ &= \max \left\{ \max_{i \in [N]} |1 - \gamma \alpha_i|, \max_{i \in [N]} |1 - \gamma L_i| \right\} \\ &= q, \end{aligned}$$

where $\lambda_{\min}(\cdot)$ and $\lambda_{\max}(\cdot)$ are the minimum and maximum eigenvalues of a matrix, respectively. Using the hypothesis that $y \in X(s)$, we find

$$\begin{aligned} \frac{\|\theta_i(y) - \hat{x}_{A,i}\|_{p_i}}{w_i} &\leq \begin{cases} n^{(p_{\min}^{-1} - \frac{1}{2})} w_{\min}^{-1} q \|y - \hat{x}_A\|_{\max} & p_{\min} < 2 \\ \frac{1}{w_{\min}} q \|y - \hat{x}_A\|_{\max} & p_{\min} \geq 2 \end{cases} \\ &\leq \begin{cases} n^{(p_{\min}^{-1} - \frac{1}{2})} w_{\min}^{-1} q^{s+1} D_o & p_{\min} < 2 \\ \frac{1}{w_{\min}} q^{s+1} D_o & p_{\min} \geq 2 \end{cases} \\ &\leq \begin{cases} q^{s+1} D_o \\ q^{s+1} D_o \end{cases}, \end{aligned}$$

where the bottom case follows from $w_{\min} \geq 1$ and the top case follows from $w_{\min} \geq 1$ and $p_{\min}^{-1} - \frac{1}{2} < 1$. Then $\theta_i(y) \in X_i(s+1)$ and Assumption 4.4 is satisfied. \blacksquare

As noted above, the fact that the construction in Equation (2) satisfies Assumption 4 implies asymptotic convergence of Algorithm 1 for all $i \in [N]$ from [17] and [18]. With this in mind, we next derive a rate of convergence for Algorithm 1.

C. Convergence Rate

The structure of the sets $\{X(s)\}_{s \in \mathbb{N}}$ allows us to determine a convergence rate. However, to do so we must first define the notion of a *communication cycle*. Starting at time $k = 0$, one cycle occurs when all agents have calculated a state update and this updated state has been sent to and received by each other agent. It is only then that each agents' copy of the ensemble state is moved from $X(0)$ to $X(1)$. Once another cycle is completed the ensemble state is moved from $X(1)$ to $X(2)$. This process repeats indefinitely, and coupled with Assumption 4, means the convergence rate is geometric in the number of cycles completed, which we show now.

Theorem 2: Let Assumptions 1-4 hold and let $\gamma \in (0, \frac{2}{L_{\max}})$. At time k , if $c(k)$ cycles have been completed, then

$$\|x^i(k) - \hat{x}_A\|_{\max} \leq q^{c(k)} D_o$$

for all $i \in [N]$.

Proof: From the definition of D_o , for all $i \in [N]$ we have $x^i(0) \in X(0)$. If agent i computes a state update, then $\theta_i(x^i(0)) \in X_i(1)$ and after one cycle is completed, say at time k , we have $x^i(k) \in X(1)$ for all i . Iterating this process, after $c(\bar{k})$ cycles have been completed by some time \bar{k} , $x^i(\bar{k}) \in X(c(\bar{k}))$. The result follows by expanding the definition of $\{X(s)\}_{s \in \mathbb{N}}$. \blacksquare

Theorem 3 can be used by a network operator to bound agents' convergence by simply observing them and without specifying when or how often agents should generate or share information. Having shown convergence of Algorithm 1, we next demonstrate its performance in practice.

V. SIMULATION

In this section we present a problem to be solved using Algorithm 1. The simulation uses a network consisting of 8 nodes and 9 edges, where we define the set $\varepsilon := [9]$ as the set of indices of the edges. There are $N = 8$

agents that are users of this network and they are each tasked with routing a flow between two nodes. The network itself is shown in Figure 1; we emphasize that the nodes in the network are not the agents themselves, but instead are simply source/destination pairs for users to route flows between. The starting and ending nodes as well as the edges traversed for each agents' flow are listed in Table I.

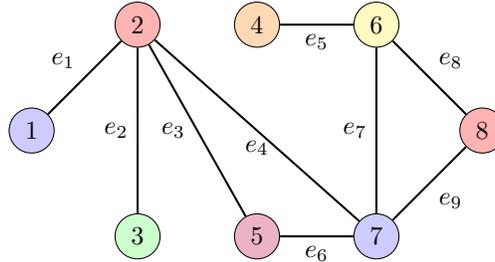


Fig. 1: The network through which eight agents must route a flow between two nodes

Agent Number	Start Node→End Node	Edges Traversed
1	1 → 7	e_1, e_3, e_6
2	2 → 8	e_4, e_7, e_8
3	3 → 4	e_2, e_4, e_7, e_5
4	5 → 6	e_3, e_4, e_7
5	1 → 4	e_1, e_3, e_6, e_7, e_5
6	3 → 8	e_2, e_4, e_9
7	4 → 5	e_5, e_8, e_9, e_6
8	6 → 2	e_7, e_4

TABLE I: Edges traversed by each agent's flow

The cost function of agent i is $f_i(x_i) = -100 \log(1 + x_i)$, and the coupling cost is $c(x) = \frac{1}{20} x^T C^T C x$, where the network connection matrix is defined as

$$C_{k,i} = \begin{cases} 1 & \text{if flow } i \text{ traverses edge } k \\ 0 & \text{otherwise} \end{cases}.$$

This problem was then implemented such that agent i had its own regularization parameter $\alpha_i > 0$, normalization constant $w_i \geq 1$, and p_i norm with $p_i \in [1, \infty]$. In particular, these parameters were chosen using $w = [12, 8, 6, 7, 6, 10, 9, 10]$ and $p = [\infty, 20, 3, 90, 6, 12, 2, 9]$, where w_i is the i^{th} element in w and p_i is defined analogously. All agents' behaviors were randomized to give each agent a 10% chance of computing an update at any timestep and to give each pair of agents a 10% chance of communicating at each timestep. Three total simulation runs were executed using the three different choices of A listed to demonstrate its effects upon convergence, with

$$A_1 = \text{diag}[3 \times 10^{-4}, 1 \times 10^{-4}, 9 \times 10^{-4}, 2 \times 10^{-4}, 0.001, 0.001, 5 \times 10^{-4}, 4 \times 10^{-4}]$$

$$A_2 = \text{diag}[0.01, 0.01, 0.003, 0.005, 0.002, 0.01, 0.005, 0.002]$$

$$A_3 = \text{diag}[0.08, 0.1, 0.1, 0.09, 0.009, 0.1, 0.08, 0.04].$$

A plot of error versus iteration count for a run with A_1 is shown in Figure 2, which shows that the regularization provided by A_1 can provide robustness to asynchrony without significantly impacting the final point obtained by Algorithm 1. In addition, close convergence to a minimizer is attained in a reasonable number of iterations, even when agents infrequently generate and share information.

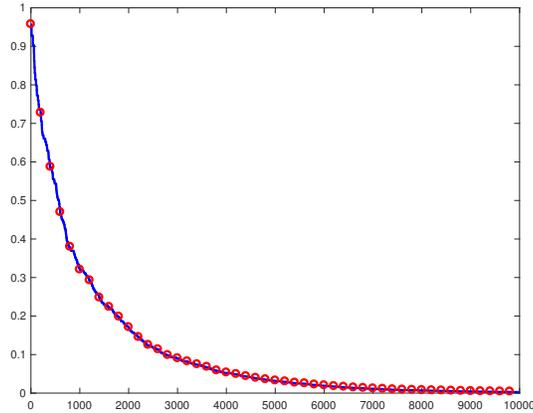


Fig. 2: Regularized and unregularized error for agent 1 where $\|A_1\| = 0.001$. Here, the regularized error is shown as a line and the unregularized error is shown by the circles. As expected, both errors converge to small final values, indicating close convergence to both \hat{x} and \hat{x}_A when $\|A\|$ is small.

To demonstrate the impact of larger regularizations, a simulation was run with A_2 , and an error plot for this run is shown in Figure 3.

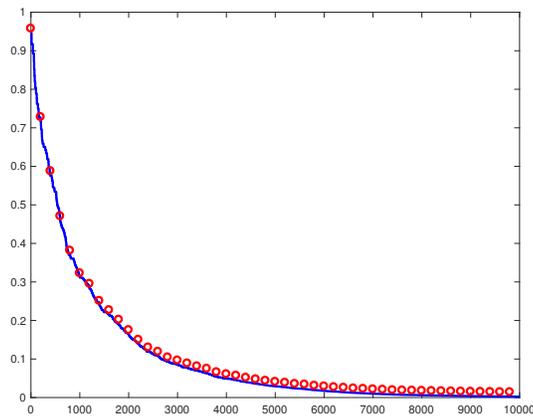


Fig. 3: Regularized and unregularized error for agent 1 where $\|A_2\| = 0.01$. The regularized error is shown as a line and the unregularized error is shown by the circles. Because $\|A\|$ is larger, the agents converge to a minimum faster, though there is a larger discrepancy between \hat{x} and \hat{x}_A , as evidenced by the asymptotic disagreement between the two curves shown here.

To further illustrate the effects of regularizing, a third and final simulation was run with A_3 , and a plot of error in this case is shown in Figure 4.

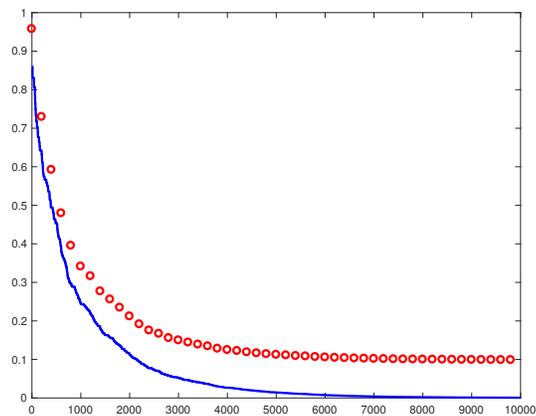


Fig. 4: Regularized and unregularized error for agent 1 where $\|A_3\| = 0.1$. The regularized error is shown as a line and the unregularized error is shown by the circles. As expected, this run converges faster (because its value of q smaller), but with the largest error in the final solution obtained, indicating that a significant acceleration in convergence comes in exchange for a less accurate solution.

To enable numerical comparisons of these convergence results, final error values for all three runs are shown in Table II, where we see that larger values of $\|A\|$ do indeed lead to larger errors.

$\ A\ $	Final Regularized Error	Final Unregularized Error
0.001	2.2575×10^{-8}	2.9558×10^{-4}
0.01	2.1837×10^{-8}	8.4922×10^{-4}
0.1	7.9827×10^{-10}	0.0848

TABLE II: Errors for agent 1

VI. CONCLUSION

This work presented an asynchronous optimization framework which allows for arbitrarily delayed communications and computations. Future extensions to this work include incorporating constraints in order to accommodate broader classes of problems [22], and using time-varying regularizations to always reach exact solutions. Future applications include use in robotic swarms where communications are unreliable and asynchrony is unavoidable.

REFERENCES

- [1] M. Khan, G. Pandurangan, and V. S. A. Kumar, “Distributed algorithms for constructing approximate minimum spanning trees in wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 1, pp. 124–139, Jan 2009.
- [2] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, “Coverage control for mobile sensing networks,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, pp. 243–255, April 2004.

- [3] M. Rabbat and R. Nowak, "Distributed optimization in sensor networks," in *Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks*, ser. IPSN '04. New York, NY, USA: ACM, 2004, pp. 20–27.
- [4] D. Mitra, *An Asynchronous Distributed Algorithm for Power Control in Cellular Radio Systems*. Boston, MA: Springer US, 1994, pp. 177–186.
- [5] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255–312, Jan 2007.
- [6] D. E. Soltero, M. Schwager, and D. Rus, "Decentralized path planning for coverage tasks using gradient descent adaptive control," *The International Journal of Robotics Research*, vol. 33, no. 3, pp. 401–425, 2014.
- [7] S. Caron and G. Kesidis, "Incentive-based energy consumption scheduling algorithms for the smart grid," in *2010 First IEEE International Conference on Smart Grid Communications*, Oct 2010, pp. 391–396.
- [8] A. I. Chen and A. Ozdaglar, "A fast distributed proximal-gradient method," in *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*. IEEE, 2012.
- [9] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *IEEE Transactions on Automatic Control*, vol. 48, no. 6, June 2003.
- [10] L. Moreau, "Stability of multiagent systems with time-dependent communication links," *IEEE Transactions on Automatic Control*, vol. 50, no. 2, pp. 169–182, Feb 2005.
- [11] A. Nedić and A. Ozdaglar, "On the rate of convergence of distributed subgradient methods for multi-agent optimization," in *Proceedings of the 46th IEEE Conference on Decision and Control 2007, CDC, 2007*, pp. 4711–4716.
- [12] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, Jan 2009.
- [13] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, April 2010.
- [14] A. Olshevsky and J. N. Tsitsiklis, "Convergence speed in distributed consensus and averaging," *SIAM J. Control Optim.*, vol. 48, no. 1, pp. 33–55, Feb. 2009.
- [15] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, May 2005.
- [16] B. Touri and A. Nedić, "Distributed consensus over network with noisy links," in *2009 12th International Conference on Information Fusion*, July 2009, pp. 146–154.
- [17] D. P. Bertsekas and J. N. Tsitsiklis, "Convergence rate and termination of asynchronous iterative algorithms," in *Proceedings of the 3rd International Conference on Supercomputing*, ser. ICS '89. New York, NY, USA: ACM, 1989, pp. 461–470.
- [18] D. P. Bertsekas and J. Tsitsiklis, "Parallel and distributed computation," *Upper Saddle River*, 1989.
- [19] M. T. Hale, A. Nedić, and M. Egerstedt, "Asynchronous multiagent primal-dual optimization," *IEEE Transactions on Automatic Control*, vol. 62, no. 9, pp. 4421–4435, Sept 2017.
- [20] C. Bishop, *Neural Networks for Pattern Recognition*, ser. Advanced Texts in Econometrics. Clarendon Press, 1995.
- [21] B. T. Polyak, "Introduction to optimization. translations series in mathematics and engineering," *Optimization Software*, 1987.
- [22] M. Hale and Y. Wardi, "Mode scheduling under dwell time constraints in switched-mode systems," in *2014 American Control Conference*, June 2014, pp. 3954–3959.