Noname manuscript No.
(will be inserted by the editor)

Distributed Nonconvex Optimization with Exponential Convergence Rate via Hybrid Systems Methods

Katherine R. Hendrickson, Dawn M. Hustig-Schultz, Matthew T. Hale, Ricardo G. Sanfelice

the date of receipt and acceptance should be inserted later

Abstract We present a hybrid systems framework for distributed multi-agent optimization in which agents execute computations in continuous time and communicate in discrete time. The optimization algorithm is analogous to a continuous-time form of parallelized coordinate descent. Agents implement an update-and-hold strategy in which gradients are computed at communication times and held constant during flows between communications. The completeness of solutions under these hybrid dynamics is established. Then, we prove that this system is globally exponentially stable to a minimizer of a possibly nonconvex, smooth objective function that satisfies the Polyak-Łojasiewicz (PL) condition. Simulation results are presented for three different applications and illustrate the convergence rates and the impact of initial conditions upon convergence.

Communicated by Sebastian U Stich.

Keywords Nonconvex optimization, distributed optimization, gradient methods, first-order algorithms

Katherine Hendrickson
Merlin Labs
Boston, MA, USA 02111
kat.hendrickson@merlinlabs.com

Dawn M. Hustig-Schultz
University of California, Santa Cruz
Santa Cruz, CA, USA 95064
dhustigs@ucsc.edu

Matthew T. Hale, Corresponding Author
Georgia Institute of Technology
Atlanta, GA, USA 30332
matthale@gatech.edu

Ricardo G. Sanfelice
University of California, Santa Cruz
Santa Cruz, CA, USA 95064
ricardo@ucsc.edu

1 Introduction

1.1 Motivation

Optimization problems arise in many areas of engineering, including machine learning [33], communications [24], robotics [38], and others. Across all application areas, the goal is to design an algorithm that will converge to a minimum of an objective function, possibly under some constraints. Recently, there has been increased interest in studying optimization algorithms in continuous time to use tools from dynamical systems to establish convergence to minimizers, e.g., in [13, 29, 34]. While a large body of optimization work focuses on convex optimization, nonconvex problems often arise in a variety of fields, including machine learning [10, 11, 21] and communication networks [8], and there has arisen interest in establishing convergence guarantees for non-convex problems.

In this paper, we develop a multi-agent framework for nonconvex optimization in which agents' computations are modeled in continuous time. This is motivated by two factors. First, we wish to leverage the large collection of tools from dynamical systems to analyze multi-agent optimization and connect to the growing body of work that uses continuous-time models of computation. Second, there also exist controllers for multi-agent systems that are designed to operate in continuous time to minimize some objective function, e.g., in consensus [25] and coverage control [9], and our analyses will connect our work to such systems. However, while individual agents' computations occur in continuous time, their communications are most naturally modeled in discrete time because communicated information arrives at its recipients at individual instants in time. Thus, the joint modeling and analysis of agents' computations and communications creates a mixture of continuous- and discrete-time elements, which leads us to develop a hybrid systems framework for multi-agent optimization.

The framework that we develop can solve a class of problems that includes some non-convex problems. In particular, we consider smooth objective functions that satisfy the Polyak-Łojasiewicz (PL) inequality [28]. Recent interest in the PL inequality and related properties has led to the development of discrete-time nonconvex optimization approaches [1, 7, 16, 19, 22], including distributed algorithms [36, 37, 41]. Problems that satisfy the PL inequality include matrix factorization [35], minimizing logistic loss over a compact set [19], and the training of some neural networks [7]. The set of functions that satisfy the PL inequality also includes those that are strongly convex, and our developments therefore apply to strongly convex functions, which are commonly studied in distributed optimization settings [2].

The algorithm that we develop is analogous to a hybrid systems version of parallelized block coordinate descent [4], in which each agent updates only a subset of all decision variables using continuous-time computations and agents communicate these updated values in discrete time to other agents. In the framework that we develop, all agents' communications are intermittent; agents' communications occur when a timer reaches zero, at which point the timer is reset to some value within a specified range. Agents use a sample-and-hold strategy in which gradients are computed at the communication times and then held constant and used in computations until the next communication event. This approach is inspired by recent work [26] that has successfully applied it to synchronization problems.

1.2 Contributions

We leverage analytical tools from the theory of hybrid systems to prove that this algorithmic framework has several desirable properties, and our contributions are:

- We define a hybrid system model for distributed optimization. To the best of our knowledge, this is the first distributed hybrid system model that uses a parallelized approach for nonconvex problems rather than a consensus-based approach.
- We show that under our model, every maximal solution is complete, with the time domain allowing arbitrarily large ordinary time t. As a result, there are no theoretical obstructions to running this algorithm for arbitrarily long periods of time.
- We use Lyapunov analysis to show that, even under intermittent information sharing, the hybrid optimization algorithm is globally exponentially stable to a minimizer of an objective function, and we derive an explicit convergence rate in terms of system parameters.
- We show robustness to inaccuracies in the measurement of the times at which communication events occur.
- Finally, we present three different applications, including those with nonconvex objective functions, that demonstrate the performance of our model.

1.3 Related Work

The developments in this paper can be regarded as hybrid counterparts to "classical" discrete-time algorithms in multi-agent optimization [4]. Related research in multi-agent continuous-time optimization includes [14, 23, 29], though those works all use a consensus-based optimization framework in which both computations and communications are modeled as occurring in continuous time. In this work, we avoid continuous-time communications in order to model problems in which constant communications are not possible, e.g., over long distances, or simply undesirable, e.g., when battery power is limited.

The closest works to the current paper are [20], [26], which also study multi-agent optimization algorithms with continuous-time computations and discrete-time communications. However, those works also use consensus-based optimization algorithms in which each agent has a local objective function, computes new values for all decision variables, and averages its decision variables with other agents. In contrast, we consider all agents having a common objective function and we only require each agent to compute updates to a small subset of the decision variables in a problem. This approach has the benefit that each agent's computational burden grows slowly as a problem grows since each agent updates only $\frac{d}{N}$ decision variables on average. When N is large, the value of $\frac{d}{N}$ can grow quite slowly as a function of d, which results in only small increases in each agent's computational burden.

In addition, the hybrid system model that we develop offers several analytical features. First, existing block coordinate descent algorithms are typically modeled in discrete time. When used in a continuous-time system, these types of discrete-time computations will be done with samples of continuous-time state values, though convergence analyses for the discrete-time updates will apply only to the samples, not to the continuously evolving intersample state values. However, within the hybrid

framework in this paper, we analyze the time evolution of both the sampled state values and the intersample state values, which characterizes state evolution at all points in time. Second, when a hybrid system is well-posed and has a compact pre-asymptotically stable set, it follows that such pre-asymptotic stability is robust to small perturbations [15, Theorem 7.21]. In this paper, we show that this robustness applies when solving the problems we consider, and thus our use of a hybrid model lets our analysis automatically inherit these robustness properties.

This paper is an extension of the conference paper [18] which applied only to strongly convex objective functions. This paper modifies the previous hybrid system model to provide global convergence and reformulates all previous results to apply to objective functions that satisfy the PL inequality. This class of functions includes some non-convex functions, and all theoretical results and proofs from [18] have been reformulated to accommodate this nonconvexity. In particular, generic PL functions can have any number of local minima and need not have strongly monotone gradients, both of which differ from strongly convex functions. As described above, we are motivated to analyze PL functions because they appear in a wide range of engineering problems, and this paper therefore extends our hybrid algorithm to apply to those problems. Moreover, a tighter convergence rate is established and more applications are demonstrated via simulation. Finally, new results regarding robustness to perturbations are presented.

1.4 Organization

The rest of the paper is organized as follows. Section 2 includes our problem statement, assumptions, and algorithm. We then present our hybrid system model for multi-agent optimization in Section 3 and establish the existence of complete solutions. Section 4 proves that the hybrid multi-agent update law is globally exponentially stable, and then Section 5 shows that this exponential stability guarantee is robust to a certain class of perturbations. We include numerical results in Section 6, and we present our conclusions in Section 7.

2 Problem Statement and Algorithm Overview

In this section, we state the class of problems that we consider, and we give an overview of the hybrid optimization algorithm that is the focus of the remainder of the paper. First, we present some general notation.

Notation and Terminology. Let $\mathbb{R}$ denote the set of real numbers, let $\mathbb{R}_{\geq 0}$ denote the non-negative reals, and let $\mathbb{R}_{>0}$ denote the positive reals. Let $\mathbb{N}$ denote the non-negative integers, and let $\mathbb{N}_{>0}$ denote the positive integers. For $d \in \mathbb{N}_{>0}$, let $\mathbf{0}_d$ be a vector of zeros in $\mathbb{R}^d$ and $\mathbf{1}_d$ be a vector of ones in $\mathbb{R}^d$. Define the set $[p] := \{1, 2, \ldots, p\}$ for any $p \in \mathbb{N}_{>0}$. For vectors $x_1, x_2, \ldots, x_n$, define $\mathrm{col}(x_1, x_2, \ldots, x_n) := (x_1^\top, x_2^\top, \ldots, x_n^\top)^\top$. Throughout the paper, $\left|\cdot\right|$ denotes the Euclidean norm. We use $\mathrm{dom} f$ to denote the domain of a function f. The set $\mathcal{K}_\infty$ denotes the set of class $\mathcal{K}_\infty$ functions, i.e., functions $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ that are (i) strictly increasing, (ii) satisfy $\alpha(0) = 0$, and (iii) satisfy $\lim_{r \to \infty} \alpha(r) = \infty$. The set $\mathcal{KL}$ denotes the set of class-$\mathcal{KL}$ functions, i.e., functions $\gamma : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ such that (i) γ is non-decreasing in its first argument, (ii) γ is non-increasing in its second

argument, (iii) $\lim_{r\to 0^+} \gamma(r,s) = 0$ for each $s \in \mathbb{R}_{\geq 0}$, and (iv) $\lim_{s\to\infty} \gamma(r,s) = 0$ for each $r \in \mathbb{R}_{\geq 0}$. We use "ODE" to mean "ordinary differential equation".

2.1 Problem Formulation

We consider a group of agents jointly solving an optimization problem that may be nonconvex. Suppose there are N agents that will each execute computations locally and then share the results of those computations with other agents. For scalability, only a single agent will update each decision variable. In many practical settings, we expect bandwidth to be limited and/or agents to have limited onboard power available, which means communications should not be constant.

Under these conditions, we consider minimization problems of the following form:

Problem 1 Given an objective function $L : \mathbb{R}^n \to \mathbb{R}$,

$$\text{minimize } L(x), \qquad x \in \mathbb{R}^n$$

while requiring that (i) only one agent updates any entry of the decision variable x, and (ii) agents require only sporadic information sharing from others.

We first assume the following about the objective function L.

Assumption 1 *The function L is twice continuously differentiable and K-smooth (namely, ∇L is K-Lipschitz).* △

Rather than requiring that L be convex, we will instead consider a class functions that includes some nonconvex problems. In particular, we are interested in problems that satisfy the Polyak-Łojasiewicz (PL) inequality [19].

Assumption 2 *The set of stationary points of L, defined as $\mathcal{X}^* = \{x \in \mathbb{R}^n : \nabla L(x) = 0\}$, is non-empty, and the function L satisfies the Polyak-Łojasiewicz (PL) inequality. Namely, there exists some constant $\beta > 0$ such that*

$$\frac{1}{2}\big|\nabla L(x)\big|^2 \geq \beta\big(L(x) - L(x^*)\big)$$

for all $x \in \mathbb{R}^n$ and all $x^* \in \mathcal{X}^*$. △

If a function satisfies the PL inequality with constant β, we say that the function is β-PL. One useful property resulting from Assumption 2 is that all stationary points of L are global minima.

Lemma 1 *Let Assumption 2 hold. Then every local minimizer of L is a global minimizer.*

Proof: See Section 2.2 of [19]. □

We define

$$L^* := L(x^*) \tag{1}$$

as the global minimum value of L, and, under Assumption 2 it is attained at every $x^* \in \mathcal{X}^*$. Assumptions 1 and 2 ensure that even nonconvex problems still retain some geometric structure that allows for convergence analysis. In particular, the combination of these two assumptions provides both an upper and lower bound on a function's gradient, which will be useful in the forthcoming analysis.

Lemma 2 *For a function L that satisfies Assumptions 1 and 2, for all $x \in \mathbb{R}^n$ and $x^* \in \mathcal{X}^*$ we have $2\beta\left(L(x) - L^*\right) \leq \left|\nabla L(x)\right|^2 \leq K^2\left|x - x^*\right|^2$, where L^* is from (1), K is the Lipschitz constant of ∇L from Assumption 1, and β is the PL-constant of L from Assumption 2.*

Proof: The left inequality follows directly from Assumption 2. The right inequality follows by noting that $\nabla L(x^*) = 0$ and that therefore $\left|\nabla L(x)\right| = \left|\nabla L(x) - \nabla L(x^*)\right|$, and then applying the Lipschitz property of L from Assumption 1. □

We refer the reader to [19] for a thorough discussion of the PL condition in relation to other function properties. Among the strong convexity, essential strong convexity, weak strong convexity, restricted secant inequality, error bound, PL, and quadratic growth conditions, the authors of [19] establish that the PL and error bound conditions are the most general under which linear convergence to minimizers is achieved. In fact, given our Assumption 1, [19, Theorem 2] establishes that any function satisfying the strong convexity, essential strong convexity, weak strong convexity, restricted secant inequality, or error bound conditions also satisfies the PL condition. Thus, we enforce the PL condition as an assumption because it unifies a wide range of problems.

2.2 Mathematical Framework

We solve Problem 1 by developing a hybrid systems framework in which agents optimize with decentralized gradient descent in continuous time and communicate their iterates with each other in discrete time. Analogously to past research that has developed distributed versions of the discrete-time gradient descent law, our update law begins with the (centralized) first-order dynamical system

$$\dot{x} + \nabla L(x) = 0. \tag{2}$$

This is motivated by the use of gradient-based controllers in multi-agent systems, e.g., in consensus [25], as well as the simplicity of distributing gradient-based updates and the robustness to intermittency of communications that results from doing so [4].

We seek to distribute (2) across a team of agents in accordance with the parallelization requirement in Problem 1. We consider N agents indexed over $i \in [N]$ and divide $x \in \mathbb{R}^n$ into N blocks. Then agent i is responsible for updating and communicating values of the i-th block, $x_i \in \mathbb{R}^{n_i}$, where $n_i \in \mathbb{N}_{>0}$ and $\sum_{i\in[N]} n_i = n$. Thus, the variable x may be written as the vertical concatenation of all agents' blocks, i.e., $x = \mathrm{col}(x_1, x_2, \ldots, x_N)$. Each agent performs gradient descent on their own block but does not perform computations on any others.

Agents' updates occur in continuous time while communications of these updates occur in discrete time. These communication events are coordinated for all agents using a shared timer τ. When this timer reaches zero, agents will broadcast their current state x_i to all other agents. The timer will then be reset to a value within a specified interval $[\tau_{\min}, \tau_{\max}]$. Without loss of generality, we assume that communications are received at the same time as they are sent (allowing for communication delays requires only adding the length of delay onto the time between communications). When $\tau = 0$, state values are communicated by agent i for all $i \in [N]$ and received by agent ℓ for all $\ell \in [N]$, and then these communicated values are gathered by agent ℓ into a vector $\eta^\ell \in \mathbb{R}^n$, with the received value of x_i being assigned to η_i^ℓ.

Note that for two agents i and ℓ, the entries η_k^i and η_k^ℓ for some $k \in [N]$ may not be equal at initialization. They will be equal, however, after at least one communication event and will remain equal for the rest of the run of the algorithm. We consider the possibility of non-equal initial conditions for two reasons. First, it may be difficult to enforce equality of agents' initial conditions among many autonomous decision-makers that lack a central coordinator. For example, agents in a large decentralized network may choose their own initial conditions without coordinating with neighboring agents. Second, analysis of non-equal initial conditions allows us to establish global convergence results, where "global" indicates that convergence occurs regardless of the initial conditions. Globality allows our convergence results to apply to many more scenarios, such as those in which agents inadvertently have small disagreements about initial conditions. It also enables us to show that the hybrid system model we develop is robust to errors in the timing of agents' communications and computations by employing robust asymptotic stability tools for hybrid systems (see Section 5).

The value of η^i is used in agent i's continuous-time computations in an update-and-hold manner between communications. Formally, agent i executes

$$\dot{x}_i = -\nabla_i L(\eta^i), \tag{3}$$

where the gradient of the function L with respect to the i^{th} block and evaluated at some vector x is written as $\nabla_i L(x) = \frac{\partial}{\partial x_i} L(x)$. This sample-and-hold method is common in the literature [26, 27] and is used to demonstrate the feasibility of the hybrid approach in multi-agent optimization. The complete algorithm is summarized in Algorithm 1.

Algorithm 1: Distributed Gradient Descent

```
1  Initialization: set η^i ∈ ℝ^n, x_i = η_i^i ∈ ℝ^{n_i}, and τ ∈ [0, τ_max], for all i ∈ [N];
2  for i ∈ {1, ..., N} do
3      while τ ≥ 0 do
4          ẋ_i = −∇_i L(η^i);
5          τ̇ = −1;
6          if τ = 0 then
7              communicate x_i to all other agents: reset η_i^ℓ to x_i for all ℓ ∈ [N];
8              reset τ to a value in [τ_min, τ_max];
9          end
10     end
11 end
```

The ODE in line 4 does not need to be solved in closed form, in the sense of finding a function of time $t \to x_i(t)$ that obeys the ODE at all times. To implement line 4, for all $i \in [N]$, the i^{th} agent only needs to allow x_i to flow along the direction $-\nabla_i L(\eta^i)$ until the $\tau = 0$ condition is satisfied in line 6. Line 7 requires agent i to "communicate x_i to all other agents" and this step only requires agent i to communicate its value of x_i when the $\tau = 0$ condition is reached. Agent i does not need to communicate the time history of x_i, nor does it even need to store it. The next section develops the hybrid system model that will be used to analyze Algorithm 1.

3 Hybrid System Model

In this section, we define a hybrid system model that encompasses all agents' current states and their most recently communicated state values. Towards defining this "combined hybrid system", we first formally state what constitutes a hybrid system, then we define the timer that governs communications. This timer allows us to define the hybrid subsystems that are distributed across the agents. Building on that definition, we then present a definition of the combined hybrid system that will be the focus of our analysis, and we verify that this model meets the "hybrid basic conditions," which are defined below. Finally, we show the existence of solutions and conclude that all maximal solutions are complete.

3.1 Hybrid System Definitions

For the purposes of this paper, a hybrid system $\mathcal{H}$ has data (C, f, D, G) that takes the general form

$$\mathcal{H} = \begin{cases} \dot{x} = f(x) & x \in C \\ x^+ \in G(x) & x \in D \end{cases}, \tag{4}$$

where $x \in \mathbb{R}^n$ is the system's state, f defines the flow map and continuous dynamics for which C is the flow set, and G is the set-valued jump map which captures the system's discrete behavior for the jump set D. Here (and below) we use the standard notational convention in which x^+ denotes the value of the state x after it undergoes a jump. The meaning of (4) is that the state x evolves according to the ODE $\dot{x} = f(x)$, which defines the flow dynamics whenever $x \in C$, and the state x undergoes a jump whenever $x \in D$, which occurs instantaneously, leading to the difference inclusion $x^+ \in G(x)$. Note that while f is a single-valued map, G is a set-valued map, meaning that at jumps, x can take any value in $G(x)$.

More information on this definition and hybrid systems can be found in [15].

Definition 1 (Hybrid Basic Conditions, [15]) A hybrid system $\mathcal{H} = (C, f, D, G)$ with data given by (4) satisfies the *hybrid basic conditions* if

- C and D are closed subsets of $\mathbb{R}^n$;
- f is defined on C and is a continuous function from C to $\mathbb{R}^n$;
- $G : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is outer semicontinuous and locally bounded relative to D, and $D \subset \text{dom } G$.

If a hybrid system meets the hybrid basic conditions, then we say that the system is *well-posed* (Theorem 6.30, [15]). Well-posedness is desirable because it lets us establish the robustness of a hybrid system to perturbations, which we do in Section 5.

The following elementary example illustrates the formulation of a hybrid model.

Example 1 (From [12]) Consider the linear time-invariant system

$$\begin{aligned} \dot{z} &= Az + Bu \\ y &= Mz, \end{aligned} \tag{5}$$

where $z \in \mathbb{R}^n$ is the state, $y \in \mathbb{R}^q$ is the output, and $u \in \mathbb{R}^p$ is the input. The input $u : [0, \infty) \to \mathbb{R}^p$ is measurable and locally bounded, and $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{m \times n}$, and $M \in \mathbb{R}^{q \times n}$ are constant matrices. Consider the problem of designing an estimator for z that operates when only sporadic measurements of the output y are available; see [12]. Mathematically, the estimator only has access to outputs $y(t_k)$ for $k \in \mathbb{N}_{>0}$ for some collection of points in time $\{t_k\}_{k \in \mathbb{N}_{>0}}$.

We assume that the sequence $\{t_k\}_{k \in \mathbb{N}_{>0}}$ is strictly increasing and unbounded. We also assume that there are two constants $0 < T_1 \leq T_2$ such that

$$\begin{aligned} &0 \leq t_1 \leq T_2 \\ &T_1 \leq t_{k+1} - t_k \leq T_2 \text{ for all } k \in \mathbb{N}_{>0}. \end{aligned} \tag{6}$$

The value of T_1 is the minimum amount of time that elapses between t_k and t_{k+1} for any $k \in \mathbb{N}_{>0}$, and the value of T_2 is the corresponding maximum amount of time.

The state estimator generates an estimate $\hat{z} \in \mathbb{R}^n$. Given a solution $t \to z(t)$ to (5) obtained with input $t \to u(t)$ and resulting in the output $t \to y(t)$, measurements $y(t_k)$ occur at discrete time instances $\{t_k\}_{k \in \mathbb{N}_{>0}}$, and the state estimate $\hat{z}$ evolves in continuous time between measurements in order to mirror the continuous-time dynamics of (5). An estimate of $t \to z(t)$ is given by $t \to \hat{z}(t)$ satisfying

$$\begin{aligned} \dot{\hat{z}}(t) &= A\hat{z}(t) + Bu(t) && \text{for all } t \neq t_k, k \in \mathbb{N}_{>0} \\ \hat{z}(t^+) &= \hat{z}(t) + L(y(t) - M\hat{z}(t)) && \text{for all } t = t_k, k \in \mathbb{N}_{>0}, \end{aligned}$$

where $L \in \mathbb{R}^{n \times n}$ and $\hat{z}(t^+)$ is the right limit of $t \to \hat{z}(t)$ at $t = t_k$. When $t \neq t_k$, the state estimate flows in continuous time according to the system dynamics in (5). When $t = t_k$, the state estimate jumps instantaneously from the value $z(t_k)$ to the value $z(t_k^+)$, which is a correction term that accounts for any differences between the measured output $y(t_k)$ and the predited output $M\hat{z}(t_k)$. As is common in state estimation problems, we examine the estimation error $t \to \omega(t) = z(t) - \hat{z}(t)$, where

$$\begin{aligned} \dot{\omega}(t) &= A\omega(t) && \text{for all } t \neq t_k, k \in \mathbb{N}_{>0} \\ \omega(t^+) &= (I - LM)\omega(t) && \text{for all } t = t_k, k \in \mathbb{N}_{>0}. \end{aligned}$$

We can formulate the ω dynamics as a hybrid system. The measurement of outputs is driven by the sequence $\{t_k\}_{k \in \mathbb{N}_{>0}}$, and we reformulate it to be state-driven to capture all possible sequences satisfying (6). We introduce a new state τ for this purpose, and τ evolves as follows. Between jumps, the value of τ counts down with unit rate. When $\tau = 0$, a jump is triggered and τ is reset to a value in $[T_1, T_2]$ before it starts counting down again. The joint dynamics of ω and τ can be written as the hybrid system $\mathcal{H}_\omega$ with dynamics

$$\mathcal{H}_\omega \begin{cases} \left.\begin{aligned} \dot{\omega} &= A\omega \\ \dot{\tau} &= -1 \end{aligned}\right\} (\omega, \tau) \in C \\ \left.\begin{aligned} \omega^+ &= (I - LM)\omega \\ \tau^+ &\in [T_1, T_2] \end{aligned}\right\} (\omega, \tau) \in D \end{cases},$$

where the flow set C and jump set D are defined as

$$\begin{aligned} C &= \{(\omega, \tau) \in \mathbb{R}^n \times \mathbb{R}_{\geq 0} : \tau \in [0, T_2]\} \\ D &= \{(\omega, \tau) \in \mathbb{R}^n \times \mathbb{R}_{\geq 0} : \tau = 0\}. \end{aligned}$$

The sets C and D are not disjoint in this model and, in particular, both C and D contain points of the form $(\omega, 0)$. This property conveys the fact that the timer τ counts down from a positive number until it reaches zero exactly and only then is a jump triggered.

A complete hybrid model requires the definition of a flow map f over C and a jump map G over D. We define a new state vector $x := (\omega^T, \tau)^T$. Then the flow and jump maps are

$$f(x) := \begin{pmatrix} A\omega \\ -1 \end{pmatrix} \qquad \text{for all } x \in C$$

and

$$G(x) := \begin{pmatrix} (I - LM)\omega \\ [T_1, T_2] \end{pmatrix} \qquad \text{for all } x \in D,$$

respectively.

The dynamics of the estimator are entirely encapsulated in the hybrid model $\mathcal{H}_\omega = (C, f, D, G)$, and the formulation of this hybrid model enables the use of a large collection of theoretical tools for its analysis [15, 31]. ◊

Hybrid systems are often analyzed for similar properties as non-hybrid systems, such as asymptotic stability. They can also exhibit several types of undesirable behavior. For example, hybrid systems can exhibit the Zeno phenomenon in which they undergo an infinite number of jumps in finite time. They can also exhibit finite escape time in which some states go to infinity in finite ordinary (flow) time. Both Zeno behavior and finite escape time cause solutions to have bounded domains. In Sections 3.2 through 3.4 we formally define the hybrid system model of Algorithm 1, and we show in Section 3.5 that it exhibits neither Zeno behavior nor finite escape time. Then in Section 4 we prove that the states of the hybrid model asymptotically converge to a minimizer, and we derive a convergence rate.

3.2 Mechanism Governing the Communication Events

We seek to account for intermittent communication events that occur only at some time instances t_j, for $j \in \mathbb{N}_{>0}$, that are not known *a priori*. We assume that the sequence $\{t_j\}_{j=1}^{\infty}$ is strictly increasing and unbounded. Between consecutive time events, some amount of time elapses which we upper and lower bound with positive scalars $\tau_{\min}$ and $\tau_{\max}$:

$$0 < \tau_{\min} \leq t_{j+1} - t_j \leq \tau_{\max} \quad \text{for all } j \in \mathbb{N}_{>0}. \tag{7}$$

The upper bound $\tau_{\max}$ prevents infinitely long communication delays and ensures convergence, while the lower bound $\tau_{\min}$ rules out Zeno behavior.

To generate communication events at times t_j satisfying (7), let τ be the timer that governs when agents exchange data, where τ is defined by

$$\dot{\tau} = -1 \qquad \tau \in [0, \tau_{\max}], \tag{8}$$

$$\tau^+ \in [\tau_{\min}, \tau_{\max}] \qquad \tau = 0, \tag{9}$$

for $\tau_{\min}, \tau_{\max} \in \mathbb{R}_{>0}$, where (8) specifies the flow behavior of τ and (9) specifies the jump behavior of τ. In words, the timer τ flows with $\dot{\tau} = -1$ until it reaches $\tau = 0$.

At that point, it stops flowing and a jump is triggered, which resets τ to a value within $[\tau_{\min}, \tau_{\max}]$, and this new value is denoted by τ^+. Since $\tau^+ \geq \tau_{\min} > 0$, the timer τ resumes flowing and this process repeats indefinitely. There is indeterminacy built into the timer in that the reset map is only confined to a compact interval, $[\tau_{\min}, \tau_{\max}]$, where $\tau_{\min}$ and $\tau_{\max}$ are both positive real numbers.

3.3 Hybrid Subsystems

Recall that for all $i \in [N]$ agent i stores their own state variable $x_i \in \mathbb{R}^{n_i}$. Communications received from all other agents are stored in $\eta^i \in \mathbb{R}^n$, including agent i's state at the most recent communication event. We define the state of agent i's hybrid system as $\xi^i = (x_i, \eta^i, \tau)$, where x_i is the state of agent i's block of the decision variable x (the one it is responsible for updating), η^i is the vector of state values communicated to agent i at communication events, and τ is defined as in (8) and (9). Applying the dynamics given in (3), this setup leads to the hybrid subsystem

$$\begin{aligned} \dot{\xi}^i &= \begin{bmatrix} -\nabla_i L(\eta^i) \\ \mathbf{0}_n \\ -1 \end{bmatrix} && \xi^i \in \mathbb{R}^{n_i} \times \mathbb{R}^n \times [0, \tau_{\max}] \\ \xi^{i+} &\in \begin{bmatrix} x_i \\ x \\ [\tau_{\min}, \tau_{\max}] \end{bmatrix} && \xi^i \in \mathbb{R}^{n_i} \times \mathbb{R}^n \times \{0\}. \end{aligned}$$

The flow and jump sets for ξ^i have non-empty intersection because they incorporate the τ dynamics from (8) and (9) above. This property ensures that the timer τ decreases all the way to 0 before it resets. Subsystem i has $\dot{\eta}^i = 0$ for all $i \in [N]$ to model agents' sample-and-hold strategy. Agent i updates only the value of x_i using the memorized values in η^i, which is why it has $\dot{x}_i = -\nabla_i L(\eta^i)$ during flows. Agent i does not update x_j for any $j \neq i$, and the value of x_j that agent i has access to only changes when agent j communicates it to agent i. The value of x_j onboard agent i is denoted η^i_j, and agent i has the dynamics $\dot{\eta}^i_j = 0$ to model the fact that agent i does not change this value. Agent i does receive the value of x_j during jumps, which is modeled by having $\eta^{i+} = x$ in the jump map. Agent i has $\dot{\eta}^i_i = 0$ because its computations are modeled by the dynamics of the state x_i, and we attain a simpler hybrid model by separating states that change during communications from those that change during computations.

3.4 Combined Hybrid System

We are now ready to form the combined hybrid system for analysis. First, we combine all η^i values into a single variable $\eta := \operatorname{col}(\eta^1, \eta^2, \ldots, \eta^N)$, which is in $\mathbb{R}^{nN}$. We define the state of the combined hybrid system as $\xi = (x, \eta, \tau) \in \mathcal{X}$, where $\mathcal{X} := \mathbb{R}^n \times \mathbb{R}^{nN} \times \mathcal{T}$ and $\mathcal{T} := [0, \tau_{\max}]$. To simplify notation, let the functions $h_i : \mathbb{R}^n \to \mathbb{R}^{n_i}$ be given by $h_i(\eta^i) = \nabla_i L(\eta^i)$ for all $i \in [N]$. We collect these together into the function $h : \mathbb{R}^{nN} \to \mathbb{R}^n$, given by

$$h(\eta) := \operatorname{col}\big(h_1(\eta^1), \ldots, h_N(\eta^N)\big). \tag{10}$$

This definition leads to the combined hybrid system $\mathcal{H} = (C, f, D, G)$ with

$$\dot{\xi} = \begin{bmatrix} -h(\eta) \\ \mathbf{0}_{nN} \\ -1 \end{bmatrix} =: f(\xi) \tag{11}$$

for every $\xi \in C := \mathcal{X}$. Similar to the hybrid subsystems, when $\tau = 0$, all agents undergo a jump. When a jump occurs, x remains constant, η^i is mapped with $\eta^{i,+} = x$ for all $i \in [N]$, and $\tau^+ \in [\tau_{\min}, \tau_{\max}]$. Formally, for each $\xi \in D := \{\xi \in \mathcal{X} : \tau = 0\}$, we define the jump map G as

$$\xi^+ \in \begin{bmatrix} x \\ \text{col}(x, \dots, x) \\ [\tau_{\min}, \tau_{\max}] \end{bmatrix} =: G(\xi), \text{ dom } G = \mathbb{R}^n \times \mathbb{R}^{nN} \times \mathbb{R}. \tag{12}$$

The connections between (11)-(12) and Problem 1 are as follows. The block coordinate descent law that we use only has agent i perform computations on x_i and (3) shows this for agent i. The first entry of f in (11) is $-h(\eta)$, and this entry replicates the dynamics in (3) for all $i \in [N]$. This entry of f therefore encodes that, for all $i \in [N]$, agent i updates the value of x_i by moving it along a negative gradient flow. Agent i stores a full copy of the decision variable x onboard itself in the state η^i. Agent i does not update x_j for any $j \neq i$ during its computations, and agent i therefore has $\dot{\eta}^i_j = 0$ for all $j \in [N]\backslash\{i\}$. It also has $\dot{\eta}^i_i = 0$ because η^i only stores values that have been communicated, and the value of x_i is copied over to η^i_i only when x_i is communicated to other agents. These zero derivatives are encoded for all agents in the second entry of f in (11), which is a zero vector of the appropriate size. The third entry of f in (11) is equal to -1 to encode the fact that the timer τ counts down with unit rate until it reaches zero.

The jump map G in (12) is triggered when $\tau = 0$ is reached. The first entry in G is x, which shows that jumps do not change the decision variables that agents compute; the state x is simply set equal to its current value, which encodes no change. The second entry in G encodes the fact that agents communicate by broadcasting their states when $\tau = 0$. Mathematically, for all $i \in [N]$, agent i sets $\eta^i = x$ and this step performs two distinct operations. First, it sets $\eta^i_i = x_i$, where x_i is the value of agent i's decision variables when $\tau = 0$ was reached. This step ensures that this value of x_i will be used in agent i's computations that will occur after the jump. Second, this step sets $\eta^i_j = x_j$ for $j \in [N]\backslash\{i\}$, which models agent j communicating to agent i the value of x_j from the time at which $\tau = 0$ was reached. These new entries of η^i will also be used in agent i's updates after the jump is completed. The third entry in G simply resets the value of τ to some value in the interval $[\tau_{\min}, \tau_{\max}]$, and this reset deliberately allows for some non-determinism in the value that τ is reset to.

This hybrid system models the information that drives all agents' computations and communications, and our forthcoming analyses will derive the properties of this model, including convergence to minimizers.

3.5 Hybrid Basic Conditions

We now demonstrate that $\mathcal{H}$ meets the hybrid basic conditions and is well-posed.

Lemma 3 *Let L satisfy Assumption 1. Then the hybrid system given by $\mathcal{H} = (C, f, D, G)$ and whose data is defined in* (11) *and* (12) *satisfies the hybrid basic conditions from Definition 1 and is nominally well-posed as a result.*

Proof: The sets C and D are closed subsets of $\mathbb{R}^n \times \mathbb{R}^{nN} \times \mathbb{R}$ by definition. Due to our assumption that ∇L is continuous, f is a continuous function from C to $\mathbb{R}^n \times \mathbb{R}^{nN} \times \mathbb{R}$. By construction, G is outer semicontinuous and locally bounded relative to D. Finally, $D \subset \text{dom } G$ because $\text{dom } G = \mathbb{R}^n \times \mathbb{R}^{nN} \times \mathbb{R}$ from (12). □

3.6 Existence of Solutions

We denote solutions to $\mathcal{H}$ by ϕ, which we parameterize by $(t, j) \in \mathbb{R}_{\geq 0} \times \mathbb{N}$, where t denotes the ordinary (continuous) time, and j is a natural number that denotes the jump (discrete) time. Here, j is the cumulative number of jumps the agents have performed. Per Definition 2.3 in [15], $\text{dom } \phi \subset \mathbb{R}_{\geq 0} \times \mathbb{N}$ is a *hybrid time domain* if for all $(T, J) \in \text{dom } \phi$, the set $\text{dom } \phi \cup ([0, T] \times \{0, 1, \ldots, J\})$ can be written as $\bigcup_{j=0}^{J-1}([t_j, t_{j+1}], j)$ for some finite sequence of times $0 = t_0 \leq t_1 \leq \cdots \leq t_J$. We say that a solution ϕ is *complete* if $\text{dom } \phi$ is unbounded and we denote the set of all maximal solutions to $\mathcal{H}$ as $\mathcal{S}_{\mathcal{H}}$. A solution ϕ to $\mathcal{H}$ is called maximal if it cannot be extended further. In addition to being well-posed, there exists a nontrivial solution to $\mathcal{H}$. Below, in Proposition 2, we will show that all maximal solutions are complete. Toward doing so, we have the following lemma.

Lemma 4 (Completeness of Solutions) *Let Assumption 1 hold and consider the hybrid system $\mathcal{H}$ defined in* (11) *and* (12). *Let $\tau_{\min}$ and $\tau_{\max}$ be such that $0 < \tau_{\min} \leq \tau_{\max}$. Then there exists a nontrivial solution to $\mathcal{H} = (C, f, D, G)$ from every initial point in $C \cup D$. Additionally, every maximal solution ϕ to the hybrid system $\mathcal{H}$ is non-Zeno and complete.*

Proof: See Appendix A.1. □

The next section also analyzes the stability of $\mathcal{H}$, and as a preliminary result we have the following lemma on how system trajectories evolve during flow intervals.

Lemma 5 *Consider the hybrid system $\mathcal{H} = (C, f, D, G)$ with data given in* (11) *and* (12). *Pick a solution $\phi = (\phi_x, \phi_\eta, \phi_\tau)$ to $\mathcal{H}$. For each $I^j := \{t : (t, j) \in \text{dom } \phi\}$ with nonempty interior and with $t_{j+1} > t_j$ such that $[t_j, t_{j+1}] = I^j$, we have*

$$\phi_{\eta^i}(t, j) = \phi_{\eta^i}(t_j, j) \tag{13}$$

$$\phi_{x_i}(t, j) = \begin{cases} \phi_{\eta_i^i}(t_j, j) - (t - t_j)\nabla_i L\big(\phi_{\eta^i}(t_j, j)\big) & j \geq 1 \\ \phi_{x_i}(0, 0) - t\nabla_i L\big(\phi_{\eta^i}(0, 0)\big) & j = 0 \end{cases} \tag{14}$$

for all $t \in (t_j, t_{j+1})$, where t_j denotes the continuous time at which the most recent jump j was performed.

Proof: Given $t \in (t_j, t_{j+1})$, the solution $\phi = (\phi_x, \phi_\eta, \phi_\tau)$ has flowed some distance given by $(t - t_j)\dot{\phi}$, where $\dot{\phi}$ is constant due to the sample-and-hold methodology. Applying our definition of f in (11) gives (13) and (14). □

4 Stability Analysis

In this section, we define the convergence set $\mathcal{A}$ and propose a Lyapunov function in Lemma 7. As an interim result, we show that if all agents initialize with the same state values, then their total distance from $\mathcal{A}$ is monotonically decreasing for all objective functions that satisfy Assumptions 1 and 2. Next, Proposition 2 expands this interim result and bounds the distance from $\mathcal{A}$ for all hybrid time (t, j) where $j \geq 1$, regardless of agents' initialization. Finally, Theorem 1 removes the condition $j \geq 1$ and establishes global exponential stability.

4.1 Convergence Set

By Assumption 2, the set of minimizers $\mathcal{X}^*$ is non-empty and may contain more than one element. Following from the properties of L, namely Assumption 2, the algorithm has converged to a minimizer $x^* \in \mathcal{X}^*$ of L if and only if it has reached a stationary point, i.e., a point at which ∇L is zero. Given a complete solution $\phi = (\phi_x, \phi_\eta, \phi_\tau)$ to the hybrid system $\mathcal{H}$, we seek to ensure that $\lim_{t+j\to\infty} \nabla L\big(\phi_x(t,j)\big) = \mathbf{0}_n$ and $\lim_{t+j\to\infty} \nabla L\big(\phi_{\eta^i}(t,j)\big) = \mathbf{0}_n$ for all $i \in [N]$. This is equivalent to a set stability problem where the convergence set for the hybrid system $\mathcal{H}$ is given by

$$\begin{aligned}\mathcal{A} &:= \{\xi = (x, \eta, \tau) \in \mathcal{X} : \nabla L(x) = \mathbf{0}_n, \nabla L(\eta^i) = \mathbf{0}_n, \tau \in [0, \tau_{\max}], \text{for all } i \in [N]\} \\ &= \mathcal{X}^* \times \left(\mathcal{X}^*\right)^N \times [0, \tau_{\max}]. \end{aligned} \tag{15}$$

Given a vector $\xi = (x, \eta, \tau) \in \mathcal{X}$, let $\hat{x}^0$ be the closest element of $\mathcal{X}^*$ to x, and let $\hat{x}^i$ be the closest element of $\mathcal{X}^*$ to η^i for each $i \in [N]$. Formally, given $\xi = (x, \eta, \tau)$, the points $\hat{x}^0$ and $\hat{x}^i$ are defined as

$$\hat{x}^0 := \underset{x^* \in \mathcal{X}^*}{\arg\min} \big|x - x^*\big| \qquad \text{and} \qquad \hat{x}^i := \underset{x^* \in \mathcal{X}^*}{\arg\min} \big|\eta^i - x^*\big| \text{ for all } i \in [N].$$

Using these definitions, the squared distance from ξ to $\mathcal{A}$ is given by $\big|\xi\big|_{\mathcal{A}}^2 := \big|x - \hat{x}^0\big|^2 + \sum_{i\in[N]} \big|\eta^i - \hat{x}^i\big|^2$.

For all $\xi = (x, \eta, \tau) \in \mathcal{A}$, the definition of $\mathcal{A}$ does not immediately imply that $x, \eta^1, \ldots, \eta^N$ all converge to the same point $x^* \in \mathcal{X}^*$. However, when combined with the dynamics of our hybrid system $\mathcal{H}$, this convergence property is guaranteed.

Lemma 6 *Consider the hybrid system $\mathcal{H}$ defined in* (11) *and* (12)*. Let $\mathcal{A}$ be as defined in* (15)*. For each maximal solution ϕ to $\mathcal{H}$, if $\phi(t,j) \in \mathcal{A}$ for $(t,j) \in \operatorname{dom} \phi$ with $t \geq \tau_{\max}$, then $\phi_x(t,j) = \phi_{\eta^1}(t,j) = \cdots = \phi_{\eta^N}(t,j) \in \mathcal{X}^*$.*

Proof: Following the definition of $\mathcal{A}$, the condition $\phi(t,j) \in \mathcal{A}$ implies both that $\nabla L\big(\phi_x(t,j)\big) = \mathbf{0}_n$ and $\nabla L\big(\phi_{\eta^i}(t,j)\big) = \mathbf{0}_n$ for all i. Note that because $t \geq \tau_{\max}$, agents have performed at least one jump. Now, for the sake of contradiction, suppose that $\phi_{\eta^i}(t,j) \neq \phi_x(t,j)$ for at least one $i \in [N]$. Then there exists at least one entry ℓ of $\phi_x(t,j)$ such that $\phi_{x_\ell}(t,j) \neq \phi_{\eta^i_\ell}(t,j)$. Because agents have performed at least one jump, it holds that $\phi_{\eta^i_\ell}(t,j) = \phi_{\eta^\ell_\ell}(t,j)$. Combining this equality with (13) and (14) provides the relationship $\phi_{x_\ell}(t,j) = \phi_{\eta^i_\ell}(t,j) - (t - t_j)\nabla_\ell L\big(\phi_{\eta^\ell}(t,j)\big)$. To satisfy the condition $\phi_{x_\ell}(t,j) \neq \phi_{\eta^i_\ell}(t,j)$, it is necessary that $\nabla_\ell L\big(\phi_{\eta^\ell}(t,j)\big) \neq 0$, which

contradicts the hypothesis that $\phi_{\eta^\ell}(t,j) \in \mathcal{X}^*$. Then $\phi_{\eta^i}(t,j) = \phi_x(t,j) \in \mathcal{X}^*$ for all $i \in [N]$ and $t \geq \tau_{\max}$. □

4.2 Bounds on the Lyapunov Function

Central to proving our main result is a Lyapunov function that is bounded above and below by $\mathcal{K}_\infty$ comparison functions α_1 and α_2, given next.

Lemma 7 *Let Assumptions 1 and 2 hold. Let $V : \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a Lyapunov function candidate for the hybrid system $\mathcal{H} = (C, f, D, G)$ defined in (11) and (12), given by*

$$V(\xi) = \big(L(x) - L^*\big) + \sum_{i \in [N]} \big(L(\eta^i) - L^*\big),$$

for all $\xi = (x, \eta, \tau) \in \mathcal{X}$, where L is the objective function and L^* is from (1). Then, there exist $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ such that $\alpha_1(|\xi|_{\mathcal{A}}) \leq V(\xi) \leq \alpha_2(|\xi|_{\mathcal{A}})$ for all $\xi \in C \cup D \cup G(D)$. In particular, for all $s \geq 0$, α_1 and α_2 are given by

$$\alpha_1(s) := \frac{\beta}{2} s^2 \quad \text{and} \quad \alpha_2(s) := \frac{K}{2} s^2.$$

Proof: See Appendix A.2. □

4.3 Global Exponential Stability

We first bound the distance to a minimizer of L over time for a class of initial conditions in Proposition 1. This result is then expanded to include all possible solutions and initial conditions in Proposition 2, which characterizes the convergence of trajectories after the first jump. Then, Theorem 1 extends Proposition 2 to all times. We first consider the case where agents all agree at initialization and $\phi_x(0,0) = \phi_{\eta^i}(0,0)$ for all $i \in [N]$.

Proposition 1 *Let Assumptions 1 and 2 hold and consider the hybrid system $\mathcal{H}$ defined in (11) and (12). Let $\mathcal{A}$ be as defined in (15) and let $\tau_{\min}$ and $\tau_{\max}$ be such that $0 < \tau_{\min} \leq \tau_{\max} < \frac{1}{K}$, where K is the Lipschitz constant of ∇L from Assumption 1. Consider a maximal solution ϕ to $\mathcal{H}$ such that $\phi_x(0,0) = \phi_{\eta^i}(0,0)$ for all i in $[N]$. Then, for all $(t,j) \in \operatorname{dom} \phi$, the following is satisfied:*

$$\big|\phi(t,j)\big|_{\mathcal{A}} \leq \sqrt{\frac{K}{\beta}} \exp\Big(-\frac{\beta}{N+1}(1 - K\tau_{\max})t\Big)\big|\phi(0,0)\big|_{\mathcal{A}},$$

where β is the PL constant of L from Assumption 2 and $1 - K\tau_{\max} > 0$ from the upper bound on $\tau_{\max}$.

Proof: See Appendix A.2. □

Proposition 1 shows that solutions to $\mathcal{H}$ converge exponentially fast to a minimizer, and the exponent agrees (up to constants) with the convergence rate for centralized continuous-time gradient descent on PL functions [40, Example 1]. It also agrees with the convergence rate for discrete-time multi-agent minimization of PL functions [41, Theorem 1] (again, up to constants).

Regarding the coefficient in Proposition 1, the constant $\sqrt{\frac{K}{\beta}}$ is tight in the sense that the bound holds with equality at some times for some functions. For example, consider $L(x) = \frac{1}{2}\|x\|^2$, which satisfies Assumption 1 with constant $K = 1$ and satisfies Assumption 2 with constant $\beta = 1$. For this choice of L, at hybrid time $(t, j) = (0, 0)$ the bound in Proposition 1 takes the form

$$\big|\phi(0,0)\big|_{\mathcal{A}} \leq \big|\phi(0,0)\big|_{\mathcal{A}},$$

which shows that the constant $\sqrt{\frac{K}{\beta}}$ cannot be made smaller.

The exponential term in Proposition 1 contains the constant $-\frac{\beta}{N+1}(1 - K\tau_{\max})$, which differs from the constant $-\beta$ that appears in the convergence rate of centralized continuous-time gradient descent [39, Example 1]. If one used our methods to analyze a centralized setup, then the state ξ would only need to contain η^1 and there would be no need for ξ to contain x nor any η^i for $i \geq 2$ because there would be no communications and no other agents to disagree with. Then one could use the Lyapunov function $V(\xi) = L(\eta^1) - L(x^*)$, and re-tracing the steps of Proposition 1 with this choice of V would eliminate the factor of $\frac{1}{N+1}$. Then the constant in the exponential term would take the form $-\beta(1 - K\tau_{\max})$, and the value of $\tau_{\max}$ could be chosen arbitrarily small to drive this constant arbitrarily close to $-\beta$. Through these changes, Proposition 1 comes arbitrarily close to recovering the convergence rate for centralized continuous-time gradient descent with PL functions.

In practice, this preliminary result is useful when agreeing on initial values is easy to implement. However, we wish to account for all possible initialization scenarios. When agents disagree on initial conditions, their computations are not guaranteed to decrease the distance to minimizers until after the first jump. The next result establishes an exponential upper bound from each initial condition, including those at which agents disagree, and it accounts for the possible increase in the distance to the set of minimizers that can occur before the first jump. It results in a larger bound than the one in Proposition 1, as the following result presents.

Proposition 2 (Exponential bound for $j \geq 1$) *Let Assumptions 1 and 2 hold and consider the hybrid system $\mathcal{H}$ defined in* (11) *and* (12)*. Let $\mathcal{A}$ be as defined in* (15) *and choose $\tau_{\min}$ and $\tau_{\max}$ such that $0 < \tau_{\min} \leq \tau_{\max} < \frac{1}{K}$, where K is the Lipschitz constant of ∇L from Assumption 1. For each maximal solution ϕ and for all $(t, j) \in \operatorname{dom} \phi$ such that $j \geq 1$, the following is satisfied:*

$$\big|\phi(t,j)\big|_{\mathcal{A}} \leq \sqrt{\frac{2K(N+1)}{\beta}} \exp\Big(-\frac{\beta}{N+1}(1 - K\tau_{\max})t\Big)\big|\phi(0,0)\big|_{\mathcal{A}},$$

where β is the PL constant of L from Assumption 2 and $1 - K\tau_{\max} > 0$ from the upper bound on $\tau_{\max}$.

Proof: See Appendix A.2. □

Of course, for global exponential stability we must show the exponential convergence to minimizers of all trajectories from all initial conditions for all times, not only for $j \geq 1$. Accordingly, the following theorem does so and gives our main result on global exponential stability.

Theorem 1 *Let Assumptions 1 and 2 hold and consider the hybrid system $\mathcal{H}$ defined in* (11) *and* (12)*. Let $\mathcal{A}$ be as defined in* (15) *and choose $\tau_{\min}$ and $\tau_{\max}$ such that $0 < \tau_{\min} \leq \tau_{\max} < \frac{1}{K}$, where K is the Lipschitz constant of ∇L from Assumption 1. Then, the set $\mathcal{A}$ is globally exponentially stable for $\mathcal{H}$ defined in* (11)-(12)*, namely, for each maximal solution ϕ to $\mathcal{H}$, for all $(t, j) \in \operatorname{dom} \phi$, we have*

$$\big|\phi(t,j)\big|_{\mathcal{A}} \leq \max\left\{\frac{\sqrt{2}}{\exp(-\rho\tau_{\max})}, \frac{\sqrt{1+2K^2\tau_{\max}^2}}{\exp(-\rho\tau_{\max})}, \sqrt{\frac{2K(N+1)}{\beta}}\right\}\exp(-\rho t)\big|\phi(0,0)\big|,$$

where β is the PL constant of L from Assumption 2, we have $1 - K\tau_{\max} > 0$ from the upper bound on $\tau_{\max}$, and $\rho = \frac{\beta}{N+1}(1 - K\tau_{\max})$.

Proof: By Lemma 4 we know that any maximal solution ϕ is also complete. Consider any t such that $(t, 0) \in \operatorname{dom} \phi$. We first seek to bound $\big|\phi(t,0)\big|_{\mathcal{A}}$ with some constant. Define $\bar{x}^0 := \arg\min_{x^* \in \mathcal{X}^*} \big|\phi_x(t,0) - x^*\big|$. Note that $\phi_{\eta^i}(t,0) = \phi_{\eta^i}(0,0)$ for all $i \in [N]$ and define $\hat{x}^i := \arg\min_{x^* \in \mathcal{X}^*} \big|\phi_{\eta^i}(t,0) - x^*\big| = \arg\min_{x^* \in \mathcal{X}^*} \big|\phi_{\eta^i}(0,0) - x^*\big|$ for all $i \in [N]$. We begin by expanding $\big|\phi(t,0)\big|_{\mathcal{A}}^2$, where

$$\big|\phi(t,0)\big|_{\mathcal{A}}^2 = \big|\phi_x(t,0) - \bar{x}^0\big|^2 + \sum_{i\in[N]} \big|\phi_{\eta^i}(t,0) - \hat{x}^i\big|^2. \tag{16}$$

We now define $\hat{x}^0 := \arg\min_{x^* \in \mathcal{X}^*} \big|\phi_x(0,0) - x^*\big|$. Note that by definition of $\bar{x}^0$, $\big|\phi_x(t,0) - \bar{x}^0\big|^2 \leq \big|\phi_x(t,0) - \hat{x}^0\big|^2$. Along with $\phi_{\eta^i}(t,0) = \phi_{\eta^i}(0,0)$ for all $i \in [N]$, this allows us to rewrite (16) as

$$\big|\phi(t,0)\big|_{\mathcal{A}}^2 \leq \big|\phi_x(t,0) - \hat{x}^0\big|^2 + \sum_{i\in[N]} \big|\phi_{\eta^i}(0,0) - \hat{x}^i\big|^2. \tag{17}$$

We first upper bound $\big|\phi_{x_i}(t,0) - \hat{x}_i^0\big|^2$ by applying Lemma 5 and using $\big|a-b\big|^2 \leq 2\big|a\big|^2 + 2\big|b\big|^2$, resulting in

$$\begin{aligned}
\big|\phi_{x_i}(t,0) - \hat{x}_i^0\big|^2 &= \big|\phi_{x_i}(0,0) - t\nabla_i L\big(\phi_{\eta^i}(0,0)\big) - \hat{x}_i^0\big|^2 \\
&\leq 2\big|\phi_{x_i}(0,0) - \hat{x}_i^0\big|^2 + 2t^2\big|\nabla_i L\big(\phi_{\eta^i}(0,0)\big) - \nabla_i L\big(\hat{x}^i\big)\big|^2 \\
&\leq 2\big|\phi_{x_i}(0,0) - \hat{x}_i^0\big|^2 + 2K^2\tau_{\max}^2\big|\phi_{\eta^i}(0,0) - \hat{x}^i\big|^2,
\end{aligned} \tag{18}$$

where the first equality applies Lemma 5, the first inequality uses $\nabla L(\hat{x}^i) = 0$, and the final inequality applies Lemma 8. Summing over all i on both sides of (18) gives

$$\big|\phi_x(t,0) - \hat{x}^0\big|^2 \leq 2\big|\phi_x(0,0) - \hat{x}^0\big|^2 + 2K^2\tau_{\max}^2 \sum_{i\in[N]} \big|\phi_{\eta^i}(0,0) - \hat{x}^i\big|^2.$$

Applying this inequality to (17) gives

$$\begin{aligned}
\big|\phi(t,0)\big|_{\mathcal{A}}^2 &\leq 2\big|\phi_x(0,0) - \hat{x}^0\big|^2 + \big(1 + 2K^2\tau_{\max}^2\big) \sum_{i\in[N]} \big|\phi_{\eta^i}(0,0) - \hat{x}^i\big|^2 \\
&\leq \max\{2, 1 + 2K^2\tau_{\max}^2\}\big|\phi(0,0)\big|_{\mathcal{A}}^2.
\end{aligned}$$

Taking the square root and combining with Proposition 2 gives the final result.

5 Robustness to Timing Errors

In this section, we show that the hybrid system $\mathcal{H}$ is robust to a class of model errors, in particular that it is robust to errors in the dynamics of the timer. By "robust" we mean that there exists a maximum nonzero perturbation level such that all solutions under such perturbations converge to a neighborhood of the set $\mathcal{A}$, where the size of the neighborhood depends on the size of the perturbation. Formally, we consider perturbed timer dynamics of the form

$$\dot{\tau}_p = -1 + \kappa, \quad \tau_p^+ \in [\tau_{\min} + \theta_{\min}, \tau_{\max} + \theta_{\max}], \tag{19}$$

where τ_p denotes the perturbed timer, $\kappa \in (-\infty, 1)$ is a constant that models skew on the timer dynamics, and the terms $\theta_{\min} \in \mathbb{R}$ and $\theta_{\max} \in \mathbb{R}$ are perturbations to $\tau_{\min}$ and $\tau_{\max}$, respectively, that satisfy

$$0 < \tau_{\min} + \theta_{\min} \leq \tau_{\max} + \theta_{\max}. \tag{20}$$

The full perturbation of the hybrid system model in (11)-(12) has state vector denoted $\xi_p = (x, \eta, \tau_p)$ whose flow dynamics are given by

$$\dot{\xi}_p = \begin{bmatrix} -h(\eta) \\ \mathbf{0}_{nN} \\ -1 + \kappa \end{bmatrix} =: f_p(\xi_p), \qquad \xi_p \in C_p := \mathbb{R}^n \times \mathbb{R}^{nN} \times [0, \tau_{\max} + \theta_{\max}],$$

where h and η are from (11), and κ and $\theta_{\max}$ are from (19). Its jump dynamics are given by

$$\xi_p^+ = \begin{bmatrix} x \\ \operatorname{col}(x, \dots, x) \\ \left[\tau_{\min} + \theta_{\min}, \tau_{\max} + \theta_{\max}\right] \end{bmatrix} =: G_p(\xi_p), \qquad \xi_p \in D_p := \{\xi_p \in \mathcal{X}_p : \tau_p = 0\}.$$

We use

$$\mathcal{H}_p := (C_p, f_p, D_p, G_p) \tag{21}$$

to denote the full hybrid system with the perturbed timer dynamics.

To enable the desired robustness result, we first require the following assumption.

Assumption 3 *The set of optimizers $\mathcal{X}^*$ is compact.*

Assumption 3 is required here so that agents' computations do not converge to a solution that is arbitrarily far away from their current iterates. It is known to hold under mild conditions, such as the condition that the objective function L is coercive [3, Proposition 2.1.1]. We have the following robustness result.

Theorem 2 *Let Assumptions 1-3 hold. Then, for the hybrid system $\mathcal{H}_p$ from* (21)*, there exists a function $\beta \in \mathcal{KL}$ such that for every $\epsilon > 0$ there exists $\rho^* \in (0, \infty)$ such that if $\max\left\{|\kappa|, |\theta_{\min}|, |\theta_{\max}|\right\} \leq \rho^*$, then each solution ϕ_p to $\mathcal{H}_p$ satisfies*

$$|\phi_p(t,j)|_{\mathcal{A}} \leq \beta\big(|\phi_p(0,0)|_{\mathcal{A}}, t + j\big) + \epsilon$$

for all $(t, j) \in \operatorname{dom} \phi$, where $\mathcal{A}$ is given in (15)*.*

Proof: By Lemma 3, the nominal hybrid system $\mathcal{H}$ satisfies the hybrid basic conditions. By Assumption 3 the set $\mathcal{X}^*$ is compact and thus the set $\mathcal{A}$ is compact. The function $x \mapsto |x|_{\mathcal{A}}$ is a proper indicator for the set $\mathcal{A}$ viewed as a subset of $\mathbb{R}^n$. That is, we have both $|x|_{\mathcal{A}} \to \infty$ as $|x| \to \infty$ and $|x|_{\mathcal{A}} = 0$ if and only if $x \in \mathcal{A}$. Theorem 1 provides a class-$\mathcal{KL}$ function β such that

$$|\phi(t,j)|_{\mathcal{A}} \leq \beta\big(|\phi(0,0)|_{\mathcal{A}}, t+j\big)$$

for each solution ϕ to $\mathcal{H}$ and all (t,j) in the domain of ϕ.

The system $\mathcal{H}_p$ in (21) can be modeled as a ρ-perturbation of $\mathcal{H}$ in (11)-(12) — see Section 2.3.5 of [31] and [31, Exercise 25]. The perturbed flow map is equal to the nominal one plus $(0,0,\kappa)$. Then, given $\kappa \in (-\infty, 1)$, there exists $\rho^a > 0$ such that $f_p(\xi) \subset f(\xi) + (0,0,\rho^a)\mathbb{B}$ for all ξ, where $\mathbb{B}$ denotes the closed Euclidean unit ball. Given $\theta_{\min}$ and $\theta_{\max}$ satisfying (20) there exists $\rho^b > 0$ such that $C_p \subset C + \rho^b\mathbb{B}$ and $[\tau_{\min} + \theta_{\min}, \tau_{\max} + \theta_{\max}] \subset [\tau_{\min}, \tau_{\max}] + \rho^b\mathbb{B}$. The jump map G_p satisfies $G_p(\xi) \subset G(\xi) + (0,0,\rho^b)\mathbb{B}$ for all ξ, and the jump set is simply $D_p = D$. The perturbed quantities f_p, C_p, G_p, and D_p satisfy Assumption 3.25 in [31] by inspection. Then all conditions in [31, Theorem 3.26] hold with $\rho := \max\{\rho^a, \rho^b\}$, and the theorem follows. □

6 Numerical Validation

Three different applications are considered in this section: quadratic programs, linear neural networks (inspired by [7]), logistic regression, and the Rosenbrock problem [30]. In all cases, the HyEq Toolbox (Version 2.04) [32] was used for simulation. Code for all simulations is available on GitHub[1].

Application 1 (Quadratic Program) *We consider N agents for the values $N \in \{5, 100, 500, 1000, 5000\}$. Each agent updates a scalar and they collaboratively minimize a quadratic function of the form*

$$L_1(x) := \frac{1}{2}x^\top Q x + b^\top x, \tag{22}$$

where $x \in \mathbb{R}^N$, Q is an $N \times N$ symmetric, positive definite matrix, and $b \in [1,5]^N$. For all experiments, $\tau_{\max} = \frac{1}{K+0.001}$ and $\tau_{\min} = \frac{1}{5}\tau_{\max}$.

The function L_1 in (22) from Application 1 is strongly convex and smooth, hence Assumptions 1 and 2 hold with parameters $\beta = \min \lambda(Q)$ and $K = \max \lambda(Q)$, where $\lambda(Q)$ denotes the set of eigenvalues of the matrix Q.

We consider $\beta = 2$ and $K = 4$ and initialize $\phi_x = \phi_{\eta^i}$ for all $i \in [N]$. Matlab's Optimization Toolbox was used to find L_1^*, which is compared to $L_1(\overline{\eta})$, the objective function evaluated at the shared value of η, throughout the experiment. As shown in Figure 1, expanding the network size does not have a significant impact on convergence. This demonstrates our algorithm's scalability and convergence that holds regardless of network size.

[1] Figures 1, 3, and 4 are from simulations generated with code at `http://github.com/kathendrickson/DistrHybridGD`. Figure 2 is from simulations generated with code at `https://github.com/corelabgt/DistrHybridGDComparison`.

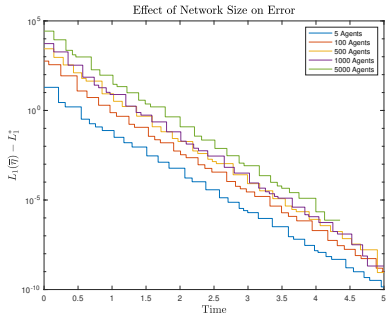


Fig. 1: Effect of network size on convergence for a strongly convex quadratic program. This is a semi-log plot, so straight lines imply exponential convergence. The horizontal axis is continuous time, and jumps are demonstrated by the sudden drops as they occur in discrete time. We see that exponential convergence is attained for all network sizes, demonstrating the scalability of our algorithm.

The preceding simulations compare problems of variable size because $x \in \mathbb{R}^N$ for $N \in \{5, 100, 500, 1000, 5000\}$. To assess scalability, we also compare problems of the same size parallelized among different numbers of agents. We consider $x \in \mathbb{R}^{5000}$ and again consider N agents for $N \in \{5, 100, 500, 1000, 5000\}$, where each agent is responsible for updating $5000/N$ decision variables. The problem setup and choices of problem parameters are the same as in the preceding simulation runs, and the results are shown in Figure 2. The curves for each run overlap almost entirely, and there is negligible impact in changing the number of agents across which the problem is partitioned, which further demonstrates that the convergence rate of our algorithm scales well.

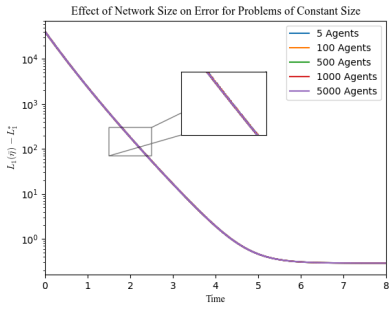


Fig. 2: Additional runs of Application 1 in which the problem size is fixed and the number of agents is variable. Changes in the number of agents have a negligible impact on the convergence rate, which further illustrates the scalability of our algorithm.

Application 2 (Logistic Regression) *We consider $N = 100$ agents collaboratively minimizing a logistic regression cost function of the form*

$$L_3(x) := \frac{1}{5}\sum_{i=1}^{5} \log(1 + \exp b_i a_i^\top x),$$

where $x \in \mathbb{R}^N$, $b_i \in [0, 10]$, and $a_i \in \{0, 1\}^N$ for $i = 1, \ldots, 5$. The parameters $\tau_{\max}$ and $\tau_{\min}$ take various values that are shown on the plots below.

While the logistic regression problem given by L_3 is smooth (satisfying Assumption 1) and convex, it is not strongly convex. However, according to [19, Section 2.3], L_3 satisfies the PL condition, which is Assumption 2, over any compact set. We therefore define the set $\left\{\xi = (x, \eta, \tau) : |\xi|_{\mathcal{A}} \leq |\phi(0, 0)|_{\mathcal{A}}\right\}$, which is compact by construction, and we know that L_3 satisfies Assumption 1 over this set. Then, Theorem 1 implies that system trajectories remain in this set for all time, and thus our convergence reuslts apply to it.

A benchmark value for $\tau_{\max}$ is set to $\overline{\tau_{\max}} = \frac{1}{K+0.001}$. As shown in Figure 3a, smaller values of $\tau_{\max}$ may result in slower convergence. While it may seem counter-intuitive that larger bounds on delays may *help* convergence, this observation echoes work in discrete-time optimization by two of the authors [17] that found it necessary to balance between (i) delaying communications to allow agents to make progress with their current state values and (ii) communicating more often to reduce disagreements among agents. In this problem, the interpretation of this idea is that a larger $\tau_{\max}$ allows agents' computations to make progress toward a minimizer before their next communication, which produces a net benefit to the overall convergence of the algorithm. Next, various choices of $\tau_{\min}$ are compared. Results shown in Figure 3b suggest that the choice of $\tau_{\min}$ does not significantly impact convergence.

Application 3 (Rosenbrock Problem [30]) *Consider $N = 2$ agents minimizing the Rosenbrock function given by*

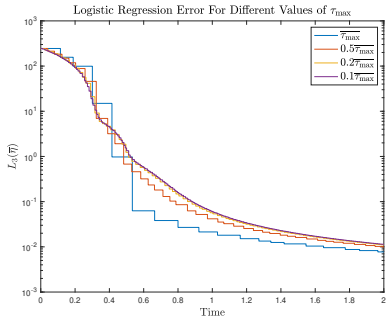
$$L_4(x) := (1 - x_1)^2 + 100(x_2 - x_1^2)^2,$$

where $x = (x_1, x_2) \in \mathbb{R}^2$. Here, the values $\tau_{\max} = 0.001$ and $\tau_{\min} = \frac{1}{5}\tau_{\max}$ were used.

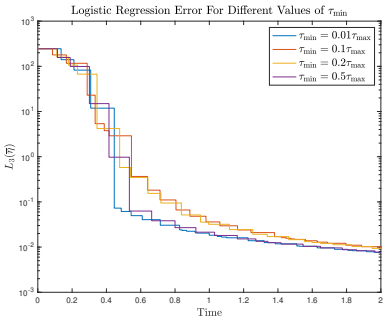
The non-convex problem given by L_4 is often used as a benchmark problem for optimization algorithms as it is difficult to solve due to the problem's geometry. However, it satisfies Assumptions 1 and 2 in the region $[-1, 1]^2$ and has a global optimum at the point $(1, 1)$ [5]. Figure 4 plots the distance of the shared value $\overline{\eta}$ from the global optimum at $(1, 1)$ throughout the simulation. While convergence is slower than the previous examples and the bound on $\tau_{\max}$ is relatively small compared to previous examples, exponential convergence is still achieved.

7 Conclusion

This paper presented a hybrid systems framework for analyzing continuous-time multi-agent optimization with discrete-time communications. Using this framework, we were able to establish that every maximal solution is complete, as well as the global exponential convergence of a block coordinate descent law to a minimizer of a smooth, possibly nonconvex, objective function that satisfies the PL inequality. Finally, three applications were considered with simulation results demonstrating the scalability and performance of our framework.



(a) Effect of $\tau_{\max}$ on convergence demonstrating that larger values of $\tau_{\max}$ may help convergence. The largest value of $\tau_{\max}$, indicated with the blue line, achieves the best performance.



(b) Effect of $\tau_{\min}$ on convergence demonstrating the insignificant impact it appears to have on convergence. In contrast to the choice of $\tau_{\max}$, for at least some problems, the choice of $\tau_{\min}$ does not seem significant. The plot here shows that both the smallest (blue line) and largest (purple line) choices for $\tau_{\min}$ have similar performance.

Fig. 3: Effects of $\tau_{\max}$ and $\tau_{\min}$ on convergence for the logistic regression problem in Application 3.

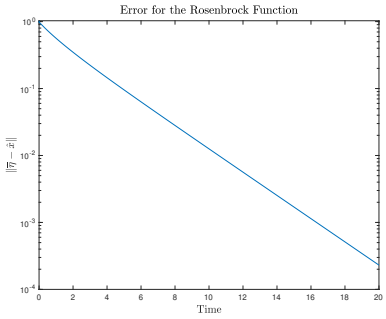


Fig. 4: Distance from the optimizer for the Rosenbrock problem, which decreases exponentially.

References

1. Raef Bassily, Mikhail Belkin, and Siyuan Ma. On exponential convergence of SGD in non-convex over-parametrized learning. arXiv preprint arXiv:1811.02564, 2018. https://doi.org/10.48550/arxiv.1811.02564.
2. Gabriel Behrendt and Matthew Hale. A totally asynchronous algorithm for time-varying convex optimization problems. IFAC-PapersOnLine, 56(2):5203–5208, 2023. https://doi.org/10.1016/j.ifacol.2023.10.116.
3. Dimitri Bertsekas, Angelia Nedic, and Asuman Ozdaglar. Convex Analysis and Optimization. Athena Scientific, 2003.
4. Dimitri P. Bertsekas and John N. Tsitsiklis. Parallel and Distributed Computation: Numerical Methods. Prentice-Hall, Inc., USA, 1989.

5. Param Budhraja, Mayank Baranwal, Kunal Garg, and Ashish Hota. Breaking the convergence barrier: Optimization via fixed-time convergent flows. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 6115–6122, 2022. https://doi.org/10.1609/aaai.v36i6.20559.
6. Jun Chai, Pedro Casau, and Ricardo G Sanfelice. Analysis and design of event-triggered control algorithms using hybrid systems tools. International Journal of Robust and Nonlinear Control, 30(15):5936–5965, 2020. https://doi.org/10.1002/rnc.5141.
7. Zachary Charles and Dimitris Papailiopoulos. Stability and generalization of learning algorithms that converge to global optima. In International Conference on Machine Learning, pages 745–754. PMLR, 2018.
8. Mung Chiang. Nonconvex optimization for communication networks. In David Y. Gao and Hanif D. Sherali, editors, Advances in Applied Mathematics and Global Optimization, pages 137–196. Springer, 2009. https://doi.org/10.1007/978-0-387-75714-8_5.
9. Jorge Cortes, Sonia Martinez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. IEEE Transactions on Robotics and Automation, 20(2):243–255, 2004. https://doi.org/10.1109/TRA.2004.824698.
10. Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In International Conference on Machine Learning, pages 1675–1685. PMLR, 2019.
11. Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In International Conference on Machine Learning, pages 1467–1476. PMLR, 2018.
12. Francesco Ferrante, Frédéric Gouaisbaut, Ricardo G Sanfelice, and Sophie Tarbouriech. State estimation of linear systems in the presence of sporadic measurements. Automatica, 73:101–109, 2016.
13. Kunal Garg and Dimitra Panagou. Fixed-time stable gradient flows: Applications to continuous-time optimization. IEEE Transactions on Automatic Control, 66(5):2002–2015, 2020. https://doi.org/10.1109/TAC.2020.3001436.
14. B. Gharesifard and J. Cortés. Distributed continuous-time convex optimization on weight-balanced digraphs. IEEE Transactions on Automatic Control, 59(3): 781–786, 2014. https://doi.org/10.1109/TAC.2013.2278132.
15. R. Goebel, R.G. Sanfelice, and A.R. Teel. Hybrid Dynamical Systems: Modeling, Stability, and Robustness. Princeton University Press, Princeton, NJ, USA, 2012.
16. Robert Gower, Othmane Sebbouh, and Nicolas Loizou. SGD for structured nonconvex functions: Learning rates, minibatching and interpolation. In International Conference on Artificial Intelligence and Statistics, pages 1315–1323. PMLR, 2021.
17. Katherine R. Hendrickson and Matthew T. Hale. Totally asynchronous primal-dual convex optimization in blocks. IEEE Transactions on Control of Network Systems, 10(1):454–466, 2023. https://doi.org/10.1109/TCNS.2022.3203366.
18. Katherine R. Hendrickson, Dawn M. Hustig-Schultz, Matthew T. Hale, and Ricardo G. Sanfelice. Exponentially converging distributed gradient descent with intermittent communication via hybrid methods. In 60th IEEE Conference on Decision and Control (CDC), pages 1186–1191, 2021. https://doi.org/10.1109/CDC45484.2021.9683567.

19. Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-łojasiewicz condition. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pages 795–811. Springer, 2016. https://doi.org/10.1007/978-3-319-46128-1_50.
20. Solmaz S. Kia, Jorge Cortés, and Sonia Martínez. Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication. Automatica, 55:254–264, 2015. https://doi.org/10.1016/j.automatica.2015.03.001.
21. Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. Advances in Neural Information Processing Systems, 28:2737–2745, 2015.
22. Nicolas Loizou, Sharan Vaswani, Issam Hadj Laradji, and Simon Lacoste-Julien. Stochastic Polyak step-size for SGD: An adaptive learning rate for fast convergence. In Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, volume 130, pages 1306–1314. PMLR, 2021.
23. J. Lu and C. Y. Tang. Zero-gradient-sum algorithms for distributed convex optimization: The continuous-time case. IEEE Transactions on Automatic Control, 57(9):2348–2354, 2012. https://doi.org/10.1109/TAC.2012.2184199.
24. Zhi-Quan Luo and Wei Yu. An introduction to convex optimization for communications and signal processing. IEEE Journal on Selected Areas in Communications, 24(8):1426–1438, 2006. https://doi.org/10.1109/JSAC.2006.879347.
25. Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE, 95(1):215–233, 2007. https://doi.org/10.1109/JPROC.2006.887293.
26. Sean Phillips and Ricardo G. Sanfelice. Robust distributed synchronization of networked linear systems with intermittent information. Automatica, 105:323–333, 2019. https://doi.org/10.1016/j.automatica.2019.03.020.
27. Sean Phillips, R. Scott Erwin, and Ricardo G. Sanfelice. Robust exponential stability of an intermittent transmission state estimation protocol. In 2018 Annual American Control Conference (ACC), pages 622–627, 2018. https://doi.org/10.23919/ACC.2018.8431144.
28. Boris Teodorovich Polyak. Gradient methods for minimizing functionals. Zhurnal vychislitel'noi matematiki i matematicheskoi fiziki, 3(4):643–653, 1963. http://dx.doi.org/10.1016/0041-5553(63)90382-3.
29. S. Rahili and W. Ren. Distributed continuous-time convex optimization with time-varying cost functions. IEEE Transactions on Automatic Control, 62(4):1590–1605, 2017. https://doi.org/10.1109/TAC.2016.2593899.
30. H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. The Computer Journal, 3(3):175–184, 1960. https://doi.org/10.1093/comjnl/3.3.175.
31. Ricardo G Sanfelice. Hybrid Feedback Control. Princeton University Press, Princeton, NJ, USA, 2021.
32. Ricardo G. Sanfelice, David Copp, and Pablo Nanez. A toolbox for simulation of hybrid systems in matlab/simulink: Hybrid equations (HyEQ) toolbox. In Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control, page 101–106, 2013. https://doi.org/10.1145/2461328.2461346.

33. Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. Optimization for Machine Learning. MIT Press, 2012.
34. Weijie Su, Stephen Boyd, and Emmanuel J. Candès. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. Journal of Machine Learning Research, 17(153):1–43, 2016.
35. Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via nonconvex factorization. In 56th IEEE Annual Symposium on Foundations of Computer Science, pages 270–289, 2015. https://doi.org/10.1109/FOCS.2015.25.
36. Matthew Ubl and Matthew T. Hale. Faster asynchronous nonconvex block coordinate descent with locally chosen stepsizes. In 61st IEEE Conference on Decision and Control (CDC), pages 4559–4564, 2022. https://doi.org/10.1109/CDC51059.2022.9993341.
37. Matthew Ubl, Matthew Hale, and Kasra Yazdani. Linear regularizers enforce the strict saddle property. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pages 10017–10024, 2023. https://doi.org/10.1609/aaai.v37i8.26194.
38. Diederik Verscheure, Bram Demeulenaere, Jan Swevers, Joris De Schutter, and Moritz Diehl. Time-optimal path tracking for robots: A convex optimization approach. IEEE Transactions on Automatic Control, 54(10):2318–2327, 2009. https://doi.org/10.1109/TAC.2009.2028959.
39. Patrick M. Wensing and Jean-Jacques Slotine. Beyond convexity—contraction and global convergence of gradient descent. PLOS ONE, 15(8):1–29, 08 2020. doi:10.1371/journal.pone.0236661. URL https://doi.org/10.1371/journal.pone.0236661.
40. Patrick M Wensing and Jean-Jacques Slotine. Beyond convexity—contraction and global convergence of gradient descent. PLOS One, 15(8):1–29, 2020. https://doi.org/10.1371/journal.pone.0243330.
41. Kasra Yazdani and Matthew Hale. Asynchronous parallel nonconvex optimization under the Polyak-Łojasiewicz condition. IEEE Control Systems Letters, 6: 524–529, 2021. https://doi.org/10.1109/LCSYS.2021.3082800.

Acknowledgements Research by M. T. Hale and K. R. Hendrickson partially supported by AFOSR Grants no. FA9550-19-1-0169 and FA9550-23-1-0120, ONR Grants no. N00014-24-1-2331 and N00014-21-1-2495, and AFRL Grant no. FA8651-22-F-1052. Research by D. M. Hustig-Schultz and R. G. Sanfelice partially supported by NSF Grants no. CNS-2039054 and CNS-2111688, by AFOSR Grants nos. FA9550-23-1-0145, FA9550-23-1-0313, and FA9550-23-1-0678, by AFRL Grant nos. FA8651-22-1-0017 and FA8651-23-1-0004, by ARO Grant no. W911NF-20-1-0253, and by DoD Grant no. W911NF-23-1-0158.

A Appendix

A.1 Section 3 Proofs

Proof of Lemma 4:
All gradients in (11) are well-defined under Assumption 1. Using Proposition 6.10 in [15] with $U = C$, let $\xi = (x, \eta, \tau) \in C \backslash D$. Because $C = \mathbb{R}^n \times \mathbb{R}^{nN} \times \mathcal{T}$, we see that the tangent cone to $\xi \in C \backslash D$ is

$$T_C(\xi) = \begin{cases} (-\infty, \infty)^n \times (-\infty, \infty)^{nN} \times (-\infty, \infty) & \tau \in (0, \tau_{\max}) \\ (-\infty, \infty)^n \times (-\infty, \infty)^{nN} \times (-\infty, 0] & \tau = \tau_{\max} \end{cases}.$$

Then $f(\xi) \subset T_C(\xi)$. Because $G(D) \subset C$, case (c) in Proposition 6.10 does not apply. We avoid case (b) of Proposition 6.10 by showing that there is no finite escape time for any solution. To that end, consider a maximal solution ϕ. Then $\phi_x(0,0)$ and $\phi_{\eta^i}(0,0)$ denote the initial values of ϕ_x and ϕ_{η^i} for all $i \in [N]$, respectively. Denote the time at which agents perform their first jump as $(t_1, 0)$. First, consider $\phi(0,0) \in C$. Then $\phi(t_1,0) = \phi(0,0) + t_1 f\big(\phi(0,0)\big)$, where f is from (11). At the first jump, ϕ_{x_i} remains the same for all $i \in [N]$ (i.e., we have $\phi_x(t_1,1) = \phi_x(t_1,0)$), we set $\phi_{\eta^i} = \text{col}(\phi_{x_1}, \ldots, \phi_{x_N})$ for all $i \in [N]$, and ϕ_τ is reset to $[\tau_{\min}, \tau_{\max}]$, all of which imply that $|\phi(t_1,1)| < \infty$. A similar argument proves the boundedness of $\phi(t_1,1)$ when $\phi(0,0) \in D$. Iterating this argument forward in time, we see that the flow map is piecewise constant over flow intervals and the jump map simply copies certain entries of ϕ_x into ϕ_η in the appropriate way, while ϕ_τ is always reset to a compact interval. Thus, repeating the preceding argument proves the finiteness of solutions across flow intervals and at jump times, which rules out finite escape time. Therefore, dom ϕ is unbounded and case (b) of Proposition 6.10 in [15] does not apply, which means that case (a) of that result must hold, namely, any maximal solution ϕ is complete. Finally, Zeno behavior is ruled out by noting that $\tau_{\min} > 0$.

A.2 Section 4 Proofs

Towards proving Lemma 7, we first state and prove several intermediate results. We note that given Assumption 1, the function $\nabla_i L$ is also Lipschitz, where $\nabla_i L := \frac{\partial}{\partial x_i} L$ is the derivative of L with respect to the i^{th} block of x.

Lemma 8 *Let ∇L be K-Lipschitz. Then $\nabla_i L$ is also K-Lipschitz for all $i \in [N]$, i.e., the inequality $\big|\nabla_i L(x) - \nabla_i L(y)\big| \leq K\big|x - y\big|$ holds for all $x, y \in \mathbb{R}^n$.*

Proof: From the definition of K-Lipschitz, we may write $\big|\nabla L(x) - \nabla L(y)\big| \leq K\big|x - y\big|$ for all $x, y \in \mathbb{R}^n$. Noting that $\big|\nabla_i L(x) - \nabla_i L(y)\big| \leq \big|\nabla L(x) - \nabla L(y)\big|$ gives the desired result. □

Furthermore, based on Proposition A.32 in [4], the smoothness of L allows us to apply the Descent Lemma given in Lemma 9.

Lemma 9 (Descent Lemma, Proposition A.32 in [4]) *Let $L : \mathbb{R}^n \to \mathbb{R}$ be continuously differentiable and have the Lipschitz property $\big|\nabla L(x) - \nabla L(y)\big| \leq K\big|x - y\big|$ for every $x, y \in \mathbb{R}^n$. Then for all x, y in $\mathbb{R}^n$,*

$$L(y) \leq L(x) + \nabla L(x)^\top (y - x) + \frac{K}{2}\big|y - x\big|^2. \tag{23}$$

Proof of Lemma 7:
By construction, $V(\xi)$ is zero only for $\xi \in \mathcal{A}$ and is positive otherwise. By Assumption 2, for any $x^* \in \mathcal{X}^*$, we have $L(x^*) = L^*$. For a fixed $\xi = (x, \eta, \tau)$, we define

$$\hat{x}^0 := \underset{x^* \in \mathcal{X}^*}{\arg\min} \big|x - x^*\big| \qquad \text{and} \qquad \hat{x}^i := \underset{x^* \in \mathcal{X}^*}{\arg\min} \big|\eta^i - x^*\big| \text{ for all } i \in [N].$$

Then $V(\xi)$ is equivalent to $V(\xi) = \big(L(x) - L(\hat{x}^0)\big) + \sum_{i \in [N]} \big(L(\eta^i) - L(\hat{x}^i)\big)$. Because ∇L is K-Lipschitz, Lemma 9 implies that $L(x) - L(\hat{x}^0) \leq \frac{K}{2}\big|x - \hat{x}^0\big|^2$ for all $x \in \mathbb{R}^n$. For the same reason, we have $L(\eta^i) - L(\hat{x}^i) \leq \frac{K}{2}\big|\eta^i - \hat{x}^i\big|$ for all $\eta^i \in \mathbb{R}^n$. Thus, $V(\xi)$ may be bounded as

$$V(\xi) \leq \frac{K}{2}\left(\big|x - \hat{x}^0\big|^2 + \sum_{i \in [N]} \big|\eta^i - \hat{x}^i\big|^2\right) = \frac{K}{2}\big|\xi\big|_{\mathcal{A}}^2.$$

Therefore, we set $\alpha_2(s) = \frac{K}{2} s^2 \in \mathcal{K}_\infty$ for all $s \geq 0$.

Because L is β-PL and has a Lipschitz gradient, it also satisfies the quadratic growth condition (QG) with constant β (see Theorem 2 in [19]). In particular, given any $x \in \mathbb{R}^n$, we

have $L(x) - L^* \geq \frac{\beta}{2} \min_{x^* \in \mathcal{X}^*} \left|x - x^*\right|^2$. Thus, using the definitions of $\hat{x}^0$ and $\hat{x}^i$ above, we can write $L(x) - L^* \geq \frac{\beta}{2}\left|x - \hat{x}^0\right|^2$ and $L(\eta^i) - L^* \geq \frac{\beta}{2}\left|\eta^i - \hat{x}^i\right|^2$ for all $i \in [N]$. This leads to

$$V(\xi) \geq \frac{\beta}{2}\left(\left|x - \hat{x}^0\right|^2 + \sum_{i \in [N]} \left|\eta^i - \hat{x}^i\right|^2\right) = \frac{\beta}{2}\left|\xi\right|_{\mathcal{A}}^2.$$

Setting $\alpha_1(s) = \frac{\beta}{2} s^2 \in \mathcal{K}_\infty$ for all $s \geq 0$ completes the proof. □

Proof of Proposition 1:
We first consider $\xi \in C$ and the Lyapunov function V defined in Lemma 7, where

$$\nabla V(\xi) = \begin{bmatrix} \nabla L(x) \\ \operatorname{col}\big(\nabla L(\eta^1), \ldots, \nabla L(\eta^N)\big) \\ 0 \end{bmatrix}.$$

This leads to

$$\Big\langle \nabla V(\xi), f(\xi) \Big\rangle = -\nabla L(x)^\top h(\eta) = -\sum_{i \in [N]} \nabla_i L(x)^\top \nabla_i L(\eta^i) \quad \text{for all} \quad \xi \in C, \tag{24}$$

where h is from (10). By Lemma 4, we know that every maximal solution is complete, and we now pick a maximal solution ϕ initialized such that $\phi_x(0,0) = \phi_{\eta^i}(0,0)$ for all $i \in [N]$. For each $I^j := \{t : (t,j) \in \operatorname{dom} \phi\}$ with nonempty interior and with $t_{j+1} > t_j$ such that $[t_j, t_{j+1}] = I^j$, the initialization of ϕ leads to a common value of η^i across all agents for all time, i.e., $\phi_{\eta^i}(t,j) = \phi_{\eta^\ell}(t,j)$ for all pairs of agents i and ℓ and all times (t,j). For simplicity, we denote this shared value with $\bar{\eta}$. This allows us to rewrite (24) as

$$\begin{aligned} \Big\langle \nabla V\big(\phi(t,j)\big), f\big(\phi(t,j)\big) \Big\rangle &= -\sum_{i \in [N]} \nabla_i L\big(\phi_x(t,j)\big)^\top \nabla_i L\big(\phi_{\bar{\eta}}(t,j)\big) \\ &= -\nabla L\big(\phi_x(t,j)\big)^\top \nabla L\big(\phi_{\bar{\eta}}(t,j)\big). \end{aligned} \tag{25}$$

We now apply Lemma 9 with $x = \phi_x(t,j)$ and $y = \phi_{\bar{\eta}}(t,j)$, giving

$$\begin{aligned} &L\big(\phi_{\bar{\eta}}(t,j)\big) \leq L\big(\phi_x(t,j)\big) + \nabla L\big(\phi_x(t,j)\big)^\top \big(\phi_{\bar{\eta}}(t,j) - \phi_x(t,j)\big) + \frac{K}{2}\big|\phi_{\bar{\eta}}(t,j) - \phi_x(t,j)\big|^2 \\ &= L\big(\phi_x(t,j)\big) + \sum_{i \in [N]} \nabla_i L\big(\phi_x(t,j)\big)^\top \big(\phi_{\bar{\eta}_i}(t,j) - \phi_{x_i}(t,j)\big) + \frac{K}{2}\sum_{i \in [N]} \big|\phi_{\bar{\eta}_i}(t,j) - \phi_{x_i}(t,j)\big|^2. \end{aligned}$$

Applying the relationships (13) and (14) from Lemma 5, we find

$$\begin{aligned} L\big(\phi_{\bar{\eta}}(t,j)\big) &\leq L\big(\phi_x(t,j)\big) + \frac{K}{2}\sum_{i \in [N]} \big|\phi_{\bar{\eta}_i}(t_j,j) - \phi_{\bar{\eta}_i}(t_j,j) + (t - t_j)\nabla_i L\big(\phi_{\bar{\eta}}(t_j,j)\big|^2 \\ &\quad + \sum_{i \in [N]} \nabla_i L\big(\phi_x(t,j)\big)^\top \Big(\phi_{\bar{\eta}_i}(t_j,j) - \phi_{\bar{\eta}_i}(t_j,j) + (t - t_j)\nabla_i L\big(\phi_{\bar{\eta}}(t_j,j)\big)\Big) \\ &= L\big(\phi_x(t,j)\big) + (t - t_j)\nabla L\big(\phi_x(t,j)\big)^\top \nabla L\big(\phi_{\bar{\eta}}(t,j)\big) + \frac{K}{2}(t - t_j)^2\big|\nabla L\big(\phi_{\bar{\eta}}(t,j)\big)\big|^2. \end{aligned}$$

Rearranging gives

$$\begin{aligned} -(t - t_j)\nabla L\big(\phi_x(t,j)\big)^\top \nabla L\big(\phi_{\bar{\eta}}(t,j)\big) \leq \\ \Big(L\big(\phi_x(t,j)\big) - L\big(\phi_{\bar{\eta}}(t,j)\big)\Big) + \frac{K}{2}(t - t_j)^2\big|\nabla L\big(\phi_{\bar{\eta}}(t,j)\big)\big|^2. \end{aligned} \tag{26}$$

We now apply (23) from Lemma 9 once more, using $x = \phi_{\bar{\eta}}(t,j)$ and $y = \phi_x(t,j)$ to write

$$\begin{aligned} L\big(\phi_x(t,j)\big) - L\big(\phi_{\bar{\eta}}(t,j)\big) &\leq \nabla L\big(\phi_{\bar{\eta}}(t,j)\big)^\top \big(\phi_x(t,j) - \phi_{\bar{\eta}}(t,j)\big) + \frac{K}{2}\big|\phi_x(t,j) - \phi_{\bar{\eta}}(t,j)\big|^2 \\ &= \sum_{i\in[N]} \nabla_i L\big(\phi_{\bar{\eta}}(t,j)\big)^\top \big(\phi_{x_i}(t,j) - \phi_{\bar{\eta}_i}(t,j)\big) + \frac{K}{2}\sum_{i\in[N]}\big|\phi_{x_i}(t,j) - \phi_{\bar{\eta}_i}(t,j)\big|^2 \\ &= -(t-t_j)\sum_{i\in[N]} \nabla_i L\big(\phi_{\bar{\eta}}(t,j)\big)^\top \nabla_i L\big(\phi_{\bar{\eta}}(t,j)\big) + \frac{K}{2}(t-t_j)^2\big|\nabla L\big(\phi_{\bar{\eta}}(t,j)\big)\big|^2 \\ &= -(t-t_j)\big|\nabla L\big(\phi_{\bar{\eta}}(t,j)\big)\big|^2 + \frac{K}{2}(t-t_j)^2\big|\nabla L\big(\phi_{\bar{\eta}}(t,j)\big)\big|^2, \end{aligned} \tag{27}$$

where the second equality applies Lemma 5. Applying the last inequality to (26) and grouping terms, we find

$$-(t-t_j)\nabla L\big(\phi_x(t,j)\big)^\top \nabla L\big(\phi_{\bar{\eta}}(t,j)\big) \leq -(t-t_j)(1-K\tau_{\max})\big|\nabla L\big(\phi_{\bar{\eta}}(t,j)\big)\big|^2.$$

Dividing by $t - t_j$, which is positive by definition, gives

$$-\nabla L\big(\phi_x(t,j)\big)^\top \nabla L\big(\phi_{\bar{\eta}}(t,j)\big) \leq -(1-K\tau_{\max})\big|\nabla L\big(\phi_{\bar{\eta}}(t,j)\big)\big|^2, \tag{28}$$

where the right hand side is negative for $1 - K\tau_{\max} > 0$, which is satisfied for $\tau_{\max} < \frac{1}{K}$. Furthermore, the right hand side of (28) will be zero only when an optimum of L has been reached, namely at a point at which ∇L is zero. Finally, applying the β-PL condition from Assumption 2 allows us to write

$$-\nabla L\big(\phi_x(t,j)\big)^\top \nabla L\big(\phi_{\bar{\eta}}(t,j)\big) \leq -2\beta(1-K\tau_{\max})\Big(L\big(\phi_{\bar{\eta}}(t,j)\big) - L^*\Big). \tag{29}$$

Looking once more at (27), we note that

$$\begin{aligned} L\big(\phi_x(t,j)\big) - L\big(\phi_{\bar{\eta}}(t,j)\big) &\leq -(t-t_j)\Big(1 - \frac{K}{2}(t-t_j)\Big)\big|\nabla L\big(\phi_{\bar{\eta}}(t,j)\big)\big|^2 \\ &\leq -\tau_{\min}\Big(1 - \frac{K}{2}\tau_{\max}\Big)\big|\nabla L\big(\phi_{\bar{\eta}}(t,j)\big)\big|^2, \end{aligned} \tag{30}$$

where the right hand side is negative since $\tau_{\max} < \frac{1}{K}$. Thus, $L\big(\phi_x(t,j)\big) \leq L\big(\phi_{\bar{\eta}}(t,j)\big)$ and

$$V\big(\phi(t,j)\big) = \Big(L\big(\phi_x(t,j)\big) - L^*\Big) + \sum_{i\in[N]}\Big(L\big(\phi_{\eta^i}(t,j)\big) - L^*\Big) \leq (N+1)\Big(L\big(\phi_{\bar{\eta}}(t,j)\big) - L^*\Big).$$

Combining this inequality with (29), we have

$$-\nabla L\big(\phi_x(t,j)\big)^\top \nabla L\big(\phi_{\bar{\eta}}(t,j)\big) \leq -\frac{2}{N+1}\beta(1-K\tau_{\max})V\big(\phi(t,j)\big).$$

Combined with (25), we conclude that

$$\langle V(\phi(t,j)), f(\phi(t,j))\rangle \leq -\frac{2}{N+1}\beta(1-K\tau_{\max})V\big(\phi(t,j)\big) \leq 0. \tag{31}$$

Next consider the change of V at jumps. For a maximal solution ϕ such that $\phi_x(0,0) = \phi_{\eta^i}(0,0)$ for all $i \in [N]$, we may write the change at jump $j+1$ as

$$\begin{aligned} V\big(G(\phi(t_{j+1},j))\big) - V\big(\phi(t_{j+1},j)\big) &= \Big(L\big(\phi_x(t_{j+1},j+1)\big) - L^*\Big) + N\Big(L\big(\phi_{\bar{\eta}}(t_{j+1},j+1)\big) - L^*\Big) \\ &\quad - \Big(L\big(\phi_x(t_{j+1},j)\big) - L^*\Big) - N\Big(L\big(\phi_{\bar{\eta}}(t_{j+1},j)\big) - L^*\Big) \\ &= N\Big(L\big(\phi_{\bar{\eta}}(t_{j+1},j+1)\big) - L\big(\phi_{\bar{\eta}}(t_{j+1},j)\big)\Big) = N\Big(L\big(\phi_x(t_{j+1},j)\big) - L\big(\phi_{\bar{\eta}}(t_{j+1},j)\big)\Big) \end{aligned}$$

for all $(t_{j+1}, j), (t_{j+1}, j+1) \in \operatorname{dom} \phi$. For this quantity to be nonpositive, it is sufficient to show that $L\big(\phi_x(t_{j+1}, j)\big) \leq L\big(\phi_{\bar{\eta}}(t_{j+1}, j)\big)$. This inequality follows directly from (30) by setting $t = t_{j+1}$, where

$$L\big(\phi_x(t_{j+1}, j)\big) - L\big(\phi_{\bar{\eta}}(t_{j+1}, j)\big) \leq -\tau_{\min}\Big(1 - \frac{K}{2}\tau_{\max}\Big)\big|\nabla L\big(\phi_{\bar{\eta}}(t_{j+1}, j)\big)\big|^2.$$

For $\tau_{\max} < \frac{1}{K}$, the term $1 - \frac{K}{2}\tau_{\max}$ is positive. Then $L\big(\phi_x(t_{j+1}, j)\big) \leq L\big(\phi_{\bar{\eta}}(t_{j+1}, j)\big)$ and

$$V\big(G(\phi(t_{j+1}, j))\big) - V\big(\phi(t_{j+1}, j)\big) \leq 0. \tag{32}$$

Following the work done in [6] and [15], we are able to perform direct integration in order to upper bound $V\big(\phi(t, j)\big)$ in terms of $V\big(\phi(0, 0)\big)$ using (31) and (32) as bounds. Thus,

$$V\big(\phi(t, j)\big) \leq \exp\Big(-\frac{2}{N+1}\beta(1 - K\tau_{\max})t\Big)V\big(\phi(0, 0)\big).$$

Using the comparison functions given in Lemma 7, we find a bound for $\big|\phi(t, j)\big|^2$ via

$$\begin{aligned}\big|\phi(t, j)\big|_{\mathcal{A}}^2 &\leq \frac{2}{\beta}\exp\Big(-\frac{2}{N+1}\beta(1 - K\tau_{\max})t\Big)V\big(\phi(0, 0)\big) \\ &\leq \frac{K}{\beta}\exp\Big(-\frac{2}{N+1}\beta(1 - K\tau_{\max})t\Big)\big|\phi(0, 0)\big|_{\mathcal{A}}^2,\end{aligned}$$

where taking the square root gives the final bound. □

Proof of Proposition 2:

Two initialization scenarios must be considered: $\phi_x(0, 0) = \phi_{\eta^i}(0, 0)$ for all $i \in [N]$ and $\phi_x(0, 0) \neq \phi_{\eta^i}(0, 0)$ for at least one $i \in [N]$. For the first case, $\phi_x(0, 0) = \phi_{\eta^i}(0, 0)$ for all $i \in [N]$, Proposition 1 applies in its original form. This is the best-case scenario that results in the smallest upper bound on the distance to $\mathcal{A}$.

Now consider the second case, when $\phi_x(0, 0) \neq \phi_{\eta^i}(0, 0)$ for at least one $i \in [N]$. After agents have completed at least one jump, all assumptions of Proposition 1 hold. Denote the time at which the first jump occurs as $(t_1, 1)$. Thus, for any maximal solution ϕ (which is complete by Lemma 4), for any $(t, j) \in \operatorname{dom} \phi$ such that $j \geq 1$, the following holds:

$$\big|\phi(t, j)\big|_{\mathcal{A}} \leq \sqrt{\frac{K}{\beta}}\exp\Big(-\frac{\beta}{N+1}(1 - K\tau_{\max})t\Big)\big|\phi(t_1, 1)\big|_{\mathcal{A}}. \tag{33}$$

We now seek to bound $\big|\phi(t_1, 1)\big|_{\mathcal{A}}$ in terms of $\big|\phi(0, 0)\big|_{\mathcal{A}}$. We first define the point $\bar{x}^0 := \arg\min_{x^* \in \mathcal{X}^*}\big|\phi_x(t_1, 1) - x^*\big|$, and, for each $i \in [N]$, we define $\bar{x}^i := \arg\min_{x^* \in \mathcal{X}^*}\big|\phi_{\eta^i}(t_1, 1) - x^*\big|$. We begin by expanding $\big|\phi(t_1, 1)\big|_{\mathcal{A}}^2$ to find

$$\big|\phi(t_1, 1)\big|_{\mathcal{A}}^2 = \big|\phi_x(t_1, 1) - \bar{x}^0\big|^2 + \sum_{i \in [N]}\big|\phi_{\eta^i}(t_1, 1) - \bar{x}^i\big|^2. \tag{34}$$

We now define $\hat{x}^0 := \arg\min_{x^* \in \mathcal{X}^*}\big|\phi_x(0, 0) - x^*\big|$. Note that by definition of $\bar{x}^0$ and $\bar{x}^i$, the following inequalities hold:

$$\begin{aligned}\big|\phi_x(t_1, 1) - \bar{x}^0\big|^2 &\leq \big|\phi_x(t_1, 1) - \hat{x}^0\big|^2 = \sum_{i \in [N]}\big|\phi_{x_i}(t_1, 1) - \hat{x}_i^0\big|^2 \\ \big|\phi_{\eta^i}(t_1, 1) - \bar{x}^i\big|^2 &\leq \big|\phi_{\eta^i}(t_1, 1) - \hat{x}^0\big|^2.\end{aligned}$$

These inequalities allow us to rewrite (34) as

$$\big|\phi(t_1, 1)\big|_{\mathcal{A}}^2 \leq \sum_{i \in [N]}\big|\phi_{x_i}(t_1, 1) - \hat{x}_i^0\big|^2 + \sum_{i \in [N]}\big|\phi_{\eta^i}(t_1, 1) - \hat{x}^0\big|^2. \tag{35}$$

Define $\hat{x}^i := \arg\min_{x^* \in \mathcal{X}^*} \left|\phi_{\eta^i}(0,0) - x^*\right|$ for all $i \in [N]$. We first upper bound $\left|\phi_{x_i}(t_1,1) - \hat{x}_i^0\right|^2$ by applying Lemma 5 and using $\left|a-b\right|^2 \leq 2\left|a\right|^2 + 2\left|b\right|^2$, resulting in

$$\begin{aligned}\left|\phi_{x_i}(t_1,1) - \hat{x}_i^0\right|^2 &= \left|\phi_{x_i}(t_1,0) - \hat{x}_i^0\right|^2 = \left|\phi_{x_i}(0,0) - t_1\nabla_i L\big(\phi_{\eta^i}(0,0)\big) - \hat{x}_i^0\right|^2 \\ &\leq 2\left|\phi_{x_i}(0,0) - \hat{x}_i^0\right|^2 + 2K^2\tau_{\max}^2\left|\phi_{\eta^i}(0,0) - \hat{x}^i\right|^2, \end{aligned} \tag{36}$$

where the first equality applies (12), the second equality applies Lemma 5, the first inequality uses $\nabla L(\hat{x}^i) = 0$, and the final inequality applies Lemma 8. Summing over all i on both sides of (36) gives

$$\left|\phi_x(t_1,1) - \hat{x}^0\right|^2 \leq 2\left|\phi_x(0,0) - \hat{x}^0\right|^2 + 2K^2\tau_{\max}^2 \sum_{i\in[N]} \left|\phi_{\eta^i}(0,0) - \hat{x}^i\right|^2. \tag{37}$$

An upper bound on $\left|\phi_{\eta^i}(t_1,1) - \hat{x}^0\right|^2$ also needs to be derived to upper bound the right hand side of (35). Recall that at any jump j, we have $\phi_{\eta^i}(t_j,j) = \phi_x(t_j,j)$ for all i. Thus, $\phi_{\eta^i}(t_1,1) = \phi_x(t_1,1)$ for all i. We use this to expand and upper bound the following:

$$\begin{aligned}\sum_{i\in[N]} \left|\phi_{\eta^i}(t_1,1) - \hat{x}^0\right|^2 &= N\left|\phi_x(t_1,1) - \hat{x}^0\right|^2 \\ &\leq 2N\left|\phi_x(0,0) - \hat{x}^0\right|^2 + 2NK^2\tau_{\max}^2 \sum_{i\in[N]} \left|\phi_{\eta^i}(0,0) - \hat{x}^i\right|^2, \end{aligned} \tag{38}$$

where the inequality applies (37). Summing (37) and (38) gives

$$\begin{aligned}&\left|\phi_x(t_1,1) - \hat{x}^0\right|^2 + \sum_{i\in[N]} \left|\phi_{\eta^i}(t_1,1) - \hat{x}^0\right|^2 \leq 2\left|\phi_x(0,0) - \hat{x}^0\right|^2 \\ &\quad + 2K^2\tau_{\max}^2 \sum_{i\in[N]} \left|\phi_{\eta^i}(0,0) - \hat{x}^i\right|^2 + 2N\left|\phi_x(0,0) - \hat{x}^0\right|^2 + 2NK^2\tau_{\max}^2 \sum_{i\in[N]} \left|\phi_{\eta^i}(0,0) - \hat{x}^i\right|^2 \\ &\quad = 2(N+1)\left|\phi_x(0,0) - \hat{x}^0\right|^2 + 2(N+1)K^2\tau_{\max}^2 \sum_{i\in[N]} \left|\phi_{\eta^i}(0,0) - \hat{x}^i\right|^2 \\ &\quad \leq 2(N+1)\left(\left|\phi_x(0,0) - \hat{x}^0\right|^2 + \sum_{i\in[N]} \left|\phi_{\eta^i}(0,0) - \hat{x}^i\right|^2\right), \end{aligned}$$

where the first equality groups terms and the second inequality uses $\tau_{\max} < \frac{1}{K}$ to simplify. Applying (35) on the left-hand side and the definition of $|\cdot|_{\mathcal{A}}^2$ on the right-hand side gives

$$\left|\phi(t_1,1)\right|_{\mathcal{A}}^2 \leq 2(N+1)\left|\phi(0,0)\right|_{\mathcal{A}}^2.$$

Taking the square root and applying the resulting inequality to (33) gives a bound for all $j \geq 1$. □