Achieving Linear Convergence in Distributed Asynchronous Multi-agent Optimization

Ye Tian, Ying Sun, and Gesualdo Scutari

***Abstract*—This papers studies multi-agent (convex and *nonconvex*) optimization over static digraphs. We propose a general distributed *asynchronous* algorithmic framework whereby i) agents can update their local variables as well as communicate with their neighbors at any time, without any form of coordination; and ii) they can perform their local computations using (possibly) delayed, out-of-sync information from the other agents. Delays need not be known to the agent or obey any specific profile, and can also be time-varying (but bounded). The algorithm builds on a tracking mechanism that is robust against asynchrony (in the above sense), whose goal is to estimate locally the average of agents' gradients. When applied to strongly convex functions, we prove that it converges at an R-linear (geometric) rate as long as the step-size is sufficiently small. A sublinear convergence rate is proved, when nonconvex problems and/or diminishing, *uncoordinated* step-sizes are considered. To the best of our knowledge, this is the first distributed algorithm with provable geometric convergence rate in such a general asynchronous setting. Preliminary numerical results demonstrate the efficacy of the proposed algorithm and validate our theoretical findings.**

***Index Terms*—Asynchrony, Delay, Directed graphs, Distributed optimization, Linear convergence, Nonconvex optimization.**

I. Introduction

We study convex and nonconvex distributed optimization over a network of agents, modeled as a directed fixed graph. Agents aim at cooperatively solving the optimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^n} F(\mathbf{x}) \triangleq \sum_{i=1}^{I} f_i(\mathbf{x}) \tag{P}$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is the cost function of agent i, assumed to be smooth (nonconvex) and known only to agent i. In this setting, optimization has to be performed in a distributed, collaborative manner: agents can only receive/send information from/to its immediate neighbors. Instances of (P) that require distributed computing have found a wide range of applications in different areas, including network information processing, resource allocation in communication networks, swarm robotic, and machine learning, just to name a few.

Many of the aforementioned applications give rise to extremely large-scale problems and networks, which naturally call for *asynchronous*, *parallel* solution methods. In fact, asynchronous modus operandi reduces the idle times of workers, mitigate communication and/or memory-access congestion, save power (as agents need not perform computations and communications at every iteration), and make algorithms more fault-tolerant. In this paper, we consider the following very general, abstract, asynchronous model [3]:

Part of this work has been presented at the 56th Annual Allerton Conference [1] and posted on arxiv [2] on March 2018. This work has been supported by the USA National Science Foundation under Grants CIF 1632599 and CIF 1719205; and in part by the Office of Naval Research under Grant N00014-16-1-2244, and the Army Research Office under Grant W911NF1810238.

The authors are with the School of Industrial Engineering, Purdue University, West-Lafayette, IN, USA; Emails: {tian110,sun578,gscutari}@purdue.edu.

(i) Agents can perform their local computations as well as communicate (possibly in parallel) with their immediate neighbors at any time, without any form of coordination or centralized scheduling; and

(ii) when solving their local subproblems, agents can use outdated information from their neighbors.

In (ii) no constraint is imposed on the delay profiles: delays can be arbitrary (but bounded), time-varying, and (possibly) dependent on the specific activation rules adopted to wakeup the agents in the network. This model captures in a unified fashion several forms of asynchrony: some agents execute more iterations than others; some agents communicate more frequently than others; and inter-agent communications can be unreliable and/or subject to unpredictable, time-varying delays.

Several forms of asynchrony have been studied in the literature–see Sec. I-A for an overview of related works. However, we are not aware of any distributed algorithm that is compliant to the asynchrony model (i)-(ii) and distributed (nonconvex) setting above. Furthermore, when considering the special case of a strongly convex function F, it is not clear how to design a (first-order) *distributed asynchronous* algorithm (as specified above) that achieves *linear convergence* rate. This paper answers these questions–see Sec. I-B and Table 1 for a summary of our contributions.

A. Literature Review

Since the seminal work [11], asynchronous parallelism has been applied to several *centralized* optimization algorithms, including block coordinate descent (e.g., [11]–[13]) and stochastic gradient (e.g., [14], [15]) methods. However, these schemes are not applicable to the networked setup considered in this paper, because they would require the knowledge of the function F from each agent. *Distributed* methods exploring (some form of) asynchrony over networks with no centralized node have been studied in [4]–[10], [16]–[26]. We group next these works based upon the features (i)-(ii) above.

(a) Random activations and no delays [16]–[20]: These schemes considered distributed *convex* unconstrained optimization over *undirected* graphs. While substantially different in the form of the updates performed by the agents– [16], [18], [20] are instances of primal-dual (proximal-based) algorithms, [19] is an ADMM-type algorithm, while [17] is based on the distributed gradient tracking mechanism introduced in [27]–[29]–all these algorithms are asynchronous in the sense of feature (i) [but not (ii)]: at each iteration, a subset of agents [16], [18], [20] (or edge-connected agents [17], [19]), chosen at random, is activated, performing then their updates and communications with their immediate neighbors; between two activations, agents are assumed to be in *idle* mode (i.e., able to *continuously* receive information). However, *no form of*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2020.2977940, IEEE Transactions on Automatic Control

Algorithm	Nonconvex Cost Function	No Idle Time	Arbitrary Delays	Parallel	Step Sizes		Digraph	Global Convergence to Exact Solutions	Rate Analysis	
					Fixed	Uncoordinated Diminishing			Linear Rate for Strongly Convex	Nonconvex
Asyn. Broadcast [4]				✓	✓	✓		In expectation (w. diminishing step)		
Asyn. Diffusion [5]					✓					
Asyn. ADMM [6]	✓				✓			Deterministic		
Dual Ascent in [7]		✓	Restricted	Restricted	✓					
ra-NRC [8]					✓		✓			
ARock [9]		✓	Restricted		✓			Almost surely	In expectation	
ASY-PrimalDual [10]		✓	Restricted		✓			Almost surely		
ASY-SONATA	✓	✓	✓	✓	✓	✓	✓	Deterministic	Deterministic	Deterministic

Table 1
COMPARISON WITH STATE-OF-ART DISTRIBUTED ASYNCHRONOUS ALGORITHMS. CURRENT SCHEMES CAN DEAL WITH UNCOORDINATED ACTIVATIONS BUT ONLY WITH SOME FORMS OF DELAYS. ASY-SONATA ENJOYS ALL THE DESIRABLE FEATURES LISTED IN THE TABLE.

delays is allowed: every agent must perform its local computations/updates using the *most updated* information from its neighbors. This means that all the actions performed by the agent(s) in an activation must be completed before a new activation (agent) takes place (wakes-up), which calls for some coordination among the agents. Finally, no convergence rate was provided for the aforementioned schemes but [17], [19].

(b) Synchronous activations and delays [21]–[26]: These schemes considered distributed constrained *convex* optimization over *undirected* graphs. They study the impact of delayed gradient information [21], [22] or communication delays (fixed [23], uniform [22], [26] or time-varying [24], [25]) on the convergence rate of distributed gradient (proximal [21], [22] or projection-based [25], [26]) algorithms or dual-averaging distributed-based schemes [23], [24]. While these schemes are all synchronous [thus lacking of feature (i)], they can tolerate *communication delays* [an instantiation of feature (ii)], converging at a *sublinear rate* to an optimal solution. Delays must be such that no losses occur–every agent's message will eventually reach its destination within a finite time.

(c) Random/cyclic activations and some form of delays [4]–[10]: The class of optimization problems along with the key features of the algorithms proposed in these papers are summarized in Table 1 and briefly discussed next. The majority of these works studied distributed (strongly) *convex* optimization over *undirected* graphs, with [5] assuming that all the functions f_i have the same minimizer, [6] considering also nonconvex objectives, and [8] being implementable also over digraphs. The algorithms in [4], [5] are gradient-based schemes; [6] is a decentralized instance of ADMM; [9] applies an asynchronous parallel ADMM scheme to distributed optimization; and [10] builds on a primal-dual method. The schemes in [7], [8] instead build on (approximate) second-order information. All these algorithms are asynchronous in the sense of feature (i): [4]–[6], [9], [10] considered random activations of the agents (or edges-connected agents) while [7], [8] studied deterministic, uncoordinated activation rules. As far as feature (ii) is concerned, some form of delays is allowed. More specifically, [4]–[6], [8] can deal with *packet losses*: the information sent by an agent to its neighbors either gets lost or received with *no delay*. They also assume that agents are *always in idle mode* between two activations. Closer to the proposed asynchronous framework are the schemes in [9], [10] wherein a probabilistic model is employed to describe the activation of the agents and the aged information used in their updates. The model requires that the random variables triggering the activation of the agents are i.i.d and *independent* of the delay vector used by the agent to performs its update. While this assumption makes the convergence analysis possible, in reality, there is a strong dependence of the delays on the activation index; see [13] for a detailed discussion on this issue and several counter examples. Other consequences of this model are: the schemes [9], [10] are *not parallel*–only one agent per time can perform the update–and a random self-delay must be used in the update of each agent (even if agents have access to their most recent information). Furthermore, [9] calls for the solution of a convex subproblem for each agent at every iteration. Referring to the convergence rate, [9] is the only scheme exhibiting linear convergence *in expectation*, when each f_i is strongly convex and the graph *undirected*. No convergence rate is available in any of the aforementioned papers, when F is nonconvex.

B. Summary of Contributions

This paper proposes a general distributed, asynchronous algorithmic framework for (strongly) convex and *nonconvex* instances of Problem (P), over *directed* graphs. The algorithm leverages a perturbed "sum-push" mechanism that is robust against asynchrony, whose goal is to track locally the average of agents' gradients; this scheme along with its convergence analysis are of independent interest. To the best of our knowledge, the proposed framework is the first scheme combining the following attractive features (cf. Table 1): (a) it is *parallel and asynchronous [in the sense (i) and (ii)]*–multiple agents can be activated at the same time (with no coordination) and/or outdated information can be used in the agents' updates; our asynchronous setting (i) and (ii) is less restrictive than the one in [9], [10]; furthermore, in contrast with [9], our scheme avoids solving possibly complicated subproblems; (b) it is applicable to *nonconvex* problems, with provable convergence to stationary solutions of (P); (c) it is implementable over *digraph*; (d) it employs either a constant step-size or *uncoordinated* diminishing ones; (e) it *converges at an R-linear rate* (resp. sublinear) when F is strongly convex (resp. nonconvex) and a constant (resp. diminishing, uncoordinated) step-size(s) is employed; this contrasts [9] wherein each f_i needs to be strongly convex; and (f) it is "protocol-free", meaning that agents need not obey any specific communication protocols or asynchronous modus operandi (as long as delays are bounded and agents update/communicate uniformly infinitely often).

On the technical side, convergence is studied introducing two techniques of independent interest, namely: i) the asynchronous agent system is reduced to a synchronous "augmented" one with no delays by adding virtual agents to the graph. While this idea was first explored in [30], [31], [32], the proposed enlarged system and algorithm differ from those used therein, which cannot deal with the general asynchronous model considered here–see Remark 13, Sec.VI; and ii) the rate analysis is employed putting forth a generalization of the small gain theorem (widely used in the literature [33]

to analyze synchronous schemes), which is expected to be broadly applicable to other distributed algorithms.

C. Notation

Throughout the paper we use the following notation. Given the matrix $\mathbf{M} \triangleq (M_{ij})_{i,j=1}^{I}$, $\mathbf{M}_{i,:}$ and $\mathbf{M}_{:,j}$ denote its i-th row vector and j-th column vector. Given the sequence $\{\mathbf{M}^t\}_{t=s}^{k}$, with $k \geq s$, we define $\mathbf{M}^{k:s} \triangleq \mathbf{M}^k\mathbf{M}^{k-1}\cdots\mathbf{M}^{s+1}\mathbf{M}^s$, if $k > s$; and $\mathbf{M}^{k:s} \triangleq \mathbf{M}^s$ otherwise. Given two matrices (vectors) $\mathbf{A}$ and $\mathbf{B}$ of same size, by $\mathbf{A} \preccurlyeq \mathbf{B}$ we mean that $\mathbf{B} - \mathbf{A}$ is a nonnegative matrix (vector). The dimensions of the all-one vector $\mathbf{1}$ and the i-th canonical vector $\mathbf{e}_i$ will be clear from the context. We use $\|\cdot\|$ to represent the Euclidean norm for a vector whereas the spectral norm for a matrix. The indicator function $\mathbb{1}[E]$ of an event E equals to 1 when the event E is true, and 0 otherwise. Finally, we use the convention $\sum_{t\in\emptyset} x^t = 0$ and $\prod_{t\in\emptyset} x^t = 1$.

II. Problem Setup and Preliminaries

A. Problem Setup

We study Problem (P) under the following assumptions.

Assumption 1 (On the optimization problem)**.**

a. Each $f_i : \mathbb{R}^n \to \mathbb{R}$ is proper, closed and L_i-Lipschitz differentiable;

b. F is bounded from below. □

Note that f_i need not be convex. We also make the blanket assumption that each agent i knows only its own f_i, but not $\sum_{j\neq i} f_j$. To state linear convergence, we will use the following extra condition on the objective function.

Assumption 2 (Strong convexity)**.** *Assumption 1(i) holds and, in addition, F is τ-strongly convex.* □

On the communication network: The communication network of the agents is modeled as a fixed, directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, I\}$ is the set of nodes (agents), and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges (communication links). If $(i, j) \in \mathcal{E}$, it means that agent i can send information to agent j. We assume that the digraph does not have self-loops. We denote by $\mathcal{N}_i^{\text{in}}$ the set of *in-neighbors* of node i, i.e., $\mathcal{N}_i^{\text{in}} \triangleq \{j \in \mathcal{V} \mid (j, i) \in \mathcal{E}\}$ while $\mathcal{N}_i^{\text{out}} \triangleq \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$ is the set of *out-neighbors* of agent i. We make the following standard assumption on the graph connectivity.

Assumption 3. *The graph $\mathcal{G}$ is strongly connected.* □

B. Preliminaries: The SONATA algorithm [34], [35]

The proposed asynchronous algorithmic framework builds on the synchronous SONATA algorithm, proposed in [34], [35] to solve (nonconvex) multi-agent optimization problems over time-varying digraphs. This is motivated by the fact that SONATA has the unique property of being provably applicable to both convex and nonconvex problems, and it achieves linear convergence when applied to strongly convex objectives F. We thus begin reviewing SONATA, tailored to (P); then we generalized it to the asynchronous setting (cf. Sec. IV).

Every agent controls and iteratively updates the tuple $(\mathbf{x}_i, \mathbf{y}_i, \mathbf{z}_i, \phi_i)$: $\mathbf{x}_i$ is agent i's copy of the shared variables $\mathbf{x}$ in (P); $\mathbf{y}_i$ acts as a local proxy of the sum-gradient ∇F; and $\mathbf{z}_i$ and ϕ_i are auxiliary variables instrumental to deal with communications over digraphs. Let $\mathbf{x}_i^k, \mathbf{z}_i^k$, ϕ_i^k, and $\mathbf{y}_i^k$ denote the value of the aforementioned variables at iteration $k \in \mathbb{N}_0$. The update of each agent i reads:

$$\mathbf{x}_i^{k+1} = \sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} w_{ij}\left(\mathbf{x}_j^k - \alpha^k\,\mathbf{y}_j^k\right), \tag{1}$$

$$\mathbf{z}_i^{k+1} = \sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} a_{ij}\mathbf{z}_j^k + \nabla f_i(\mathbf{x}_i^{k+1}) - \nabla f_i(\mathbf{x}_i^k), \tag{2}$$

$$\phi_i^{k+1} = \sum_{j\in\mathcal{N}_i^{\text{in}}\cup\{i\}} a_{ij}\phi_j^k, \tag{3}$$

$$\mathbf{y}_i^{k+1} = \mathbf{z}_i^{k+1}/\phi_i^{k+1}, \tag{4}$$

with $\mathbf{z}_i^0 = \mathbf{y}_i^0 = \nabla f_i(\mathbf{x}_i^0)$ and $\phi_i^0 = 1$, for all $i \in \mathcal{V}$. In (1), $\mathbf{y}_i^k$ is a local estimate of the average-gradient $(1/I)\sum_{i=1}^I \nabla f_i(\mathbf{x}_i^k)$. Therefore, every agent, first moves along the estimated gradient direction, generating $\mathbf{x}_i^k - \alpha^k\,\mathbf{y}_i^k$ (α^k is the step-size); and then performs a consensus step to force asymptotic agreement among the local variables $\mathbf{x}_i$. Steps (2)-(4) represent a perturbed-push-sum update, aiming at tracking the gradient $(1/I)\,\nabla F$ [28], [29], [35]. The weight-matrices $\mathbf{W} \triangleq (w_{ij})_{i,j=1}^I$ and $\mathbf{A} \triangleq (a_{ij})_{i,j=1}^I$ satisfy the following standard assumptions.

Assumption 4 (On the weight-matrices)**.** *The weight-matrices $\mathbf{W} \triangleq (w_{ij})_{i,j=1}^I$ and $\mathbf{A} \triangleq (a_{ij})_{i,j=1}^I$ satisfy (we will write $\mathbf{M} \triangleq (m_{ij})_{i,j=1}^I$ to denote either $\mathbf{A}$ or $\mathbf{W}$):*

a. $\exists\, \bar{m} > 0$ such that $m_{ii} \geq \bar{m}$, for all $i \in \mathcal{V}$; and $m_{ij} \geq \bar{m}$, for all $(j, i) \in \mathcal{E}$; $m_{ij} = 0$, otherwise;

b. $\mathbf{W}$ is row-stochastic, that is, $\mathbf{W}\,\mathbf{1} = \mathbf{1}$;

c. $\mathbf{A}$ is column-stochastic, that is, $\mathbf{A}^T\,\mathbf{1} = \mathbf{1}$; □

In [33], a special instance of SONATA, was proved to converges at an R-linear rate when F is strongly convex. This result was extended to constrained, nonsmooth (composite), distributed optimization in [36]. A natural question is whether SONATA works also in an asynchronous setting still converging at a linear rate. Naive asynchronization of the updates (1)-(4)–such as using uncoordinated activations and/or replacing instantaneous information with a delayed one–would not work. For instance, the tracking (2)-(4) calls for the invariance of the averages, i.e., $\sum_{i=1}^I \mathbf{z}_i^k = \sum_{i=1}^I \nabla f_i(\mathbf{x}^k)$, for all $k \in \mathbb{N}_0$. It is not difficult to check that any perturbation in (2)-e.g., in the form of delays or packet losses–puts in jeopardy this property.

To cope with the above challenges, a first step is robustifying the gradient tracking scheme. In Sec. III, we introduce P-ASY-SUM-PUSH–an asynchronous, perturbed, instance of the push-sum algorithm [37], which serves as a unified algorithmic framework to accomplish several tasks over digraphs in an asynchronous manner, such as solving the average consensus problem and tracking the average of agents' time-varying signals. Building on P-ASY-SUM-PUSH, in Sec. IV, we finally present the proposed distributed asynchronous optimization framework, termed ASY-SONATA.

III. Perturbed Asynchronous Sum-Push

We present P-ASY-SUM-PUSH; the algorithm was first introduced in our conference paper [1], which we refer to for

details on the genesis of the scheme and intuitions; here we directly introduce the scheme and study its convergence.

Consider an asynchronous setting wherein agents compute and communicate independently without coordination. Every agent i maintains state variables $\mathbf{z}_i$, ϕ_i, $\mathbf{y}_i$, along with the following auxiliary variables that are instrumental to deal with uncoordinated activations and delayed information: i) the cumulative-mass variables $\boldsymbol{\rho}_{ji}$ and σ_{ji}, with $j \in \mathcal{N}_i^{\text{out}}$, which capture the cumulative (sum) information generated by agent i up to the current time and to be sent to agent $j \in \mathcal{N}_i^{\text{out}}$; consequently, $\boldsymbol{\rho}_{ij}$ and σ_{ij} are received by i from its in-neighbors $j \in \mathcal{N}_i^{\text{in}}$; and ii) the buffer variables $\tilde{\boldsymbol{\rho}}_{ij}$ and $\tilde{\sigma}_{ij}$, with $j \in \mathcal{N}_i^{\text{in}}$, which store the information sent from $j \in \mathcal{N}_i^{\text{in}}$ to i and used by i in its last update. Values of these variables at iteration $k \in \mathbb{N}_0$ are denoted by the same symbols with the superscript "k". Note that, because of the asynchrony, each agent i might have outdated $\boldsymbol{\rho}_{ij}$ and σ_{ij}; $\boldsymbol{\rho}_{ij}^{k-d_j^k}$ (resp. $\sigma_{ij}^{k-d_j^k}$) is a delayed version of the current $\boldsymbol{\rho}_{ij}^k$ (resp. σ_{ij}^k) owned by j at time k, where $0 \leq d_j^k \leq D < \infty$ is the delay. Similarly, $\tilde{\boldsymbol{\rho}}_{ij}$ and $\tilde{\sigma}_{ij}$ might differ from the last information generated by j for i, because agent i might not have received that information yet (due to delays) or never will (due to packet losses).

The proposed asynchronous algorithm, P-ASY-SUM-PUSH, is summarized in Algorithm 1. A global iteration clock (not known to the agents) is introduced: $k \to k+1$ is triggered based upon the completion from one agent, say i^k, of the following actions. **(S.2):** agent i^k maintains a local variable $\tau_{i^k j}$, for each $j \in \mathcal{N}_{i^k}^{\text{in}}$, which keeps track of the "age" (generated time) of the $(\boldsymbol{\rho}, \sigma)$-variables that it has received from its in-neighbors and *already* used. If $k - d_j^k$ is larger than the current counter $\tau_{i^k j}^{k-1}$, indicating that the received $(\boldsymbol{\rho}, \sigma)$-variables are newer than those currently stored, agent i^k accepts $\boldsymbol{\rho}_{i^k j}^{k-d_j^k}$ and $\sigma_{i^k j}^{k-d_j^k}$, and updates $\tau_{i^k j}$ as $k - d_j^k$; otherwise, the variables will be discarded and $\tau_{i^k j}$ remains unchanged. Note that (5) can be performed without any coordination. It is sufficient that each agent attaches a time-stamp to its produced information reflecting it local timing counter. We describe next the other steps, assuming that new information has come in to agent i^k, that is, $\tau_{i^k j} = k - d_j^k$. **(S.3.1):** In (6), agent i^k builds the intermediate "mass" $\mathbf{z}_{i^k}^{k+\frac{1}{2}}$ based upon its current information $\mathbf{z}_{i^k}^k$ and $\tilde{\boldsymbol{\rho}}_{i^k j}^k$, and the (possibly) delayed one from its in-neighbors, $\boldsymbol{\rho}_{i^k j}^{k-d_j^k}$; and $\boldsymbol{\epsilon}^k \in \mathbb{R}^n$ is an exogenous perturbation (later this perturbation will be properly chosen to accomplish specific goals, see Sec. IV). Note that the way agent i^k forms its own estimates $\boldsymbol{\rho}_{i^k j}^{k-d_j^k}$ is *immaterial* to the description of the algorithm. The local buffer $\tilde{\boldsymbol{\rho}}_{i^k j}^k$ stores the value of $\boldsymbol{\rho}_{i^k j}$ that agent i^k used in its last update. Therefore, if the information in $\boldsymbol{\rho}_{i^k j}^{k-d_j^k}$ is not older than the one in $\tilde{\boldsymbol{\rho}}_{i^k j}^k$, the difference $\boldsymbol{\rho}_{i^k j}^{k-d_j^k} - \tilde{\boldsymbol{\rho}}_{i^k j}^k$ in (6) will capture the sum of the $a_{i^k j}\mathbf{z}_j$'s that have been generated by $j \in \mathcal{N}_{i^k}^{\text{in}}$ for i^k up until $k - d_j^k$ and not used by agent i^k yet. For instance, in a synchronous setting, one would have $\boldsymbol{\rho}_{i^k j}^k - \tilde{\boldsymbol{\rho}}_{i^k j}^k = a_{i^k j}\mathbf{z}_j^{k+\frac{1}{2}}$. **(S.3.2):** the generated $\mathbf{z}_{i^k}^{k+\frac{1}{2}}$ is "pushed back" to agent i^k itself and its out-neighbors. Specifically, out of the total mass $\mathbf{z}_{i^k}^{k+\frac{1}{2}}$ generated, agent i^k gets $a_{ii}\,\mathbf{z}_i^{k+\frac{1}{2}}$, determining the update $\mathbf{z}_i^k \to \mathbf{z}_i^{k+1}$ while the remaining is allocated to the agents $j \in \mathcal{N}_{i^k}^{\text{out}}$, with $\boldsymbol{a}_{ji^k}\,\mathbf{z}_{i^k}^{k+\frac{1}{2}}$ cumulating to the mass buffer $\boldsymbol{\rho}_{ji^k}^k$ and generating the update $\boldsymbol{\rho}_{ji^k}^k \to \boldsymbol{\rho}_{ji^k}^{k+1}$, to be sent to agent j. **(S.3.3):** each local buffer variable $\tilde{\boldsymbol{\rho}}_{i^k j}^k$ is updated to account for the use of new information from $j \in \mathcal{N}_{i^k}^{\text{in}}$. The final information is then read on the $\mathbf{y}$-variables [cf. **(S.3.4)**].

Algorithm 1 P-ASY-SUM-PUSH (Global View)

Data: $\mathbf{z}_i^0 \in \mathbb{R}^n$, $\phi_i^0 = 1$, $\tilde{\boldsymbol{\rho}}_{ij}^0 = 0$, $\tilde{\sigma}_{ij}^0 = 0$, $\tau_{ij}^{-1} = -D$, for all $j \in \mathcal{N}_i^{\text{in}}$ and $i \in \mathcal{V}$; $\sigma_{ij}^t = 0$ and $\boldsymbol{\rho}_{ij}^t = 0$, for all $t = -D, \ldots, 0$; and $\{\boldsymbol{\epsilon}^k\}_{k\in\mathbb{N}_0}$. Set $k = 0$.

While: a termination criterion is not met **do**

(S.1) Pick $(i^k, \mathbf{d}^k)$, with $\mathbf{d}^k \triangleq (d_j^k)_{j\in\mathcal{N}_{i^k}^{\text{in}}}$;

(S.2) Set (purge out the old information):

$$\tau_{i^k j}^k = \max\left(\tau_{i^k j}^{k-1}, k - d_j^k\right), \quad \forall j \in \mathcal{N}_{i^k}^{\text{in}}; \tag{5}$$

(S.3) Update the variables performing

- (S.3.1) **Sum step:**

$$\mathbf{z}_{i^k}^{k+\frac{1}{2}} = \mathbf{z}_{i^k}^k + \sum_{j\in\mathcal{N}_{i^k}^{\text{in}}} \left(\boldsymbol{\rho}_{i^k j}^{\tau_{i^k j}^k} - \tilde{\boldsymbol{\rho}}_{i^k j}^k\right) + \boldsymbol{\epsilon}^k \tag{6}$$

$$\phi_{i^k}^{k+\frac{1}{2}} = \phi_{i^k}^k + \sum_{j\in\mathcal{N}_{i^k}^{\text{in}}} \left(\sigma_{i^k j}^{\tau_{i^k j}^k} - \tilde{\sigma}_{i^k j}^k\right)$$

- (S.3.2) **Push step:**

$$\begin{aligned} \mathbf{z}_{i^k}^{k+1} &= a_{i^k i^k}\,\mathbf{z}_{i^k}^{k+\frac{1}{2}}, \quad \phi_{i^k}^{k+1} = a_{i^k i^k}\,\phi_{i^k}^{k+\frac{1}{2}} \\ \boldsymbol{\rho}_{ji^k}^{k+1} &= \boldsymbol{\rho}_{ji^k}^k + a_{ji^k}\,\mathbf{z}_{i^k}^{k+\frac{1}{2}}, \\ \sigma_{ji^k}^{k+1} &= \sigma_{ji^k}^k + a_{ji^k}\,\phi_{i^k}^{k+\frac{1}{2}}, \quad \forall j \in \mathcal{N}_{i^k}^{\text{out}} \end{aligned} \tag{7}$$

- (S.3.3) **Mass-Buffer update:**

$$\tilde{\boldsymbol{\rho}}_{i^k j}^{k+1} = \boldsymbol{\rho}_{i^k j}^{\tau_{i^k j}^k}, \quad \tilde{\sigma}_{i^k j}^{k+1} = \sigma_{i^k j}^{\tau_{i^k j}^k}, \quad \forall j \in \mathcal{N}_{i^k}^{\text{in}} \tag{8}$$

- (S.3.4) **Set:** $\mathbf{y}_{i^k}^{k+1} = \mathbf{z}_{i^k}^{k+1}/\phi_{i^k}^{k+1}$.

(S.4) Untouched state variables shift to state $k+1$ while keeping the same value; $k \leftarrow k+1$.

Remark 5. *(Global view description)* Note that each agent's update is fully defined, once i^k and $\mathbf{d}^k$ are given. The selection $(i^k, \mathbf{d}^k)$ in **(S.1)** is not performed by anyone; it is instead an *a-posteriori* description of agents' actions: All agents act asynchronously and continuously; the agent completing the "push" step and updating its own variables triggers *retrospectively* the iteration counter $k \to k+1$ and determines the pair $(i^k, \mathbf{d}^k)$ along with all quantities involved in the other steps. Differently from most of the current literature, this "global view" description of the agents' actions allows us to abstract from specific computation-communication protocols and asynchronous modus operandi and captures by a unified model a gamut of asynchronous schemes.

Convergence is given under the following assumptions.

Assumption 6 (On the asynchronous model)**.** *Suppose:*

a. $\exists\ 0 < T < \infty$ *such that* $\cup_{t=k}^{k+T-1} i^t = \mathcal{V}$*, for all* $k \in \mathbb{N}_0$*;*
b. $\exists\ 0 < D < \infty$ *such that* $0 \leq d_j^k \leq D$*, for all* $j \in \mathcal{N}_{i^k}^{in}$ *and* $k \in \mathbb{N}_0$. □

The next theorem studies convergence of P-ASY-SUM-PUSH, establishing geometric decay of the error $\|\mathbf{y}_i^k - (1/I)\cdot \mathbf{m}_z^k\|$, even in the presence of unknown (bounded) perturbations, where $\mathbf{m}_z^k \triangleq \sum_{i=1}^I \mathbf{z}_i^k + \sum_{(j,i)\in\mathcal{E}}(\boldsymbol{\rho}_{ij}^k - \tilde{\boldsymbol{\rho}}_{ij}^k)$ represents the "total mass" of the system at iteration k.

Theorem 7. *Let* $\{\mathbf{y}^k \triangleq [\mathbf{y}_1^k,\dots,\mathbf{y}_I^k]^\top,\ \mathbf{z}^k \triangleq [\mathbf{z}_1^k,\dots,\mathbf{z}_I^k]^\top, (\boldsymbol{\rho}_{ij}^k, \tilde{\boldsymbol{\rho}}_{ij}^k)_{(j,i)\in\mathcal{E}}\}_{k\in\mathbb{N}_0}$ *be the sequence generated by Algorithm 1, under Assumption 3, 6, and with* $\mathbf{A} \triangleq (a_{ij})_{i,j=1}^I$ *satisfying Assumption 4 (i),(iii). Define* $K_1 \triangleq (2\,I-1)\cdot T + I\cdot D$*. There exist constants* $\rho \in (0,1)$ *and* $C_1 > 0$*, such that*

$$\left\|\mathbf{y}_i^{k+1} - (1/I)\cdot \mathbf{m}_z^{k+1}\right\| \leq C_1\left(\rho^k \left\|\mathbf{z}^0\right\| + \sum_{l=0}^{k}\rho^{k-l}\left\|\boldsymbol{\epsilon}^l\right\|\right), \tag{9}$$

for all $i \in \mathcal{V}$ *and* $k \geq K_1 - 1$.

Furthermore, $\mathbf{m}_z^k = \sum_{i=1}^I \mathbf{z}_i^0 + \sum_{t=0}^{k-1}\boldsymbol{\epsilon}^t$.

Proof. See Sec. VI. □

Discussion: Several comments are in order.

1) On the asynchronous model: Algorithm 1 captures a gamut of asynchronous *parallel* schemes and architectures, through the mechanism of generation of $(i^k, \mathbf{d}^k)$. Assumption 6 on $(i^k, \mathbf{d}^k)$ is quite mild: (a) controls the frequency of the updates whereas (b) limits the age of the old information used in the computations; they can be easily enforced in practice. For instance, (a) is readily satisfied if each agent wakes up and performs an update whenever some independent internal clock ticks or it is triggered by some of the neighbors; (b) imposes conditions on the frequency and quality of the communications: information used by each agent cannot become infinitely old, implying that successful communications must occur sufficiently often. This however does not enforce any specific protocol on the activation/idle time/communication. For instance, i) agents need not perform the actions in Algorithm 1 sequentially or inside the same activation round; or ii) executing the "push" step does not mean that agents must broadcast their new variables in the same activation; this would just incur a delay (or packet loss) in the communication.

Note that the time-varying nature of the delays $\mathbf{d}^k$ permits to model also packet losses, as detailed next. Suppose that at iteration k_1 agent j sends its current ρ,σ-variables to its out-neighbor ℓ and they get lost; and let k_2 be the subsequent iteration when j updates again. Let t be the first iteration after k_1 when agent ℓ performs its update; it will use information from j such that $t - d_j^t \notin [k_1+1, k_2]$, for some $d_j^t \leq D < \infty$. If $t - d_j^t < k_1 + 1$, no newer information from j has been used by ℓ; otherwise $t - d_j^t \geq k_2 + 1$ (implying $k_2 < t$), meaning that agent ℓ has used information not older than $k_2 + 1$.

2) Comparison with [8], [30], [38]: The use of counter variables [such as $(\boldsymbol{\rho}, \sigma, \tilde{\boldsymbol{\rho}}, \tilde{\sigma})$-variables in our scheme] was first introduced in [30] to design a synchronous average consensus algorithm robust to packet losses. In [38], this scheme was extended to deal with uncoordinated (deterministic) agents' activations whereas [8] built on [38] to design, in the same setting, a distributed Newton-Raphson algorithm. There are important differences between P-ASY-SUM-PUSH and the aforementioned schemes, namely: i) none of them can deal with *delays but packet losses*; ii) [30] is *synchronous*; and iii) [8], [38] are not *parallel* schemes, as at each iteration only one agent is allowed to wake up and transmit information to its neighbors. For instance, [8], [38] cannot model synchronous parallel (Jacobi) updates. Hence, the convergence analysis of P-ASY-SUM-PUSH calls for a new line of proof, as introduced in Sec. VI.

3) Beyond average consensus: By choosing properly the perturbation signal $\boldsymbol{\epsilon}^k$, P-ASY-SUM-PUSH can solve different problems. Some examples are discussed next.

(i) Error free: $\boldsymbol{\epsilon}^k = \mathbf{0}$. P-ASY-SUM-PUSH solves the average consensus problem and (9) reads

$$\left\|\mathbf{y}_i^{k+1} - (1/I)\cdot\sum_{i=1}^I \mathbf{z}_i^0\right\| \leq C_1\,\rho^k\,\left\|\mathbf{z}^0\right\|.$$

(ii) Vanishing error: $\lim_{k\to\infty}\|\boldsymbol{\epsilon}^k\| = 0$. Using [29, Lemma 7(a)], (9) reads $\lim_{k\to\infty}\|\mathbf{y}_i^{k+1} - \mathbf{m}_z^{k+1}\| = 0$.

(iii) Asynchronous tracking. Each agent i owns a (time-varying) signal $\{\mathbf{u}_i^k\}_{k\in\mathbb{N}_0}$; the average tracking problem consists in asymptotically track the average signal $\bar{\mathbf{u}}^k \triangleq (1/I)\cdot\sum_{i=1}^I \mathbf{u}_i^k$, that is,

$$\lim_{k\to\infty}\left\|\mathbf{y}_i^{k+1} - \bar{\mathbf{u}}^{k+1}\right\| = 0, \quad \forall i \in \mathcal{V}. \tag{10}$$

Under mild conditions on the signal, this can be accomplished in a distributed and asynchronous fashion, using P-ASY-SUM-PUSH, as formalized next.

Corollary 7.1. *Consider, the following setting in* P-ASY-SUM-PUSH*:* $\mathbf{z}_i^0 = \mathbf{u}_i^0$*, for all* $i \in \mathcal{V}$*;* $\boldsymbol{\epsilon}^k = \mathbf{u}_{i^k}^{k+1} - \tilde{\mathbf{u}}_{i^k}^k$*, with*

$$\tilde{\mathbf{u}}_i^{k+1} = \begin{cases}\mathbf{u}_i^{k+1} & \text{if } i = i^k;\\ \tilde{\mathbf{u}}_i^k & \text{otherwise;}\end{cases} \qquad \tilde{\mathbf{u}}_i^0 = \mathbf{u}_i^0;$$

Then (9) *holds, with* $\mathbf{m}_z^{k+1} = \sum_{i=1}^I \tilde{\mathbf{u}}_i^{k+1}$*. Furthermore, if* $\lim_{k\to\infty}\sum_{i=1}^I\left\|\mathbf{u}_i^{k+1} - \mathbf{u}_i^k\right\| = 0$*, then* (10) *holds.*

Proof. See the technical report [2, Appendix E]. □

This instance of P-ASY-SUM-PUSH will be used in Sec. IV to perform asynchronous gradient tracking.

Remark 8 (Asynchronous average consensus)**.** To the best of our knowledge, the error-free instance of the P-ASY-SUM-PUSH discussed above is the first (stepsize-free) scheme that provably solves the *average* consensus problem at a linear rate, under the general asynchronous model described by Assumption 6. In fact, the existing asynchronous consensus schemes [31] [32] achieve an agreement among the agents' local variables whose value is not in general the average of their initial values, but instead some *unknown* function of them and the asynchronous modus operandi of the agents. Related to the P-ASY-SUM-PUSH is the ra-AC algorithm in [38], which enjoys the same convergence property but under a more restrictive and specific asynchronous model (no delays but packet losses and single-agent activation per iteration).

IV. Asynchronous SONATA (ASY-SONATA)

We are ready now to introduce our distributed asynchronous algorithm–ASY-SONATA. The algorithm combines SONATA (cf. Sec. II-B) with P-ASY-SUM-PUSH (cf. Sec. III), the latter replacing the synchronous tracking scheme (2)-(4). The "global view" of the scheme is given in Algorithm 2.

Algorithm 2 ASY-SONATA (Global View)

Data: For all agent i and $\forall j \in \mathcal{N}_i^{\text{in}}$, $\mathbf{x}_i^0 \in \mathbb{R}^n$, $\mathbf{z}_i^0 = \nabla f_i(\mathbf{x}_i^0)$, $\phi_i^0 = 1$, $\tilde{\boldsymbol{\rho}}_{ij}^0 = 0$, $\tilde{\sigma}_{ij}^0 = 0$, $\tau_{ij}^{-1} = -D$. And for $t = -D, -D+1, \ldots, 0$, $\boldsymbol{\rho}_{ij}^t = 0$, $\sigma_{ij}^t = 0$, $\mathbf{v}_i^t = 0$. Set $k = 0$.

While: a termination criterion is not met **do**

(S.1) Pick $(i^k, \mathbf{d}^k)$;

(S.2) Set:

$$\tau_{i^k j}^k = \max(\tau_{i^k j}^{k-1}, k - d_j^k), \quad \forall j \in \mathcal{N}_{i^k}^{\text{in}}.$$

(S.3) Local Descent:

$$\mathbf{v}_{i^k}^{k+1} = \mathbf{x}_{i^k}^k - \gamma^k \mathbf{z}_{i^k}^k. \tag{11}$$

(S.4) Consensus:

$$\mathbf{x}_{i^k}^{k+1} = w_{i^k i^k} \mathbf{v}_{i^k}^{k+1} + \sum_{j \in \mathcal{N}_{i^k}^{\text{in}}} w_{i^k j} \mathbf{v}_j^{\tau_{i^k j}^k}.$$

(S.5) Gradient Tracking:

- (S.5.1) **Sum step:**

$$\mathbf{z}_{i^k}^{k+\frac{1}{2}} = \mathbf{z}_{i^k}^k + \sum_{j \in \mathcal{N}_{i^k}^{\text{in}}} \left(\boldsymbol{\rho}_{i^k j}^{\tau_{i^k j}^k} - \tilde{\boldsymbol{\rho}}_{i^k j}^k \right) + \nabla f_{i^k}(\mathbf{x}_{i^k}^{k+1}) - \nabla f_{i^k}(\mathbf{x}_{i^k}^k)$$

- (S.5.2) **Push step:**

$$\mathbf{z}_{i^k}^{k+1} = a_{i^k i^k} \mathbf{z}_{i^k}^{k+\frac{1}{2}},$$
$$\boldsymbol{\rho}_{j i^k}^{k+1} = \boldsymbol{\rho}_{j i^k}^k + a_{j i^k} \mathbf{z}_{i^k}^{k+\frac{1}{2}}, \quad \forall j \in \mathcal{N}_{i^k}^{\text{out}}$$

- (S.5.3) **Mass-Buffer update:**

$$\tilde{\boldsymbol{\rho}}_{i^k j}^{k+1} = \boldsymbol{\rho}_{i^k j}^{\tau_{i^k j}^k}, \quad \forall j \in \mathcal{N}_{i^k}^{\text{in}}$$

(S.6) Untouched state variables shift to state $k+1$ while keeping the same value; $k \leftarrow k+1$.

In ASY-SONATA, agents continuously and with no coordination perform: i) their local computations [cf. **(S.3)**], possibly using an out-of-sync estimate $\mathbf{z}_{i^k}^k$ of the average gradient; in (11), γ^k is a step-size (to be properly chosen); ii) a consensus step on the $\mathbf{x}$-variables, using possibly outdated information $\mathbf{v}_j^{\tau_{i^k j}^k}$ from their in-neighbors [cf. **(S.4)**]; and iii) gradient tracking [cf. **(S.5)**] to update the local estimate $\mathbf{z}_{i^k}^k$, based on the current cumulative mass variables $\boldsymbol{\rho}_{i^k j}^{\tau_{i^k j}^k}$, and buffer variables $\tilde{\boldsymbol{\rho}}_{i^k j}^k$, $j \in \mathcal{N}_{i^k}^{\text{in}}$.

Note that in Algorithm 1, the tracking variable $\mathbf{y}_{i^k}^{k+1}$ is obtained rescaling $\mathbf{z}_{i^k}^{k+1}$ by the factor $1/\phi_{i^k}^{k+1}$. In Algorithm 2, we absorbed the scaling $1/\phi_{i^k}^{k+1}$ in the step size and use directly $\mathbf{z}_{i^k}^{k+1}$ as a proxy of the average gradient, eliminating thus the ϕ-variables (and the related σ-, $\tilde{\sigma}$-variables). Also, for notational simplicity and without loss of generality, we assumed that the $\mathbf{v}$- and $\boldsymbol{\rho}$- variables are subject to the same delays (e.g., they are transmitted within the same packet); same convergence results hold if different delays are considered.

We study now convergence of the scheme, under a constant step-size or diminishing, uncoordinated ones.

A. Constant Step-size

Theorem 9 below establishes *linear* convergence of ASY-SONATA when F is strongly convex.

Theorem 9 (Geometric convergence)**.** *Consider (P) under Assumption 2, and let $\mathbf{x}^\star$ denote its unique solution. Let $\{(\mathbf{x}_i^k)_{i=1}^I\}_{k \in \mathbb{N}_0}$ be the sequence generated by Algorithm 2, under Assumption 3, 6, and with weight-matrices $\mathbf{W}$ and $\mathbf{A}$ satisfying Assumption 4. Then, there exists a constant $\bar{\gamma}_1 > 0$ [cf. (46)] such that if $\gamma^k \equiv \gamma \leq \bar{\gamma}_1$, it holds*

$$M_{sc}(\mathbf{x}^k) \triangleq \|\mathbf{x}^k - \mathbf{1}_I \otimes \mathbf{x}^\star\| = \mathcal{O}(\lambda^k), \tag{12}$$

with $\lambda \in (0,1)$ given by

$$\lambda = \begin{cases} 1 - \frac{\tau \bar{m}^{2K_1} \gamma}{2} & \text{if } \gamma \in (0, \hat{\gamma}_1], \\ \rho + \sqrt{J_1 \gamma} & \text{if } \gamma \in (\hat{\gamma}_1, \hat{\gamma}_2), \end{cases} \tag{13}$$

where $\hat{\gamma}_1$ and $\hat{\gamma}_2$ are some constants strictly smaller than $\bar{\gamma}_1$, and $J_1 \triangleq (1-\rho)^2/\hat{\gamma}_2$.

Proof. See Sec. VII. □

When F is convex (resp. nonconvex), we introduce the following merit function to measure the progresses of the algorithm towards optimality (resp. stationarity) and consensus:

$$M_F(\mathbf{x}^k) \triangleq \max\{\left\|\nabla F(\bar{\mathbf{x}}^k)\right\|^2, \left\|\mathbf{x}^k - \mathbf{1}_I \otimes \bar{\mathbf{x}}^k\right\|^2\}, \tag{14}$$

where $\mathbf{x}^k \triangleq [\mathbf{x}_1^{k\top}, \cdots, \mathbf{x}_I^{k\top}]^\top$ and $\bar{\mathbf{x}}^k \triangleq (1/I) \cdot \sum_{i=1}^I \mathbf{x}_i^k$. Note that M_F is a valid merit function, since it is continuous and $M_F(\mathbf{x}) = 0$ if and only if all $\mathbf{x}_i$'s are consensual and optimal (resp. stationary solutions).

Theorem 10 (Sublinear convergence)**.** *Consider (P) under Assumption 1 (thus possibly nonconvex). Let $\{(\mathbf{x}_i^k)_{i=1}^I\}_{k \in \mathbb{N}_0}$ be the sequence generated by Algorithm 2, in the same setting of Theorem 9. Given $\delta > 0$, let T_δ be the first iteration $k \in \mathbb{N}_0$ such that $M_F(\mathbf{x}^k) \leq \delta$. Then, there exists a $\bar{\gamma}_2 > 0$ [cf. (53)], such that if $\gamma^k \equiv \gamma \leq \bar{\gamma}_2$, $T_\delta = \mathcal{O}(1/\delta)$. The values of the above constants is given in the proof.*

Proof. See Sec. VIII. □

Theorem 9 states that consensus and optimization errors of the sequence generated by ASY-SONATA vanish at a linear rate. We are not aware of any other scheme enjoying such a property in such a distributed, asynchronous computing environment. For general, possibly nonconvex instances of Problem (P), Theorem 10 shows that both consensus and optimization errors of the sequence generated by ASY-SONATA vanish at $\mathcal{O}(1/\delta)$ sublinear rate.

The choice of a proper stepsize calls for the estimates of $\bar{\gamma}_1$ and $\bar{\gamma}_2$ in Theorems 9 and 10, which depend on the following quantities: the optimization parameters L_i (Lipschitz constants of the gradients) and τ (strongly convexity constant), the

network connectivity parameter ρ, and the constants D and T due to the asynchrony (cf. Assumption 6). Notice that the dependence of the stepsize on L_i, τ, and ρ is common to all the existing distributed synchronous algorithms and so is that on T and D to (even centralized) asynchronous algorithms [3]. While L_i, τ, and ρ can be acquired following approaches discussed in the literature (see, e.g., [33, Remark 4]), it is less clear how to estimate D and T, as they are related to the asynchronous model, generally not known to the agents. As an example, we address this question considering the following fairly general model for the agents' activations and asynchronous communications. Suppose that the length of any time window between consecutive "push" steps of any agent belongs to $[p_{\min}, p_{\max}]$, for some $p_{\max} \geq p_{\min} > 0$, and one agent always sends out its updated information immediately after the completion of its "push" step. The traveling time of each packet is at most D^{tv}. Also, at least one packet is successfully received every D^{ls} successive one-hop communications. Note that there is a vast literature on how to estimate D^{tv} and D^{ls}, based upon the specific channel model under consideration; see, e.g., [39], [40]. In this setting, it is not difficult to check that one can set $T = (I-1)\lceil p_{\max}/p_{\min}\rceil + 1$ and $D = I\lceil D^{\text{tv}}/p_{\min}\rceil D^{\text{ls}}$. To cope with the issue of estimating $\bar\gamma_1$ and $\bar\gamma_2$, in the next section we show how to employ in ASY-SONATA diminishing, uncoordinated stepsizes.

B. Uncoordinated diminishing step-sizes

The use of a diminishing stepsize shared across the agents is quite common in synchronous distributed algorithms. However, it is not clear how to implement such option in an asynchronous setting, without enforcing any coordination among the agents (they should know the global iteration counter k). In this section, we provide for the first time a solution to this issue. Inspired by [41], our model assumes that each agent, *independently* and with *no coordination* with the others, draws the step-size from a local sequence $\{\alpha^t\}_{t\in\mathbb{N}_0}$, according to its local clock. The sequence $\{\gamma^k\}_{k\in\mathbb{N}_0}$ in (11) will be thus the result of the "uncoordinated samplings" of the local out-of-sync sequences $\{\alpha^t\}_{t\in\mathbb{N}_0}$. The next theorem shows that in this setting, ASY-SONATA converges at a sublinear rate for both convex and nonconvex objectives.

Theorem 11. *Consider Problem (P) under Assumption 1 (thus possibly nonconvex). Let $\{(\mathbf{x}_i^k)_{i=1}^I\}_{k\in\mathbb{N}_0}$ be the sequence generated by Algorithm 2, in the same setting of Theorem 9, but with the agents using a local step-size sequence $\{\alpha^t\}_{t\in\mathbb{N}_0}$ satisfying $\alpha^t \downarrow 0$ and $\sum_{t=0}^\infty \alpha^t = \infty$. Given $\delta > 0$, let T_δ be the first iteration $k \in \mathbb{N}_0$ such that $M_F(\mathbf{x}^k) \leq \delta$. Then*

$$T_\delta \leq \inf\Big\{k \in \mathbb{N}_0 \,\Big|\, \sum_{t=0}^{k} \gamma^t \geq c/\delta\Big\}, \tag{15}$$

where c is a positive constant.

Proof. See Sec. VIII. □

V. Numerical Results

We test ASY-SONATA on a strongly convex and nonconvex instance of Problem (P) over digraphs, namely: the regularized logistic regression (RLR) and the robust classification (RC) problems. Both formulations can be abstracted as:

$$\min_{\mathbf{x}} \frac{1}{|\mathcal{D}|}\sum_{i=1}^{I}\sum_{j\in\mathcal{D}_i} V(y_j\cdot \ell_{\mathbf{x}}(\mathbf{u}_j)) + \lambda\,\|\nabla \ell_{\mathbf{x}}(\cdot)\|_2^2\,, \tag{16}$$

where $\mathcal{D} = \cup_{i=1}^I \mathcal{D}_i$ is the set of indices of the data distributed across the agents, with agent i owning $\mathcal{D}_i$, and $\mathcal{D}_i \cap \mathcal{D}_l = \emptyset$, for all $i \neq l$; $\mathbf{u}_j$ and $y_j \in \{-1, 1\}$ are the feature vector and associated label of the j-th sample in $\mathcal{D}$; $\ell_{\mathbf{x}}(\cdot)$ is a linear function, parameterized by $\mathbf{x}$; and V is the loss function. More specifically, if the RLR problem is considered, V reads $V(r) = \log(1+e^{-r})$ while for the RC problem, we have [42]

$$V(r) = \begin{cases} 0, & \text{if } r > 1; \\ \frac{1}{4}r^3 - \frac{3}{4}r + \frac{1}{2}, & \text{if } -1 \leq r \leq 1; \\ 1, & \text{if } r < -1. \end{cases}$$

Data: We use the following data sets for the RLR and RC problems. (RLR): We set $\ell_{\mathbf{x}}(\mathbf{u}) = \mathbf{x}^\top\mathbf{u}$, $n = 100$, each $|\mathcal{D}_i| = 20$, and $\lambda = 0.01$. The underlying statistical model is the following: We generated the ground truth $\widehat{\mathbf{x}}$ with i.i.d. $\mathcal{N}(0,1)$ components; each training pair $(\mathbf{u}_j, y_j)$ is generated independently, with each element of $\mathbf{u}_j$ being i.i.d. $\mathcal{N}(0,1)$ and y_j is set as 1 with probability $1/\left(1+\exp(-\ell_{\widehat{\mathbf{x}}}(\mathbf{u}_j))\right)$, and -1 otherwise. (RC): We use the Cleveland Heart Disease Data set with 14 features [43], preprocessing it by deleting observations with missing entries, scaling features between 0-1, and distributing the data to agents evenly. We set $\ell_{\mathbf{x}}(\mathbf{u}) = \mathbf{e}_{15}^\top\mathbf{x} + \sum_{d=1}^{14}\mathbf{e}_d^\top\mathbf{x}\,\mathbf{e}_d^\top\mathbf{u}$. **Network model:** We simulated a digraph of $I = 30$ agents. Each agent has 7 out-neighbors; one of them belongs to a directed cycle connecting all the agents while the other 6 are picked uniformly at random. One row and one column stochastic matrix with uniform weights are generated. **Asynchronous model:** a) Activation lists are generated by concatenating *random rounds*. To generate one round, we first sample its length uniformly from the interval $[I, T]$, with $T = 90$. Within a round, we first have each agent appearing exactly once and then sample agents uniformly for the remaining spots. Finally a random shuffle of the agents order is performed on each round; b) Each transmitted message has (integer) traveling time which is sampled uniformly from the interval $[0, D^{\text{tv}}]$, with $D^{\text{tv}} = 90$.

We compare the performance of our algorithm with AsySubPush [44] and AsySPA [45], which appeared online during the revision process of our paper. AsySubPush and AsySPA differ from ASY-SONATA in the following aspects: i) they do not employ any gradient tracking mechanism; ii) they cannot handle packet losses and purge out old information from the system (information is used as it is received); iii) when F is strongly convex, they provably converge at *sublinear* rate; and iv) they cannot handle nonconvex F. The step sizes of all algorithms are manually tuned to obtain the best practical performance. We run two instances of ASY-SONATA, one employing a constant step size $\gamma = 0.4$ and the other one using the diminishing step size rule $\alpha^{t+1} = \alpha^t(1 - 0.001\cdot\alpha^t)$, where $\alpha^0 = 0.5$ and t is the local iteration counter. For AsySubPush (resp. AsySPA) we set, for each agent i, $\alpha_i = 0.0001$ (resp. $\rho(k) = c/\sqrt{k}$ with $c = 0.01$) in RLC and $\alpha_i = 0.00001$ (resp. $\rho(k) = c/\sqrt{k}$ with $c = 0.001$) in RC. The result is averaged over 20 Monte Carlo experiments with different di-

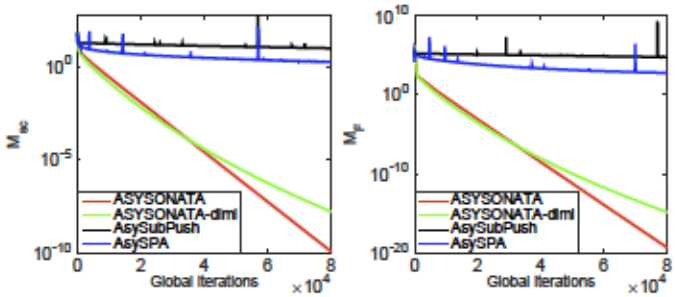


Figure 1. L: regularized logistic regression; R: robust classification.

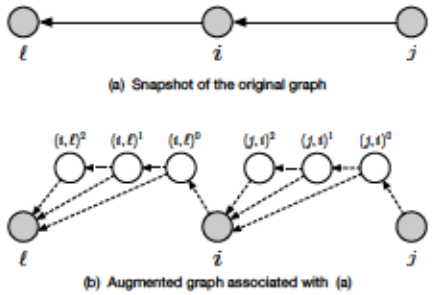


Figure 2. Example of augmented graph, when the maximum delay $D = 2$; three noncomputing agents are added for each edge $(j,i) \in \mathcal{E}$.

graph instances, and is presented in Fig. 1; for each algorithm, we plot the merit functions M_{sc} (left panel) and M_{F} (right panel) evaluated in the generated trajectory versus the global iteration counter k. Consistently with the convergence theory, ASY-SONATA with a constant step size exhibits a linear convergence rate. Also, ASY-SONATA outperforms the other two algorithms; this is mainly due to i) the presence in ASY-SONATA of an asynchronous gradient tracking mechanism which provides, at each iteration, a better estimate of ∇F; and ii) the possibility in ASY-SONATA to discard old information when received after a newer one [cf. (5)].

VI. Convergence Analysis of P-ASY-SUM-PUSH

We prove Theorem 7; we assume $n = 1$, without loss of generality. The proof is organized in the following two steps. **Step 1:** We first reduce the asynchronous agent system to a synchronous "augmented" one with no delays. This will be done adding virtual agents to the graph $\mathcal{G}$ along with their state variables, so that P-ASY-SUM-PUSH will be rewritten as a (synchronous) perturbed push-sum algorithm on the augmented graph. While this idea was first explored in [30], [31], there are some important differences between the proposed enlarged systems and those used therein, see Remark 13. **Step 2:** We conclude the proof establishing convergence of the perturbed push-sum algorithm built in Step 1.

A. Step 1: Reduction to a synchronous perturbed push-sum

1) The augmented graph: We begin constructing the augmented graph–an enlarged agent system obtained adding virtual agents to the original graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Specifically, we associate to each edge $(j,i) \in \mathcal{E}$ an ordered set of virtual nodes (agents), one for each of the possible delay values, denoted with a slight abuse of notation by $(j,i)^0, (j,i)^1, \dots, (j,i)^D$; see Fig. 2. Roughly speaking, these virtual nodes store the "information on fly" based upon its associated delay, that is, the information that has been generated by $j \in \mathcal{N}_i^{\text{in}}$ for i but not used (received) by i yet. Adopting the terminology in [31], nodes in the original graph $\mathcal{G}$ are termed *computing agents* while the virtual nodes will be called *noncomputing agents*. With a slight abuse of notation, we define the set of computing and noncomputing agents as $\widehat{\mathcal{V}} \triangleq \mathcal{V} \cup \{(i,j)^d \,|\, (i,j) \in \mathcal{E},\, d = 0,1,\dots,D\}$, and its cardinality as $S \triangleq \left|\widehat{\mathcal{V}}\right| = (I + (D+1)\,|\mathcal{E}|)$. We now identify the neighbors of each agent in this augmented systems. Computing agents no longer communicate among themselves; each $j \in \mathcal{V}$ can only send information to the noncomputing nodes $(j,i)^0$, with $i \in \mathcal{N}_j^{\text{out}}$. Each noncomputing agent $(j,i)^d$ can either send information to the next noncomputing agent, that is $(j,i)^{d+1}$ (if any), or to the computing agent i; see Fig. 2(b).

To describe the information stored by the agents in the augmented system at each iteration, let us first introduce the following quantities: $\mathcal{T}_i \triangleq \left\{k \,\middle|\, i^k = i,\, k \in \mathbb{N}_0\right\}$ is the set of global iteration indices at which the computing agent $i \in \mathcal{V}$ wakes up; and, given $k \in \mathbb{N}_0$, let $\mathcal{T}_i^k \triangleq \left\{t \in \mathcal{T}_i \,\middle|\, t \le k\right\}$. It is not difficult to conclude from (7) and (8) that

$$\rho_{ij}^k = \sum_{t \in \mathcal{T}_j^{k-1}} a_{ij} z_j^{t+1/2} \text{ and } \tilde{\rho}_{ij}^k = \rho_{ij}^{\tau_{ij}^{k-1}}, \quad (j,i) \in \mathcal{E}. \tag{17}$$

At iteration $k = 0$, every computing agent i stores z_i^0, whereas the values of the noncomputing agents are initialized to 0. At the beginning of iteration k, every computing agent i will store z_i^k whereas every noncomputing agent $(j,i)^d$, with $0 \le d \le D-1$, stores the mass $a_{ij} z_j$ (if any) generated by j for i at iteration $k-d-1$ (thus $k-d-1 \in \mathcal{T}_j^{k-1}$), i.e., $a_{ij} z_j^{k-(d+1)+1/2}$ (cf. Step 3.2), and not been used by i yet (thus $k-d > \tau_{ij}^{k-1}$); otherwise it stores 0. Formally, we have

$$z_{(j,i)^d}^k \triangleq a_{ij} z_j^{t+1/2} \cdot \mathbb{1}\left[t = k-d-1 \in \mathcal{T}_j^{k-1} \;\&\; t+1 > \tau_{ij}^{k-1}\right]. \tag{18}$$

The virtual node $(j,i)^D$ cumulates all the masses $a_{ij} z_j^{k-(d+1)+1/2}$ with $d \ge D$, not received by i yet:

$$z_{(j,i)^D}^k \triangleq \sum_{t \in \mathcal{T}_j^{k-D-1},\, t+1 > \tau_{ij}^{k-1}} a_{ij} z_j^{t+1/2}. \tag{19}$$

We write next P-ASY-SUM-PUSH on the augmented graph in terms of the z-variables of both the computing and noncomputing agents, absorbing the $(\rho, \tilde{\rho})$-variables using (17)-(19).

The sum-step over the augmented graph. In the sum-step, the update of the z-variables of the computing agents reads:

$$z_{i^k}^{k+\frac{1}{2}} = z_{i^k}^k + \sum_{j \in \mathcal{N}_{i^k}^{\text{in}}} \left(\rho_{i^k j}^{\tau_{i^k j}^k} - \tilde{\rho}_{i^k j}^k\right) + \epsilon^k$$

$$\overset{(17)-(19)}{=} z_{i^k}^k + \sum_{j \in \mathcal{N}_{i^k}^{\text{in}}} \sum_{d=k-\tau_{i^k j}^k}^{D} z_{(j,i^k)^d}^k + \epsilon^k; \tag{20a}$$

$$z_j^{k+\frac{1}{2}} = z_j^k, \quad j \in \mathcal{V} \setminus \{i^k\}. \tag{20b}$$

In words, node i^k builds the update $z_{i^k}^k \to z_{i^k}^{k+\frac{1}{2}}$ based upon the masses transmitted by the noncomputing agents $(j,i^k)^{k-\tau_{i^k j}^k}, (j,i^k)^{k-\tau_{i^k j}^k+1}, \dots, (j,i^k)^D$ [cf. (20a)]. All the

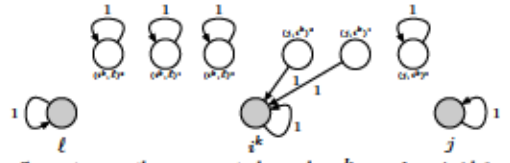


Figure 3. Sum step on the augmented graph: $\tau^k_{i^k j} = k-1$ (delay one); the two noncomputing agents, $(j,i^k)^1$ and $(j,i^k)^2$, send their masses to i^k.

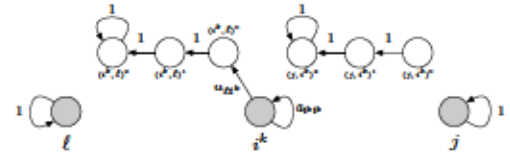


Figure 4. Push step on the augmented graph: Agent i^k keeps $a_{i^k i^k} z_{i^k}^{k+1/2}$ while sending $a_{\ell i^k} z_{i^k}^{k+1/2}$ to the virtual nodes $(i^k,\ell)^0$, $\ell \in \mathcal{N}^{\text{out}}_{i^k}$.

other computing agents keep their masses unchanged [cf. (20b)]. The updates of the noncomputing agents is set to

$$z^{k+\frac{1}{2}}_{(j,i^k)^d} \triangleq 0, \quad d = k - \tau^k_{i^k j}, \dots, D, \quad j \in \mathcal{N}^{\text{in}}_{i^k}; \tag{20c}$$

$$z^{k+\frac{1}{2}}_{(j',i)^\tau} \triangleq z^k_{(j',i)^\tau}, \quad \text{for all the other } (j',i)^\tau \in \widehat{\mathcal{V}}. \tag{20d}$$

The noncomputing agents in (20c) set their variables to zero (as they transferred their masses to i^k) while the other noncomputing agents keep their variables unchanged [cf. (20d)]. Fig. 3 illustrates the sum-step over the augmented graph.

The push-step over the augmented graph. In the push-step, the update of the z-variables of the computing agents reads:

$$z^{k+1}_{i^k} = a_{i^k i^k} z^{k+\frac{1}{2}}_{i^k}; \tag{21a}$$

$$z^{k+1}_{j} = z^{k+\frac{1}{2}}_{j}, \qquad \text{for } j \in \mathcal{V} \setminus \{i^k\}. \tag{21b}$$

In words, agent i^k keeps the portion $a_{i^k i^k} z^{k+\frac{1}{2}}_{i^k}$ of the new generated mass [cf. (21a)] whereas the other computing agents do not change their variables [cf. (21b)]. The noncomputing agents update as:

$$z^{k+1}_{(i^k,\ell)^0} \triangleq a_{\ell i^k} z^{k+1/2}_{i^k}, \quad \ell \in \mathcal{N}^{\text{out}}_{i^k}; \tag{21c}$$

$$z^{k+1}_{(i,j)^0} \triangleq 0, \quad (i,j) \in \mathcal{E}, \quad i \neq i^k; \tag{21d}$$

$$z^{k+1}_{(i,j)^d} \triangleq z^{k+\frac{1}{2}}_{(i,j)^{d-1}}, \quad d = 1, \dots, D-1, \quad (i,j) \in \mathcal{E}; \tag{21e}$$

$$z^{k+1}_{(i,j)^D} \triangleq z^{k+\frac{1}{2}}_{(i,j)^D} + z^{k+\frac{1}{2}}_{(i,j)^{D-1}}, \quad (i,j) \in \mathcal{E}. \tag{21f}$$

In words, the computing agent i^k pushes its masses $a_{\ell i^k} z^{k+\frac{1}{2}}_{i^k}$ to the noncomputing agents $(i^k,\ell)^0$, with $\ell \in \mathcal{N}^{\text{out}}_{i^k}$ [cf. (21c)]. As the other noncomputing agents $(i,j)^0$, $i \neq i^k$, do not receive any mass for their associated computing agents, they set their variables to zero [cf. (21d)]. Finally the other noncomputing agents $(i,j)^d$, with $0 \leq d \leq D-1$, transfers their mass to the next noncomputing node $(j,i)^{d+1}$ [cf. (21f), (21e)]. This push-step is illustrated in Fig. 4.

The following result establishes the equivalence between the update of the enlarged system with that of Algorithm 1.

Proposition 12. *Consider the setting of Theorem 7. The values of the z-variables of the computing agents in* (20)-(21) *coincide with those of the z-variables generated by* P-ASY-SUM-PUSH *(Algorithm 1), for all iterations $k \in \mathbb{N}_0$.*

Proof. By construction, the updates of the computing agents as in (20a)-(20b) and (21a)-(21b) coincide with the z-updates in the sum- and push-steps of P-ASY-SUM-PUSH, respectively. Therefore, we only need to show that the updates of the noncomputing agents are consistent with those of the $(\rho,\tilde{\rho})$-variables in P-ASY-SUM-PUSH. This follows using (17) and noting that the updates (21c)-(21f) are compliant with (18) and (19). For instance, by (17)-(18), it must be $z^{k+1}_{(i^k,j)^0} = a_{ji^k} z_j^{t+1/2} \cdot \mathbb{1}[t = k \in \mathcal{T}^k_{i^k} \text{ and } t+1 > \tau^k_{ji^k}] = a_{ji^k} z_j^{k+1/2}$, which in fact coincides with (21c). The other equations (21d)–(21f) can be similarly validated. □

Proposition 12 opens the way to study convergence of P-ASY-SUM-PUSH via that of the synchronous perturbed push-sum algorithm (20)-(21). To do so, it is convenient to rewrite (20)-(21) in vector-matrix form, as described next.

We begin introducing an enumeration rule for the components of the z-vector in the augmented system. We enumerate all the elements of $\mathcal{E}$ as $1, 2, \dots, |\mathcal{E}|$. The computing agents in $\widehat{\mathcal{V}}$ are indexed as in $\mathcal{V}$, that is, $1, 2, \dots, I$. Each noncomputing agent $(j,i)^d$ is indexed as $I + d|\mathcal{E}| + s$, where s is the index associated with (j,i) in $\mathcal{E}$; we will use interchangeably $z_{I+d|\mathcal{E}|+s}$ and $z_{(j,i)^d}$. We define the z-vector as $\widehat{\mathbf{z}} = [z_t]_{t=1}^S$; and its value at iteration $k \in \mathbb{N}_0$ is denoted by $\widehat{\mathbf{z}}^k$.

The transition matrix $\mathbf{S}^k$ of the sum step is defined as

$$S^k_{hm} \triangleq \begin{cases} 1, & \text{if } m \in \{(j,i^k)^d \mid k - \tau^k_{i^k j} \leq d \leq D\} \\ & \text{and } h = i^k; \\ 1, & \text{if } m \in \widehat{\mathcal{V}} \setminus \{(j,i^k)^d \mid k - \tau^k_{i^k j} \leq d \leq D\} \\ & \text{and } h = m; \\ 0, & \text{otherwise.} \end{cases}$$

Let $\boldsymbol{\varepsilon}^k \triangleq \epsilon^k \mathbf{e}_{i^k}$ be the $S-$dimensional perturbation vector. The sum-step can be written in compact form as

$$\widehat{\mathbf{z}}^{k+\frac{1}{2}} = \mathbf{S}^k \widehat{\mathbf{z}}^k + \boldsymbol{\varepsilon}^k. \tag{22}$$

Define the transition matrix $\mathbf{P}^k$ of the push step as

$$P^k_{hm} \triangleq \begin{cases} a_{ji^k}, & \text{if } m = i^k \text{ and } h = (j,i^k)^0, j \in \mathcal{N}^{\text{out}}_{i^k}; \\ a_{i^k i^k}, & \text{if } m = h = i^k; \\ 1, & \text{if } m = h \in \mathcal{V} \setminus i^k; \\ 1, & \text{if } m = (i,j)^d,\ h = (i,j)^{d+1}, \\ & (i,j) \in \mathcal{E},\ 0 \leq d \leq D-1; \\ 1, & \text{if } m = h = (i,j)^D,\ (i,j) \in \mathcal{E}; \\ 0, & \text{otherwise} \end{cases}$$

Then, the push-step can be written as

$$\widehat{\mathbf{z}}^{k+1} = \mathbf{P}^k \widehat{\mathbf{z}}^{k+\frac{1}{2}}. \tag{23}$$

Combing (22) and (23), yields

$$\widehat{\mathbf{z}}^{k+1} = \widehat{\mathbf{A}}^k \widehat{\mathbf{z}}^k + \mathbf{p}^k, \quad \widehat{\mathbf{A}}^k \triangleq \mathbf{P}^k \mathbf{S}^k, \quad \mathbf{p}^k \triangleq \mathbf{P}^k \boldsymbol{\varepsilon}^k. \tag{24}$$

The updates of the ϕ variables and the definition of the ϕ-vector are similar as above. In summary, the P-ASY-SUM-PUSH algorithm can be rewritten in compact form as

$$\widehat{\mathbf{z}}^{k+1} = \widehat{\mathbf{A}}^k \widehat{\mathbf{z}}^k + \mathbf{p}^k, \quad \mathbf{p}^k = \epsilon^k \mathbf{P}^k \mathbf{e}_{i^k}; \tag{25a}$$

$$\widehat{\boldsymbol{\phi}}^{k+1} = \widehat{\mathbf{A}}^k \widehat{\boldsymbol{\phi}}^k; \tag{25b}$$

with initialization: $z_i^0 \in \mathbb{R}$ and $\phi_i^0 = 1$, for $i \in \mathcal{V}$; and $z_i^0 = 0$ and $\phi_i^0 = 0$, for $i \in \widehat{\mathcal{V}} \setminus \mathcal{V}$.

Remark 13 (Comparison with [30]–[32], [38])**.** *The idea of reducing asynchronous (consensus) algorithms into synchronous ones over an augmented system was already explored in [31], [32], [38]. However, there are several important differences between the models therein and the proposed augmented graph. First of all, [38] extends the analysis in [30] to deal with asynchronous activations, but both work consider only packet losses (no delays). Second, our augmented graph model departs from that in [31], [32] in the following aspects: i) in our model, the virtual nodes are associated with the edges of the original graph rather than the nodes; ii) the noncomputing nodes store the information on fly (i.e., generated by a sender but not received by the intended receiver yet), while in [31], [32], each noncomputing agent owns a delayed copy of the message generated by the associated computing agent; and iii) the dynamics* (25) *over the augmented graph used to describe the P-ASY-SUM-PUSH procedure is different from those of the asynchronous consensus schemes [31, (1)] and [32, (1)].*

B. Step 2: Proof of Theorem 7

1) Preliminaries: We begin studying some properties of the matrix product $\widehat{\mathbf{A}}^{k:t}$, which will be instrumental to prove convergence of the perturbed push-sum scheme (25).

Lemma 14. *Let $\{\widehat{\mathbf{A}}^k\}_{k\in\mathbb{N}_0}$ be the sequence of matrices in* (25)*, generated by Algorithm 1, under Assumption 6, and with $\mathbf{A} \triangleq (a_{ij})_{i,j=1}^I$ satisfying Assumption 4 (i),(iii). The following hold: for all $k \in \mathbb{N}_0$, a) $\widehat{\mathbf{A}}^k$ is column stochastic; and b) the entries of the first I rows of $\widehat{\mathbf{A}}^{k+K_1-1:k}$ are uniformly lower bounded by $\eta \triangleq \bar{m}^{K_1} \in (0,1)$, with $K_1 \triangleq (2I-1)\cdot T + I \cdot D$.*

Proof. The lemma essentially proves that $(\widehat{\mathbf{A}}^{k+K_1-1:k})^\top$ is a SIA (Stochastic Indecomposable Aperiodic) matrix [32], by showing that for any time length of K_1 iterations, there exists a path from any node m in the augmented graph to any computing node h. While at a high level the proof shares some similarities with that of [31, Lemma 2] and [32, Lemma 5 (a)], there are important differences due to the distinct modeling of our augmented system; because of the space limitation, we omit further details and refer to the technical report [2, Appendix A]. □

The key result of this section is stated next and shows that as $k-t$ increases, $\widehat{\mathbf{A}}^{k:t}$ approaches a column stochastic rank one matrix at a linear rate. Given Lemma 14, the proof follows the path of [31, Lemma 4, Lemma 5], [32, Lemma 4, Lemma 5(b,c)] and thus is omitted.

Lemma 15. *In the setting above, there exists a sequence of stochastic vectors $\{\boldsymbol{\xi}^k\}_{k\in\mathbb{N}_0}$ such that, for any $k \geq t \in \mathbb{N}_0$ and i, $j \in \{1,\cdots,S\}$, there holds*

$$\left|\widehat{A}_{ij}^{k:t} - \xi_i^k\right| \leq C\rho^{k-t}, \tag{26}$$

with

$$C \triangleq 2\frac{1+\bar{m}^{-K_1}}{1-\bar{m}^{K_1}}, \quad \rho \triangleq (1-\bar{m}^{K_1})^{\frac{1}{K_1}} \in (0,1).$$

Furthermore, $\xi_i^k \geq \eta$, for all $i \in \mathcal{V}$ and $k \in \mathbb{N}_0$.

2) Proof of Theorem 7: Applying (25) telescopically, yields: $\widehat{\mathbf{z}}^{k+1} = \widehat{\mathbf{A}}^{k:0}\widehat{\mathbf{z}}^0 + \sum_{l=1}^k \widehat{\mathbf{A}}^{k:l}\mathbf{p}^{l-1} + \mathbf{p}^k$ and $\widehat{\boldsymbol{\phi}}^{k+1} = \widehat{\mathbf{A}}^{k:0}\widehat{\boldsymbol{\phi}}^0$,which using the column stochasticity of $\widehat{\mathbf{A}}^{k:t}$, yields

$$\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1} = \mathbf{1}^\top\widehat{\mathbf{z}}^0 + \sum_{l=0}^k \mathbf{1}^\top\mathbf{p}^l, \quad \mathbf{1}^\top\widehat{\boldsymbol{\phi}}^{k+1} = \mathbf{1}^\top\widehat{\boldsymbol{\phi}}^0 = I. \tag{27}$$

Using (27) and $\phi_i^{k+1} \geq I\eta$, for all $i \in \mathcal{V}$ and $k \geq K_1 - 1$ [due to Lemma 14(b)], we have: for $i \in \mathcal{V}$ and $k \geq K_1 - 1$,

$$\begin{aligned}
&\left|\frac{z_i^{k+1}}{\phi_i^{k+1}} - \frac{\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1}}{I}\right| \leq \frac{1}{I\eta}\left|z_i^{k+1} - \frac{\phi_i^{k+1}}{I}(\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1})\right| \\
&\leq \frac{1}{I\eta}\left|z_i^{k+1} - \xi_i^k\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1}\right| + \frac{1}{I\eta}\left|\left(\xi_i^k - \frac{\phi_i^{k+1}}{I}\right)\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1}\right| \\
&\leq \frac{1}{I\eta}\left|z_i^{k+1} - \xi_i^k\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1}\right| \\
&\quad + \frac{1}{I\eta}\left|\xi_i^k - \frac{\widehat{\mathbf{A}}_{i,:}^{k:0}\widehat{\boldsymbol{\phi}}^0}{I}\right| \cdot \left|\mathbf{1}^\top\widehat{\mathbf{z}}^0 + \sum_{l=0}^k \mathbf{1}^\top\mathbf{p}^l\right| \\
&\overset{(26)}{\leq} \frac{1}{I\eta}\left|z_i^{k+1} - \xi_i^k\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1}\right| + \frac{C\rho^k}{\sqrt{I}\eta}\left(\left\|\mathbf{z}^0\right\| + \sum_{l=0}^k \left|\epsilon^l\right|\right)
\end{aligned} \tag{28}$$

The next lemma provides a bound of $\left|z_i^{k+1} - \xi_i^k\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1}\right|$.

Lemma 16. *Let $\{\widehat{\mathbf{z}}^k\}_{k=0}^\infty$ be the sequence generated by the perturbed system* (25a)*, under Assumption 6, $\mathbf{A} = (a_{ij})_{i,j=1}^I$ satisfying Assumption 4 (i), (iii), and given $\{\epsilon^k\}_{k\in\mathbb{N}_0}$. For any $i \in \mathcal{V}$ and $k \geq 0$, there holds*

$$\left|z_i^{k+1} - \xi_i^k\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1}\right| \leq C_0\left(\rho^k\left\|\mathbf{z}^0\right\| + \sum_{l=0}^k \rho^{k-l}\left|\epsilon^l\right|\right), \tag{29}$$

with $\{\boldsymbol{\xi}^k\}_{n\in\mathbb{N}_0}$ defined in Lemma 15 and $C_0 \triangleq C\sqrt{2S}/\rho$.

Proof.

$$\begin{aligned}
&\left|z_i^{k+1} - \xi_i^k\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1}\right| \overset{(25a)}{=} \left|\left(\widehat{\mathbf{A}}_{i,:}^{k:0}\widehat{\mathbf{z}}^0 + \sum_{l=1}^k \widehat{\mathbf{A}}_{i,:}^{k:l}\mathbf{p}^{l-1} + p_i^k\right)\right. \\
&\quad \left. - \xi_i^k\left(\mathbf{1}^\top\widehat{\mathbf{z}}^0 + \sum_{l=0}^k \mathbf{1}^\top\mathbf{p}^l\right)\right| \leq \left|p_i^k\right| + \left|\mathbf{1}^\top\mathbf{p}^k\right| \\
&\quad + \left\|\widehat{\mathbf{A}}_{i,:}^{k:0} - \xi_i^k\mathbf{1}^\top\right\|\left\|\widehat{\mathbf{z}}^0\right\| + \sum_{l=1}^k \left\|\widehat{\mathbf{A}}_{i,:}^{k:l} - \xi_i^k\mathbf{1}^\top\right\|\left\|\mathbf{p}^{l-1}\right\| \\
&\overset{(26)}{\leq} \frac{\sqrt{S}}{\rho}C\left(\rho^k\left\|\widehat{\mathbf{z}}^0\right\| + \sum_{l=0}^k \rho^{k-l}\left\|\mathbf{P}^l\right\|\left|\epsilon^l\right|\right) \\
&\overset{(a)}{\leq} C_0\left(\rho^k\left\|\mathbf{z}^0\right\| + \sum_{l=0}^k \rho^{k-l}\left|\epsilon^l\right|\right),
\end{aligned}$$

where in (a) we used $\left\|\mathbf{P}^l\right\| \leq \sqrt{\left\|\mathbf{P}^l\right\|_1\left\|\mathbf{P}^l\right\|_\infty} \leq \sqrt{2}$. □

Combing Eq. (28) and (29) leads to

$$\left|\frac{z_i^{k+1}}{\phi_i^{k+1}} - \frac{\mathbf{1}^\top\widehat{\mathbf{z}}^{k+1}}{I}\right| \leq C_1\left(\rho^k\left\|\mathbf{z}^0\right\| + \sum_{l=0}^k \rho^{k-l}\left|\epsilon^l\right|\right),$$

where we defined $C_1 \triangleq C_0 \cdot 2/(I\,\eta)$.

Recalling the definition of $\mathfrak{m}_z^k \triangleq \sum_{i=1}^I z_i^k + \sum_{(j,i)\in\mathcal{E}}(\rho_{ij}^k - \tilde{\rho}_{ij}^k)$, to complete the proof, it remains to show that

$$\mathfrak{m}_z^k \overset{(I)}{=} \sum_{i=1}^{I} z_i^0 + \sum_{t=0}^{k-1} \epsilon^t \overset{(II)}{=} \mathbf{1}^\top \widehat{\mathbf{z}}^k. \tag{30}$$

We prove next the equalities (I) and (II) separately.

Proof of (I): Since $\mathfrak{m}_z^0 = \sum_{i=1}^I z_i^0$, it suffices to show that $\mathfrak{m}_z^{k+1} = \mathfrak{m}_z^k + \epsilon^k$ for all $k \in \mathbb{N}_0$. Since agent i^k triggers $k \to k+1$, we only need to show that

$$\begin{aligned} &z_{i^k}^{k+1} + \sum_{j \in \mathcal{N}_{i^k}^{\text{in}}} (\rho_{i^k j}^{k+1} - \tilde{\rho}_{i^k j}^{k+1}) + \sum_{j \in \mathcal{N}_{i^k}^{\text{out}}} (\rho_{j i^k}^{k+1} - \tilde{\rho}_{j i^k}^{k+1}) \\ =&z_{i^k}^{k} + \sum_{j \in \mathcal{N}_{i^k}^{\text{in}}} (\rho_{i^k j}^{k} - \tilde{\rho}_{i^k j}^{k}) + \sum_{j \in \mathcal{N}_{i^k}^{\text{out}}} (\rho_{j i^k}^{k} - \tilde{\rho}_{j i^k}^{k}) + \epsilon^k. \end{aligned}$$

This is proved in [1, Eq. (13)], and thus is omitted.

Proof of (II): Using (27), yields $\mathbf{1}^\top \widehat{\mathbf{z}}^{k+1} = \mathbf{1}^\top \widehat{\mathbf{z}}^0 + \sum_{l=0}^k \mathbf{1}^\top \mathbf{p}^l = \mathbf{1}^\top \widehat{\mathbf{z}}^k + \mathbf{1}^\top \boldsymbol{\epsilon}^k = \sum_{i=1}^I z_i^0 + \sum_{t=0}^k \epsilon^t$. □

VII. ASY-SONATA–Proof of Theorem 9

We organize the proof in the following steps: **Step 1:** We introduce and study convergence of an auxiliary perturbed consensus scheme, which serves as a unified model for the descent and consensus updates in ASY-SONATA–the main result is summarized in Proposition 18; **Step 2:** We introduce the consensus and gradient tracking errors along with a suitably defined optimization error; and we derive bounds connecting these quantities, building on results in Step 1 and convergence of P-ASY-SUM-PUSH–see Proposition 19. The goal is to prove that the aforementioned errors vanish at a linear rate. To do so, **Step 3** introduces a general form of the small gain theorem–Theorem 23–along with some technical results, which allows us to establish the desired linear convergence through the boundedness of the solution of an associated linear system of inequalities. **Step 4** builds such a linear system for the error quantities introduced in Step 2 and proves the boundedness of its solution, proving thus Theorem 9. The rate expression (13) is derived in Appendix C. Through the proof we assume $n = 1$ (scalar variables); and define $C_L \triangleq \max_{i=1,\dots,I} L_i$ and $L \triangleq \sum_{i=1}^I L_i$.

Step 1: A perturbed asynchronous consensus scheme

We introduce a unified model to study the dynamics of the consensus and optimization errors in ASY-SONATA, which consists in pulling out the tracking update (Step 5) and treat the z-variables–the term $-\gamma^k z_{i^k}^k$ in (11)–as an exogenous perturbation δ^k. More specifically, consider the following scheme (with a slight abuse of notation, we use the same symbols as in ASY-SONATA):

$$v_{i^k}^{k+1} = x_{i^k}^k + \delta^k, \tag{31a}$$

$$x_{i^k}^{k+1} = w_{i^k i^k} v_{i^k}^{k+1} + \sum_{j \in \mathcal{N}_{i^k}^{\text{in}}} w_{i^k j} v_j^{k-d_j^k}, \tag{31b}$$

$$v_j^{k+1} = v_j^k,\ x_j^{k+1} = x_j^k, \quad \forall j \in \mathcal{V} \setminus \{i^k\}, \tag{31c}$$

with given $x_i^0 \in \mathbb{R}$, $v_i^t = 0$, $t = -D, -D+1, \dots, 0$, for all $i \in \mathcal{V}$. We make the blanket assumption that agents' activations and delays satisfy Assumption 6.

Let us rewrite (31) in a vector-matrix form. Define $\mathbf{x}^k \triangleq [x_1^k, \cdots, x_I^k]^\top$ and $\mathbf{v}^k \triangleq [v_1^k, \cdots, v_I^k]^\top$. Construct the $(D+2)I$ dimensional concatenated vectors

$$\mathbf{h}^k \triangleq [\mathbf{x}^{k\top}, \mathbf{v}^{k\top}, \mathbf{v}^{k-1\top}, \cdots, \mathbf{v}^{k-D\top}]^\top, \quad \boldsymbol{\delta}^k \triangleq \delta^k \mathbf{e}_{i^k}; \tag{32}$$

and the augmented matrix $\widehat{\mathbf{W}}^k$, defined as

$$\widehat{W}_{rm}^k \triangleq \begin{cases} w_{i^k i^k}, & \text{if } r = m = i^k; \\ w_{i^k j}, & \text{if } r = i^k,\ m = j + (d_j^k + 1)I; \\ 1, & \text{if } r = m \in \{1, 2, \dots, 2I\} \setminus \{i^k, i^k + I\}; \\ 1, & \text{if } r \in \{2I+1, 2I+2, \dots, (D+2)I\} \\ & \quad \cup \{i^k + I\} \text{ and } m = r - I; \\ 0, & \text{otherwise.} \end{cases}$$

System (31) can be rewritten in compact form as

$$\mathbf{h}^{k+1} = \widehat{\mathbf{W}}^k (\mathbf{h}^k + \boldsymbol{\delta}^k), \tag{33}$$

The following lemma captures the asymptotic behavior of $\widehat{\mathbf{W}}^k$.

Lemma 17. *Let $\{\widehat{\mathbf{W}}^k\}_{k \in \mathbb{N}_0}$ be the sequence of matrices in (33), generated by (31), under Assumption 6 and with $\mathbf{W}$ satisfying Assumption 4 (i), (ii). The following hold: for all $k \in \mathbb{N}_0$, a) $\widehat{\mathbf{W}}^k$ is row stochastic; b) there exists a sequence of stochastic vectors $\{\boldsymbol{\psi}^k\}_{k \in \mathbb{N}_0}$ such that*

$$\left\| \widehat{\mathbf{W}}^{k:t} - \mathbf{1} \boldsymbol{\psi}^{t\top} \right\| \le C_2 \rho^{k-t}, \quad C_2 \triangleq \frac{2\sqrt{(D+2)I}(1 + \bar{m}^{-K_1})}{1 - \bar{m}^{-K_1}} \tag{34}$$

Furthermore, $\psi_i^k \ge \eta = \bar{m}^{K_1}$, for all $k \ge 0$ and $i \in \mathcal{V}$.

Proof. The proof follows similar techniques as in [31], [32], and can be found in the technical report [2, Appendix G]. □

We define now a proper consensus error for (33). Writing $\mathbf{h}^k$ in (33) recursively, yields

$$\mathbf{h}^{k+1} = \widehat{\mathbf{W}}^{k:0} \mathbf{h}^0 + \sum_{l=0}^{k} \widehat{\mathbf{W}}^{k:l} \boldsymbol{\delta}^l. \tag{35}$$

Using Lemma 17, for any fixed $N \in \mathbb{N}_0$, we have

$$\lim_{k \to \infty} (\widehat{\mathbf{W}}^{k:0} \mathbf{h}^0 + \sum_{l=0}^{N} \widehat{\mathbf{W}}^{k:l} \boldsymbol{\delta}^l) = \mathbf{1} \boldsymbol{\psi}^{0\top} \mathbf{h}^0 + \sum_{l=0}^{N} \mathbf{1} \boldsymbol{\psi}^{l\top} \boldsymbol{\delta}^l.$$

Define

$$x_{\boldsymbol{\psi}}^0 \triangleq \boldsymbol{\psi}^{0\top} \mathbf{h}^0, \quad x_{\boldsymbol{\psi}}^{k+1} \triangleq \boldsymbol{\psi}^{0\top} \mathbf{h}^0 + \sum_{l=0}^{k} \boldsymbol{\psi}^{l\top} \boldsymbol{\delta}^l,\ k \in \mathbb{N}_0. \tag{36}$$

Applying (36) inductively, it is easy to check that

$$x_{\boldsymbol{\psi}}^{k+1} = x_{\boldsymbol{\psi}}^k + \boldsymbol{\psi}^{k\top} \boldsymbol{\delta}^k = x_{\boldsymbol{\psi}}^k + \psi_{i^k}^k \delta^k. \tag{37}$$

We are now ready to state the main result of this subsection, which is a bound of the consensus disagreement $\|\mathbf{h}^{k+1} - \mathbf{1} x_{\boldsymbol{\psi}}^{k+1}\|$ in terms of the magnitude of the perturbation.

Proposition 18. *In the above setting, the consensus error $\|\mathbf{h}^{k+1} - \mathbf{1} x_{\boldsymbol{\psi}}^{k+1}\|$ satisfies: for all $k \in \mathbb{N}_0$,*

$$\left\| \mathbf{h}^{k+1} - \mathbf{1} x_{\boldsymbol{\psi}}^{k+1} \right\| \le C_2 \rho^k \left\| \mathbf{h}^0 - \mathbf{1} x_{\boldsymbol{\psi}}^0 \right\| + C_2 \sum_{l=0}^{k} \rho^{k-l} \left| \delta^l \right|.$$

Proof. The proof follows readily from (35), (36), and Lemma 17; we omit further details. □

Step 2: Consensus, tracking, and optimization errors

1) Consensus disagreement: As anticipated, the updates of ASY-SONATA are also described by (31), if one sets therein $\delta^k = -\gamma^k z_{i^k}^k$ (with $z_{i^k}^k$ satisfying Step 5 of ASY-SONATA). Let $\mathbf{h}^k$ and $x_{\boldsymbol{\psi}}^k$ be defined as in (32) and (36), respectively, with $\delta^k = -\gamma^k z_{i^k}^k$. The consensus error at iteration k is defined as

$$E_{\mathbf{c}}^k \triangleq \left\|\mathbf{h}^k - \mathbf{1}x_{\boldsymbol{\psi}}^k\right\|. \tag{38}$$

2) Gradient tracking error: The gradient tracking step in ASY-SONATA is an instance of P-ASY-SUM-PUSH, with $\epsilon^k = \nabla f_{i^k}(\mathbf{x}_{i^k}^{k+1}) - \nabla f_{i^k}(\mathbf{x}_{i^k}^k)$. By Proposition 12, P-ASY-SUM-PUSH is equivalent to (25). In view of Lemma 16 and the following property $\mathbf{1}^\top \widehat{\mathbf{z}}^k = \sum_{i=1}^I \nabla f_i(x_i^0) + \sum_{t=0}^{k-1}\left(\nabla f_{i^t}(x_{i^t}^{t+1}) - \nabla f_{i^t}(x_{i^t}^t)\right) = \sum_{i=1}^I \nabla f_i(x_i^k)$ where the first equality follows from (30) and $\epsilon^k = \nabla f_{i^k}(\mathbf{x}_{i^k}^{k+1}) - \nabla f_{i^k}(\mathbf{x}_{i^k}^k)$ while in the second equality we used $x_j^{t+1} = x_j^t$, for $j \neq i^t$, the tracking error at iteration k along with the magnitude of the tracking variables are defined as

$$E_{\mathbf{t}}^k \triangleq \left|z_{i^k}^k - \xi_{i^k}^{k-1}\,\bar{g}^k\right|,\ E_{\mathbf{z}}^k \triangleq \left|z_{i^k}^k\right|,\ \bar{g}^k \triangleq \sum_{i=1}^I \nabla f_i(x_i^k), \tag{39}$$

with $\xi_i^{-1} \triangleq \eta$, $i \in \mathcal{V}$. Let $\mathbf{g}^k \triangleq [\nabla f_1(x_1^k), \dots, \nabla f_I(x_I^k)]^\top$.

3) Optimization error: Let $x^\star$ be the unique minimizer of F. Given the definition of consensus disagreement in (38), we define the optimization error at iteration k as

$$E_{\mathbf{o}}^k \triangleq \left|x_{\boldsymbol{\psi}}^k - x^\star\right|. \tag{40}$$

Note that this is a natural choice as, if consensual, all agents' local variables will converge to a limit point of $\{x_{\boldsymbol{\psi}}^k\}_{k\in\mathbb{N}_0}$.

4) Connection among $E_{\mathbf{c}}^k$, $E_{\mathbf{t}}^k$, $E_{\mathbf{z}}^k$, and $E_{\mathbf{o}}^k$: The following proposition establishes bounds on the above quantities.

Proposition 19. *Let $\{\mathbf{x}^k, \mathbf{v}^k, \mathbf{z}^k\}_{k\in\mathbb{N}_0}$ be the sequence generated by ASY-SONATA, in the setting of Theorem 9, but possibly with a time-varying step-size $\{\gamma^k\}_{k\in\mathbb{N}_0}$. The error quantities $E_{\mathbf{c}}^k$, $E_{\mathbf{t}}^k$, $E_{\mathbf{z}}^k$, and $E_{\mathbf{o}}^k$ satisfy: for all $k \in \mathbb{N}_0$,*

$$E_{\mathbf{c}}^{k+1} \leq C_2\rho^k E_{\mathbf{c}}^0 + C_2\sum_{l=0}^k \rho^{k-l}\gamma^l E_{\mathbf{z}}^l. \tag{41a}$$

$$E_{\mathbf{t}}^{k+1} \leq 3C_0C_L\sum_{l=0}^k \rho^{k-l}\left(E_{\mathbf{c}}^l + \gamma^l E_{\mathbf{z}}^l\right) + C_0\rho^k\left\|\mathbf{g}^0\right\|; \tag{41b}$$

$$E_{\mathbf{z}}^k \ \leq\ E_{\mathbf{t}}^k + C_L\sqrt{I}E_{\mathbf{c}}^k + L\,E_{\mathbf{o}}^k \tag{41c}$$

Further assume $\gamma^k \leq 1/L$, $k \in \mathbb{N}_0$; then

$$E_{\mathbf{o}}^{k+1} \leq \sum_{l=0}^k\left(\prod_{t=l+1}^k\left(1-\tau\eta^2\gamma^t\right)\right)\left(C_L\sqrt{I}E_{\mathbf{c}}^l + E_{\mathbf{t}}^l\right)\gamma^l + \prod_{t=0}^k\left(1-\tau\eta^2\gamma^t\right)E_{\mathbf{o}}^0, \tag{41d}$$

where $\eta \in (0,1)$ is defined in Lemma 15 and τ is the strongly convexity constant of F.

Proof. Eq. (41a) follows readily from Proposition 18.

We prove now (41b). Recall $\mathbf{1}^\top\widehat{\mathbf{z}}^k = \bar{g}^k$. Using Lemma 16 with $\epsilon^k = \nabla f_{i^k}(\mathbf{x}_{i^k}^{k+1}) - \nabla f_{i^k}(\mathbf{x}_{i^k}^k)$, we obtain: for all $i \in \mathcal{V}$,

$$\begin{aligned}
&\left|z_i^{k+1} - \xi_i^k\,\bar{g}^{k+1}\right| \\
&\leq C_0\,\rho^k\left\|\mathbf{g}^0\right\| + C_0C_L\sum_{l=0}^k\rho^{k-l}\left|x_{i^l}^{l+1} - x_{i^l}^l\right| \\
&\leq C_0\,\rho^k\left\|\mathbf{g}^0\right\| + C_0C_L\sum_{l=0}^k\rho^{k-l}\left\|\mathbf{h}^{l+1} - \mathbf{h}^l\right\| \\
&\overset{(a)}{=} C_0\,\rho^k\left\|\mathbf{g}^0\right\| \\
&+ C_0C_L\sum_{l=0}^k\rho^{k-l}\left\|\left(\widehat{\mathbf{W}}^l - \mathbf{I}\right)\left(\mathbf{h}^l - \mathbf{1}x_{\boldsymbol{\psi}}^l\right) - \gamma^l z_{i^l}^l\widehat{\mathbf{W}}^l\mathbf{e}_{i^l}\right\| \\
&\leq C_0\rho^k\left\|\mathbf{g}^0\right\| + C_0C_L\sum_{l=0}^k\rho^{k-l}\Bigg(\|\widehat{\mathbf{W}}^l\|\gamma^l E_{\mathbf{z}}^l \\
&+ \left(\|\widehat{\mathbf{W}}^l\| + \|\mathbf{I}\|\right)E_{\mathbf{c}}^l\Bigg) \\
&\overset{(b)}{\leq} C_0\rho^k\left\|\mathbf{g}^0\right\| + 3C_0C_L\sum_{l=0}^k\rho^{k-l}\left(E_{\mathbf{c}}^l + \gamma^l E_{\mathbf{z}}^l\right),
\end{aligned}$$

where in (a) we used (33) and the row stochasticity of $\widehat{\mathbf{W}}^k$ [Lemma 17(a)]; and (b) follows from $\|\widehat{\mathbf{W}}^l\| \leq \sqrt{\|\widehat{\mathbf{W}}^l\|_1\|\widehat{\mathbf{W}}^l\|_\infty} \leq \sqrt{3}$. This proves (41b).

Eq. (41c) follows readily from

$$\begin{aligned}
E_{\mathbf{z}}^k = \left|z_{i^k}^k\right| &\leq \left|z_{i^k}^k - \xi_{i^k}^{k-1}\,\bar{g}^k\right| + \xi_{i^k}^{k-1}\left|\bar{g}^k - \nabla F(x_{\boldsymbol{\psi}}^k)\right| \\
&\quad + \xi_{i^k}^{k-1}\left|\nabla F(x_{\boldsymbol{\psi}}^k) - \nabla F(x^\star)\right|.
\end{aligned}$$

Finally, we prove (41d). Using (41c) and $x_{\boldsymbol{\psi}}^{k+1} = x_{\boldsymbol{\psi}}^k - \gamma\psi_{i^k}^k z_{i^k}^k$ [cf. (37) and recall $\delta^k = -\gamma z_{i^k}^k$], we can write

$$\begin{aligned}
E_{\mathbf{o}}^{k+1} &= \left|x_{\boldsymbol{\psi}}^k - \gamma^k\psi_{i^k}^k z_{i^k}^k - x^\star\right| \\
&\leq \gamma^k\psi_{i^k}^k\xi_{i^k}^{k-1}\left|\nabla F(x_{\boldsymbol{\psi}}^k) - \bar{g}^k\right| + \gamma^k\psi_{i^k}^k\left|\xi_{i^k}^{k-1}\,\bar{g}^k - z_{i^k}^k\right| \\
&+ \left|x_{\boldsymbol{\psi}}^k - \gamma^k\psi_{i^k}^k\xi_{i^k}^{k-1}\nabla F(x_{\boldsymbol{\psi}}^k) - x^\star\right| \\
&\overset{(a)}{\leq} \left(1-\tau\eta^2\gamma^k\right)E_{\mathbf{o}}^k + C_L\sqrt{I}\gamma^k\left\|\mathbf{h}^k - \mathbf{1}x_{\boldsymbol{\psi}}^k\right\| + \gamma^k E_{\mathbf{t}}^k
\end{aligned}$$

where in (a) we used $\eta^2 \leq \psi_{i^k}^k\xi_{i^k}^{k-1} < 1$ (cf. Lemma 15) and $|x - \gamma\nabla F(x) - x^\star| \leq (1-\tau\gamma)\,|x - x^\star|$, which holds for $\gamma \leq 1/L$. The desired result (41d) follows readily by applying the above inequality telescopically. □

Step 3: The generalized small gain theorem

The last step of our proof is to show that the error quantities $E_{\mathbf{c}}^k$, $E_{\mathbf{t}}^k$, $E_{\mathbf{z}}^k$, and $E_{\mathbf{o}}^k$ vanish linearly. This is not a straightforward task, as these quantities are interconnected through the inequalities (41). This subsection provides tools to address this issue. The key result is a generalization of the small gain theorem (cf. Theorem 23), first used in [33].

Definition 20 ([33])**.** *Given the sequence $\{u^k\}_{k=0}^\infty$, a constant $\lambda \in (0,1)$, and $N \in \mathbb{N}$, let us define*

$$|u|^{\lambda,N} = \max_{k=0,\dots,N}\frac{\left|u^k\right|}{\lambda^k},\quad |u|^\lambda = \sup_{k\in\mathbb{N}_0}\frac{\left|u^k\right|}{\lambda^k}.$$

If $|u|^\lambda$ is upper bounded, then $u^k = \mathcal{O}(\lambda^k)$, for all $k \in \mathbb{N}_0$.

The following lemma shows how one can interpret the inequalities in (41) using the notions introduced in Definition 20.

Lemma 21. *Let $\{u^k\}_{k=0}^{\infty}, \{v_i^k\}_{k=0}^{\infty}, i = 1, \dots, m$, be nonnegative sequences; let $\lambda_0, \lambda_1, \dots, \lambda_m \in (0,1)$; and let $R_0, R_1, \dots, R_m \in \mathbb{R}_+$ such that*

$$u^{k+1} \leq R_0(\lambda_0)^k + \sum_{i=1}^{m} R_i \sum_{l=0}^{k} (\lambda_i)^{k-l} v_i^l, \quad \forall k \in \mathbb{N}_0.$$

Then, there holds

$$|u|^{\lambda,N} \leq u^0 + \frac{R_0}{\lambda} + \sum_{i=1}^{m} \frac{R_i}{\lambda - \lambda_i} |v_i|^{\lambda,N},$$

for any $\lambda \in (\max_{i=0,1,\dots,m} \lambda_i, 1)$ and $N \in \mathbb{N}$.

Proof. See Appendix A. □

Lemma 22. *Let $\{u^k\}_{k=0}^{\infty}$ and $\{v^k\}_{k=0}^{\infty}$ be two nonnegative sequences. The following hold*

a. $u^k \leq v^k$, *for all* $k \in \mathbb{N}_0 \implies |u|^{\lambda,N} \leq |v|^{\lambda,N}$, *for any* $\lambda \in (0,1)$ *and* $N \in \mathbb{N}$;

b.
$$|\beta_1 u + \beta_2 v|^{\lambda,N} \leq |\beta_1| |u|^{\lambda,N} + |\beta_2| |v|^{\lambda,N},$$
for any $\beta_1, \beta_2 \in \mathbb{R}$, $\lambda \in (0,1)$, and positive integer N.

The major result of this section is the generalized small gain theorem, as stated next.

Theorem 23. *(Generalized Small Gain Theorem) Given nonnegative sequences $\{u_i^k\}_{k=0}^{\infty}, i = 1, \dots, m$, a non-negative matrix $\mathbf{T} \in \mathbb{R}^{m \times m}$, $\boldsymbol{\beta} \in \mathbb{R}^m$, and $\lambda \in (0,1)$ such that*

$$\mathbf{u}^{\lambda,N} \preccurlyeq \mathbf{T}\mathbf{u}^{\lambda,N} + \boldsymbol{\beta}, \quad \forall N \in \mathbb{N}, \tag{43}$$

where $\mathbf{u}^{\lambda,N} \triangleq [|u_1|^{\lambda,N}, \dots, |u_m|^{\lambda,N}]^\top$. If $\rho(\mathbf{T}) < 1$, then $|u_i|^{\lambda}$ is bounded, for all $i = 1, \dots, m$. That is, each u_i^k vanishes at a R-linear rate $\mathcal{O}(\lambda^k)$.

Proof. See Appendix B. □

Then following results are instrumental to find a sufficient condition for $\rho(\mathbf{T}) < 1$.

Lemma 24. *Consider a polynomial $p(z) = z^m - a_1 z^{m-1} - a_2 z^{m-2} - \dots - a_{m-1} z - a_m$, with $z \in \mathbb{C}$ and $a_i \in \mathbb{R}_+$, $i = 1, \dots m$. Define $z_p \triangleq \max\{|z_i| \mid p(z_i) = 0, \ i = 1, \dots, m\}$. Then, $z_p < 1$ if and only if $p(1) > 0$.*

Proof. See the technical report [2, Appendix F]. □

Step 4: Linear convergence rate (proof of Theorem 9)

Our path to prove linear convergence rate passes through Theorem 23: we first cast the set of inequalities (41) into a system in the form (43), and then study the spectral properties of the resulting coefficient matrix.

Given $\gamma < 1/L$, define $\mathcal{L}(\gamma) \triangleq 1 - \tau\eta^2\gamma$; and choose $\lambda \in \mathbb{R}$ such that

$$\max(\rho, \mathcal{L}(\gamma)) < \lambda < 1. \tag{44}$$

Note that $\mathcal{L}(\gamma) < 1$, as $\gamma < 1/L$; hence (44) is nonempty.

Applying Lemma 21 and Lemma 22 to the set of inequalities (41) with $\gamma^k \equiv \gamma$, we obtain the system (42) at the top of the page. By Theorem 23, to prove the desired linear convergence rate, it is sufficient to show that $\rho(\mathbf{K}) < 1$. The characteristic polynomial $p_{\mathbf{K}}(t)$ of $\mathbf{T}$ satisfies the conditions of Lemma 24; hence $\rho(\mathbf{K}) < 1$ if and only if $p_{\mathbf{K}}(1) > 0$, that is,

$$\begin{aligned} &\left(\left(1 + \frac{L\gamma}{\lambda - \mathcal{L}(\gamma)}\right) \frac{b_2}{\lambda - \rho} + b_1 + \frac{Lb_2\gamma}{\lambda - \mathcal{L}(\gamma)}\right) \frac{C_2\gamma}{\lambda - \rho} \\ &+ \left(1 + \frac{L\gamma}{\lambda - \mathcal{L}(\gamma)}\right) \frac{b_2\gamma}{\lambda - \rho} \triangleq \mathfrak{B}(\lambda;\gamma) < 1. \end{aligned} \tag{45}$$

By the continuity of $\mathfrak{B}(\lambda;\gamma)$ and (44), $\mathfrak{B}(1;\gamma) < 1$ is sufficient to claim the existence of some $\lambda \in (\max(\rho, \mathcal{L}(\gamma)), 1)$ such that $\mathfrak{B}(\lambda;\gamma) < 1$. Hence, setting $\mathfrak{B}(1;\gamma) < 1$, yields $0 < \gamma < \bar{\gamma}_1$, with

$$\bar{\gamma}_1 \triangleq \frac{\tau\eta^2(1-\rho)^2}{(\tau\eta^2 + L)b_2(C_2 + 1 - \rho) + (b_1\tau\eta^2 + Lb_2)C_2(1-\rho)}. \tag{46}$$

It is easy to check that $\bar{\gamma}_1 < 1/L$. Therefore, $0 < \gamma < \bar{\gamma}_1$ is sufficient for $E_{\mathbf{c}}^k, E_{\mathbf{t}}^k, E_{\mathbf{z}}^k, E_{\mathbf{o}}^k$ to vanish with an R-Linear rate. The desired result, $\left|x_i^k - x^\star\right| = \mathcal{O}(\lambda^k)$, $i \in \mathcal{V}$, follows readily from $E_{\mathbf{c}}^k = \mathcal{O}(\lambda^k)$ and $E_{\mathbf{o}}^k = \mathcal{O}(\lambda^k)$. The explicit expression of the rate λ, as in (13), is derived in Appendix C.

VIII. ASY-SONATA: Proof of Theorems 10 and 11

Through the section, we use the same notation as in Sec.VII.

A. Preliminaries

We begin establishing a connection between the merit function M_F defined in (14) and the error quantities $E_{\mathbf{c}}^k$, $E_{\mathbf{t}}^k$, and $E_{\mathbf{z}}^k$, defined in (38), (39), and (40) respectively.

Lemma 25. *The merit function M_F satisfies*

$$M_F(\mathbf{x}^k) \leq C_3 (E_{\mathbf{c}}^k)^2 + 3\eta^{-2} \left((E_{\mathbf{t}}^k)^2 + (E_{\mathbf{z}}^k)^2\right), \tag{47}$$

with $C_3 \triangleq 3C_L^2 I + \frac{3L^2}{I} + 6C_L L + 4$.

Proof. Define $\mathbf{J} \triangleq (1/I) \cdot \mathbf{1}\mathbf{1}^\top$ and $\bar{x}^k \triangleq (1/I) \cdot \mathbf{1}^\top \mathbf{x}^k$; and recall the definition of ξ_i^k (cf. Lemma 15) and that $x_{\boldsymbol{\psi}}^{k+1} = x_{\boldsymbol{\psi}}^k - \gamma^k \psi_{i^k}^k z_{i^k}^k$. [cf. (37)]. We have

$$\begin{aligned} M_F(\mathbf{x}^k) &\leq \left|\nabla F(\bar{x}^k)\right|^2 + 2\left\|\mathbf{x}^k - \mathbf{1}x_{\boldsymbol{\psi}}^k\right\|^2 + 2\left\|\mathbf{J}\left(\mathbf{1}x_{\boldsymbol{\psi}}^k - \mathbf{x}^k\right)\right\|^2 \\ &\leq \left|\nabla F(\bar{x}^k)\right|^2 + 4\left\|\mathbf{x}^k - \mathbf{1}x_{\boldsymbol{\psi}}^k\right\|^2. \end{aligned} \tag{48}$$

We bound now $\left|\nabla F(\bar{x}^k)\right|$; we have

$$\begin{aligned} &\left|\nabla F(\bar{x}^k)\right| \leq \left|\nabla F(x_{\boldsymbol{\psi}}^k)\right| + L\left|\bar{x}^k - x_{\boldsymbol{\psi}}^k\right| \\ &\leq \left|\nabla F(x_{\boldsymbol{\psi}}^k) - \bar{g}^k\right| + \left|\bar{g}^k - (\xi_{i^k}^{k-1})^{-1} z_{i^k}^k\right| + (\xi_{i^k}^{k-1})^{-1}\left|z_{i^k}^k\right| \\ &\quad + \frac{L}{\sqrt{I}}\left\|\mathbf{J}\left(\mathbf{x}^k - \mathbf{1}x_{\boldsymbol{\psi}}^k\right)\right\| \\ &\leq \left(C_L\sqrt{I} + \frac{L}{\sqrt{I}}\right) E_{\mathbf{c}}^k + \eta^{-1} E_{\mathbf{t}}^k + \eta^{-1} E_{\mathbf{z}}^k, \end{aligned} \tag{49}$$

where in the last inequality we used $\xi_{i^k}^k \geq \eta$ for all k (cf. Lemma 15) and $\|\mathbf{J}(\mathbf{x}^k - \mathbf{1}x_{\boldsymbol{\psi}}^k)\| \leq E_{\mathbf{c}}^k$.

Eq. (47) follows readily from (48) and (49). □

Our ultimate goal is to show that the RHS of (47) is summable. To do so, we need two further results, Proposition 26 and Lemma 27 below. Proposition 26 establishes a connection between $F(x_{\boldsymbol{\psi}}^{k+1})$ and $E_{\mathbf{c}}^k$, $E_{\mathbf{t}}^k$, and $E_{\mathbf{z}}^k$.

Proposition 26. *In the above setting, there holds: $k \in \mathbb{N}_0$,*

$$\begin{bmatrix} |E_{\mathbf{z}}|^{\lambda,N} \\ |E_{\mathbf{c}}|^{\lambda,N} \\ |E_{\mathbf{t}}|^{\lambda,N} \\ |E_{\mathbf{o}}|^{\lambda,N} \end{bmatrix} \preccurlyeq \underbrace{\begin{bmatrix} 0 & b_1 & 1 & L \\ \frac{C_2\gamma}{\lambda-\rho} & 0 & 0 & 0 \\ \frac{b_2\gamma}{\lambda-\rho} & \frac{b_2}{\lambda-\rho} & 0 & 0 \\ 0 & \frac{b_2\gamma}{\lambda-\mathcal{L}(\gamma)} & \frac{\gamma}{\lambda-\mathcal{L}(\gamma)} & 0 \end{bmatrix}}_{\triangleq \mathbf{K}} \begin{bmatrix} |E_{\mathbf{z}}|^{\lambda,N} \\ |E_{\mathbf{c}}|^{\lambda,N} \\ |E_{\mathbf{t}}|^{\lambda,N} \\ |E_{\mathbf{o}}|^{\lambda,N} \end{bmatrix} + \begin{bmatrix} 0 \\ \left(1+\frac{C_2}{\lambda}\right)E_{\mathbf{c}}^0 \\ \frac{C_0\|\mathbf{g}^0\|}{\lambda} + E_{\mathbf{t}}^0 \\ \frac{1+\lambda}{\lambda}E_{\mathbf{o}}^0 \end{bmatrix}, \quad b_1 \triangleq C_L\sqrt{I}, \quad b_2 \triangleq 3C_0C_L. \tag{42}$$

$$\begin{aligned} F(x_{\boldsymbol{\psi}}^{k+1}) &\le F(x_{\boldsymbol{\psi}}^0) + \frac{1}{2}\left(L+\alpha^{-1}+\beta^{-1}\right)\sum_{t=0}^{k}(E_{\mathbf{z}}^t)^2(\gamma^t)^2 \\ &- \eta\sum_{t=0}^{k}(E_{\mathbf{z}}^t)^2\gamma^t + \frac{\alpha}{2}C_L^2 I\sum_{t=0}^{k}(E_{\mathbf{c}}^t)^2 + \frac{\beta}{2}\eta^{-2}\sum_{t=0}^{k}(E_{\mathbf{t}}^t)^2, \end{aligned} \tag{50}$$

where α and β are two arbitrary positive constants.

Proof. By descent lemma, we get

$$\begin{aligned} &F(x_{\boldsymbol{\psi}}^{k+1}) \le \\ &F(x_{\boldsymbol{\psi}}^k) + \gamma^k\psi_{i^k}^k\left\langle\nabla F(x_{\boldsymbol{\psi}}^k), -z_{i^k}^k\right\rangle + \frac{L(\gamma^k\psi_{i^k}^k)^2}{2}\left|z_{i^k}^k\right|^2 \\ &\le F(x_{\boldsymbol{\psi}}^k) + \frac{L\gamma^{k^2}}{2}\left|z_{i^k}^k\right|^2 + \gamma^k\psi_{i^k}^k\left\langle(\xi_{i^k}^{k-1})^{-1}z_{i^k}^k, -z_{i^k}^k\right\rangle \\ &+ \gamma^k\psi_{i^k}^k\left\langle\nabla F(x_{\boldsymbol{\psi}}^k) - \bar{g}^k, -z_{i^k}^k\right\rangle \\ &+ \gamma^k\psi_{i^k}^k\left\langle\bar{g}^k - (\xi_{i^k}^{k-1})^{-1}z_{i^k}^k, -z_{i^k}^k\right\rangle \\ &\le F(x_{\boldsymbol{\psi}}^k) + \frac{L\gamma^{k^2}}{2}\left|z_{i^k}^k\right|^2 - \gamma^k\eta\left|z_{i^k}^k\right|^2 \\ &+ \gamma^k C_L\sum_{j=1}^{I}\left|x_{\boldsymbol{\psi}}^k - x_j^k\right|\left|z_{i^k}^k\right| + \gamma^k\eta^{-1}E_{\mathbf{t}}^k\left|z_{i^k}^k\right| \\ &\le F(x_{\boldsymbol{\psi}}^k) + \frac{L\gamma^{k^2}}{2}\left|z_{i^k}^k\right|^2 - \gamma^k\eta\left|z_{i^k}^k\right|^2 \\ &+ \gamma^k C_L\sqrt{I}E_{\mathbf{c}}^k\left|z_{i^k}^k\right| + \gamma^k\eta^{-1}E_{\mathbf{t}}^k\left|z_{i^k}^k\right| \\ &\le F(x_{\boldsymbol{\psi}}^k) + \frac{L\gamma^{k^2}}{2}\left|z_{i^k}^k\right|^2 - \gamma^k\eta\left|z_{i^k}^k\right|^2 + \frac{\alpha}{2}C_L^2 I(E_{\mathbf{c}}^k)^2 \\ &+ \frac{\alpha^{-1}}{2}\left|z_{i^k}^k\right|^2\gamma^{k^2} + \frac{\beta}{2}\eta^{-2}(E_{\mathbf{t}}^k)^2 + \frac{\beta^{-1}}{2}\left|z_{i^k}^k\right|^2(\gamma^k)^2 \\ &\le F(x_{\boldsymbol{\psi}}^k) + \frac{1}{2}\left(L+\alpha^{-1}+\beta^{-1}\right)(E_{\mathbf{z}}^k)^2(\gamma^k)^2 \\ &- \eta(E_{\mathbf{z}}^k)^2\gamma^k + \frac{\alpha}{2}C_L^2 I(E_{\mathbf{c}}^k)^2 + \frac{\beta}{2}\eta^{-2}(E_{\mathbf{t}}^k)^2. \end{aligned}$$

Applying the above inequality inductively one gets (50). □

The last result we need is a bound of $\sum_{t=0}^{k}(E_{\mathbf{c}}^t)^2$ and $\sum_{t=0}^{k}(E_{\mathbf{t}}^t)^2$ in (50) in terms of $\sum_{t=0}^{k}(E_{\mathbf{z}}^t)^2(\gamma^t)^2$.

Lemma 27. *Define*

$$\varrho_{\mathbf{c}} \triangleq \frac{2C_2^2}{(1-\rho)^2} \quad \textit{and} \quad \varrho_{\mathbf{t}} \triangleq \frac{36\left(C_0C_L\right)^2\left(2C_2^2+(1-\rho)^2\right)}{(1-\rho)^4}.$$

The following holds: $k \in \mathbb{N}$,

$$\begin{aligned} \sum_{t=0}^{k}(E_{\mathbf{c}}^t)^2 &\le c_{\mathbf{c}} + \varrho_{\mathbf{c}}\sum_{t=0}^{k}(E_{\mathbf{z}}^t)^2(\gamma^t)^2, \\ \sum_{t=0}^{k}(E_{\mathbf{t}}^t)^2 &\le c_{\mathbf{t}} + \varrho_{\mathbf{t}}\sum_{t=0}^{k}(E_{\mathbf{z}}^t)^2(\gamma^t)^2, \end{aligned} \tag{51}$$

where $c_{\mathbf{c}}$ and $c_{\mathbf{t}}$ are some positive constants.

Proof. The proof follows from Proposition 19 and Lemma 28 below, which is a variant of [28] (its proof is thus omitted).

Lemma 28. *Let $\{u^k\}_{k=0}^{\infty}, \{v_i^k\}_{k=0}^{\infty}, i = 1, \ldots, m$, be nonnegative sequences; $\lambda \in (0,1)$; and $R_0 \in \mathbb{R}_+$ such that*

$$u^{k+1} \le R\lambda^k + \sum_{l=0}^{k}\lambda^{k-l}v^l.$$

Then, there holds: $k \in \mathbb{N}$,

$$\sum_{l=0}^{k}(u^l)^2 \le (u^0)^2 + \frac{2R^2}{1-\lambda^2} + \frac{2}{(1-\lambda)^2}\sum_{l=0}^{k}(v^l)^2.$$

□

Using (51) in (50), we finally obtain

$$\sum_{t=0}^{k}(E_{\mathbf{z}}^t)^2\gamma^t(\eta - \gamma^t C_4(\alpha,\beta)) \le F(x_{\boldsymbol{\psi}}^0) - F^{\inf} + C_5(\alpha,\beta) \tag{52}$$

with $C_4(\alpha,\beta) \triangleq (1/2)\left(L+\alpha^{-1}+\beta^{-1}+C_L^2 I\alpha\varrho_{\mathbf{c}}+\eta^{-2}\beta\varrho_{\mathbf{t}}\right)$ and $C_5(\alpha,\beta) = (1/2)\left(C_L^2 I\alpha c_{\mathbf{c}} + \eta^{-2}\beta c_{\mathbf{t}}\right)$; and $F^{\inf} > -\infty$ is the lower bound of F.

We are now ready to prove Theorems 10 and 11.

B. Proof of Theorem 10

Set $\gamma^k \equiv \gamma$, for all $k \in \mathbb{N}_0$. By (52), one infers that $\sum_{t=0}^{\infty}E_{\mathbf{z}}^{t\,2} < \infty$ if γ satisfies $0 < \gamma < \bar{\gamma}_2(\alpha,\beta)$, with $\bar{\gamma}_2(\alpha,\beta) \triangleq \eta/C_4(\alpha,\beta)$.Note that $\bar{\gamma}_2(\alpha,\beta)$ is maximized setting $\alpha = \alpha^\star = \left(C_L\sqrt{I\varrho_{\mathbf{c}}}\right)^{-1}$ and $\beta = \beta^\star = \eta\varrho_{\mathbf{t}}^{-1/2}$, resulting in

$$\bar{\gamma}_2(\alpha^\star,\beta^\star) = (2\eta)/(L + 2C_L\sqrt{I\varrho_{\mathbf{c}}} + 2\eta^{-1}\sqrt{\varrho_{\mathbf{t}}}). \tag{53}$$

Let $0 < \gamma < \bar{\gamma}_2(\alpha^\star,\beta^\star)$. Given $\delta > 0$, let T_δ be the first iteration $k \in \mathbb{N}_0$ such that $M_F(\mathbf{x}^k) \le \delta$. Then we have

$$\begin{aligned} T_\delta\cdot\delta &< \sum_{k=0}^{T_\delta-1}M_F(\mathbf{x}^k) \le \sum_{k=0}^{\infty}M_F(\mathbf{x}^k) \\ &\overset{(47)}{\le} C_3\sum_{k=0}^{\infty}(E_{\mathbf{c}}^k)^2 + 3\eta^{-2}\sum_{k=0}^{\infty}\left((E_{\mathbf{t}}^k)^2 + (E_{\mathbf{z}}^k)^2\right) \\ &\overset{(51),(52)}{\le} \frac{F(x_{\boldsymbol{\psi}}^0) - F^{\inf} + C_5(\alpha^\star,\beta^\star)}{\gamma(\eta - \gamma C_4(\alpha^\star,\beta^\star))}\cdot C_6 + C_7 < \infty \end{aligned}$$

where $C_6 \triangleq C_3\varrho_{\mathbf{c}}(\gamma)^2 + 3\eta^{-2}\left(\varrho_{\mathbf{t}}(\gamma)^2 + 1\right)$ and C_7 is some constant. Therefore, $T_\delta = \mathcal{O}(1/\delta)$.

C. Proof of Theorem 11.

We begin showing that the step-size sequence $\{\gamma^t\}_{t\in\mathbb{N}_0}$ induced by the local step-size sequence $\{\alpha^t\}_{t\in\mathbb{N}_0}$ and the asynchrony mechanism satisfying Assumption 6 is nonsummable. The proof is straightforward and is thus omitted.

Lemma 29. *Let $\{\gamma^t\}_{t\in\mathbb{N}_0}$ be the global step-size sequence resulted from Algorithm 2, under Assumption 6. Then, there hold: $\lim_{t\to\infty}\gamma^t = 0$ and $\sum_{t=0}^{\infty}\gamma^t = \infty$.*

Since $\lim_{t\to\infty}\gamma^t=0$, there exists a sufficiently large $k\in\mathbb{N}$, say $\bar{k}$, such that $\eta-\gamma^k C_4(\alpha^\star,\beta^\star)\geq\eta/2$ for all $k>\bar{k}$. It is not difficult to check that this, together with (52), yields $\sum_{k=0}^{\infty}(E_{\mathbf{z}}^k)^2\gamma^k<\infty$. We can then write

$$\sum_{k=0}^{\infty}M_F(\mathbf{x}^k)\gamma^k \overset{(47)}{\leq} C_3\sum_{k=0}^{\infty}(E_{\mathbf{c}}^k)^2\gamma^k+3\eta^{-2}\sum_{k=0}^{\infty}\left((E_{\mathbf{t}}^k)^2+(E_{\mathbf{z}}^k)^2\right)\gamma^k<C_8, \tag{54}$$

for some finite constant C_8, where in the last inequality we used (51), $\sum_{k=0}^{\infty}(E_{\mathbf{z}}^k)^2\gamma^k<\infty$ and $\lim_{t\to\infty}\gamma^t=0$.

Let $N_\delta\triangleq\inf\left\{k\in\mathbb{N}_0:\sum_{t=0}^{k}\gamma^t\geq C_8/\delta\right\}$. Note that N_δ exists, as $\sum_{k=0}^{\infty}\gamma^k=\infty$ (cf. Lemma 29). Let $T_\delta\triangleq\inf\left\{k\in\mathbb{N}_0:M_F(\mathbf{x}^k)\leq\delta\right\}$. It must be $T_\delta\leq N_\delta$. In fact, suppose by contradiction that $T_\delta>N_\delta$; and thus $M_F(\mathbf{x}^k)>\delta$, for $0\leq k\leq N_\delta$. It would imply $\sum_{k=0}^{N_\delta}M_F(\mathbf{x}^k)\gamma^k>\delta\sum_{k=0}^{N_\delta}\gamma^k\geq\delta\cdot(C_8/\delta)=C_8$, which contradicts (54). This proves (15).

IX. Conclusions

We proposed ASY-SONATA, a distributed asynchronous algorithmic framework for convex and nonconvex (unconstrained, smooth) multi-agent problems, over digraphs. The algorithm is robust against uncoordinated agents' activation and (communication/computation) (time-varying) delays. When employing a constant step-size, ASY-SONATA achieves a linear rate for strongly convex objectives–matching the rate of a centralized gradient algorithm–and sublinear rate for (non)convex problems. Sublinear rate is also established when agents employ uncoordinated diminishing step-sizes, which is more realistic in a distributed setting. To the best of our knowledge, ASY-SONATA is the first distributed algorithm enjoying the above properties, in the general asynchronous setting described in the paper.

Appendix

A. Proof of Lemma 21

Fix $N\in\mathbb{N}$, and let k such that $1\leq k+1\leq N$. We have:

$$\begin{aligned}\frac{u^{k+1}}{\lambda^{k+1}}&\leq\frac{R_0}{\lambda}\left(\frac{\lambda_0}{\lambda}\right)^k+\sum_{i=1}^{m}\frac{R_i}{\lambda}\sum_{l=0}^{k}\left(\frac{\lambda_i}{\lambda}\right)^{k-l}\frac{v_i^l}{\lambda^l}\\&\leq\frac{R_0}{\lambda}+\sum_{i=1}^{m}\frac{R_i}{\lambda}\left|v_i\right|^{\lambda,N}\sum_{l=0}^{k}\left(\frac{\lambda_i}{\lambda}\right)^{k-l}\\&\leq\frac{R_0}{\lambda}+\sum_{i=1}^{m}\frac{R_i}{\lambda-\lambda_i}\left|v_i\right|^{\lambda,N}.\end{aligned}$$

Hence,

$$\begin{aligned}\left|u\right|^{\lambda,N}&\leq\max\left(u_0,\frac{R_0}{\lambda}+\sum_{i=1}^{m}\frac{R_i}{\lambda-\lambda_i}\left|v_i\right|^{\lambda,N}\right)\\&\leq u^0+\frac{R_0}{\lambda}+\sum_{i=1}^{m}\frac{R_i}{\lambda-\lambda_i}\left|v_i\right|^{\lambda,N}.\quad\square\end{aligned}$$

B. Proof of Theorem 23

From [46, Ch. 5.6], we know that if $\rho(\mathbf{T})<1$, then $\lim_{k\to\infty}\mathbf{T}^k=0$, the series $\sum_{k=0}^{\infty}\mathbf{T}^k$ converges (wherein we define $\mathbf{T}^0\triangleq\mathbf{I}$), $\mathbf{I}-\mathbf{T}$ is invertible and $\sum_{k=0}^{\infty}\mathbf{T}^k=(\mathbf{I}-\mathbf{T})^{-1}$.

Given $N\in\mathbb{N}$, using (43) recursively, yields: $\mathbf{u}^{\lambda,N}\leq\mathbf{T}\mathbf{u}^{\lambda,N}+\boldsymbol{\beta}\leq\mathbf{T}\left(\mathbf{T}\mathbf{u}^{\lambda,N}+\boldsymbol{\beta}\right)+\boldsymbol{\beta}=\mathbf{T}^2\mathbf{u}^{\lambda,N}+(\mathbf{T}+\mathbf{I})\,\boldsymbol{\beta}\leq\cdots\leq\mathbf{T}^\ell\mathbf{u}^{\lambda,N}+\sum_{k=0}^{\ell-1}\mathbf{T}^k\boldsymbol{\beta}$, for any $\ell\in\mathbb{N}$. Let $\ell\to\infty$, we get $\mathbf{u}^{\lambda,N}\leq(\mathbf{I}-\mathbf{T})^{-1}\boldsymbol{\beta}$. Since this holds for any given $N\in\mathbb{N}$, we have $\mathbf{u}^\lambda\leq(\mathbf{I}-\mathbf{T})^{-1}\boldsymbol{\beta}$. Hence, $\mathbf{u}^\lambda$ is bounded, and thus each u_i^k vanishes at an R-linear rate $\mathcal{O}(\lambda^k)$. $\square$

C. Proof of the rate decay (13) (Theorem 9)

Let $\lambda\geq\mathcal{L}(\gamma)+\epsilon\gamma$, with $\epsilon>0$ to be properly chosen. Then,

$$\begin{aligned}\mathfrak{B}(\lambda;\gamma)\leq&\left(1+\frac{L}{\epsilon}\right)\frac{b_2\gamma}{\lambda-\rho}\\&+\left(\left(1+\frac{L}{\epsilon}\right)\frac{b_2}{\lambda-\rho}+b_1+\frac{Lb_2}{\epsilon}\right)\frac{C_2\gamma}{\lambda-\rho}.\end{aligned}\tag{55}$$

Using $\lambda-\rho<1$, a sufficient condition for the RHS of the above inequality being strictly less than 1 is

$$\left(b_1C_2+\frac{Lb_2C_2}{\epsilon}+\left(1+\frac{L}{\epsilon}\right)b_2(1+C_2)\right)\gamma\leq(\lambda-\rho)^2.\tag{56}$$

Now set $\epsilon=(\tau\eta^2)/2$. Since the RHS of the above inequality can be arbitrarily close to $(1-\rho)^2$, an upper bound of γ is

$$\hat{\gamma}_2\triangleq(1-\rho)^2\Big/\underbrace{\left(b_1C_2+\frac{2Lb_2C_2}{\tau\eta^2}+\left(1+\frac{2L}{\tau\eta^2}\right)b_2(1+C_2)\right)}_{\triangleq J_1}.$$

According to $\lambda\geq\mathcal{L}(\gamma)+\epsilon\gamma$ and (56), we get

$$\lambda=\max\left(1-\frac{\tau\eta^2\gamma}{2},\quad\rho+\sqrt{J_1\gamma}\right).\tag{57}$$

Notice that when γ goes from 0 to $\hat{\gamma}_2$, the first argument inside the max operator decreases from 1 to $1-(\tau\eta^2\hat{\gamma}_2)/2$, while the second argument increases from ρ to 1. Letting $1-\frac{\tau\eta^2\gamma}{2}=\rho+\sqrt{J_1\gamma}$, we get the solution as $\hat{\gamma}_1=\left(\frac{\sqrt{J_1+2\tau\eta^2(1-\rho)}-\sqrt{J_1}}{\tau\eta^2}\right)^2$. The expression of λ as in (13) follows readily. $\square$

References

[1] Y. Tian, Y. Sun, and G. Scutari, "Asy-sonata: Achieving linear convergence for distributed asynchronous optimization," in *Proc. of Allerton 2018*, Oct. 3-5.

[2] ——, "Achieving linear convergence in distributed asynchronous multi-agent optimization," *arXiv:1803.10359*, Mar. 2018.

[3] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: numerical methods*. Englewood Cliffs, NJ: Prentice-Hall, 1989.

[4] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE Trans. Auto. Contr.*, vol. 56, no. 6, pp. 1337–1351, 2011.

[5] X. Zhao and A. H. Sayed, "Asynchronous adaptation and learning over networks–Part I/Part II/Part III: Modeling and stability analysis/Performance analysis/Comparison analysis," *IEEE Trans. Signal Process.*, vol. 63, no. 4, pp. 811–858, 2015.

[6] S. Kumar, R. Jain, and K. Rajawat, "Asynchronous optimization over heterogeneous networks via consensus admm," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 3, no. 1, pp. 114–129, 2017.

[7] M. Eisen, A. Mokhtari, and A. Ribeiro, "Decentralized quasi-newton methods," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2613–2628, 2017.

[8] N. Bof, R. Carli, G. Notarstefano, L. Schenato, and D. Varagnolo, "Newton-raphson consensus under asynchronous and lossy communications for peer-to-peer networks," *arXiv:1707.09178*, 2017.

[9] Z. Peng, Y. Xu, M. Yan, and W. Yin, "Arock: an algorithmic framework for asynchronous parallel coordinate updates," *SIAM J. Sci. Comput.*, vol. 38, no. 5, pp. A2851–A2879, 2016.

[10] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed, "Decentralized consensus optimization with asynchrony and delays," *IEEE Trans. Signal Inf. Process. Netw.*, vol. PP, no. 99, 2017.

[11] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Auto. Contr.*, vol. 31, no. 9, pp. 803–812, 1986.

[12] J. Liu and S. J. Wright, "Asynchronous stochastic coordinate descent: Parallelism and convergence properties," *SIAM J. Optim.*, vol. 25, no. 1, pp. 351–376, 2015.

[13] L. Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari, "Asynchronous parallel algorithms for nonconvex optimization," *Math. Prog.*, June 2019.

[14] F. Niu, B. Recht, C. Re, and S. J. Wright, "Hogwild: a lock-free approach to parallelizing stochastic gradient descent," in *Proc. of NIPS 2011*, pp. 693–701.

[15] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Proc. of NIPS 2015*, pp. 2719–2727.

[16] I. Notarnicola and G. Notarstefano, "Asynchronous distributed optimization via randomized dual proximal gradient," *IEEE Trans. Auto. Contr.*, vol. 62, no. 5, pp. 2095–2106, 2017.

[17] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Trans. Auto. Contr.*, vol. 63, no. 2, pp. 434–448, 2017.

[18] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *Proc. of CDC 2013*, pp. 3671–3676.

[19] E. Wei and A. Ozdaglar, "On the $o(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers," in *Proc. of GlobalSIP 2013*, pp. 551–554.

[20] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization," *IEEE Trans. Auto. Contr.*, vol. 61, no. 10, pp. 2947–2957, 2016.

[21] H. Wang, X. Liao, T. Huang, and C. Li, "Cooperative distributed optimization in multiagent networks with delays," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 45, no. 2, pp. 363–369, 2015.

[22] J. Li, G. Chen, Z. Dong, and Z. Wu, "Distributed mirror descent method for multi-agent optimization with delay," *Neurocomputing*, vol. 177, pp. 643–650, 2016.

[23] K. I. Tsianos and M. G. Rabbat, "Distributed dual averaging for convex optimization under communication delays," in *Proc. of ACC 2012*, pp. 1067–1072.

[24] ——, "Distributed consensus and optimization under communication delays," in *Proc. of Allerton 2011*, pp. 974–982.

[25] P. Lin, W. Ren, and Y. Song, "Distributed multi-agent optimization subject to nonidentical constraints and communication delays," *Automatica*, vol. 65, pp. 120–131, 2016.

[26] T. T. Doan, C. L. Beck, and R. Srikant, "Impact of communication delays on the convergence rate of distributed optimization algorithms," *arXiv:1708.03277*, 2017.

[27] P. Di Lorenzo and G. Scutari, "Distributed nonconvex optimization over networks," in *Proc. of IEEE CAMSAP 2015*, Dec. 2015.

[28] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes," in *Proc. of CDC 2015*, Dec., pp. 2055–2060.

[29] P. Di Lorenzo and G. Scutari, "Next: In-network nonconvex optimization," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 2, no. 2, pp. 120–136, 2016.

[30] C. N. Hadjicostis, N. H. Vaidya, and A. D. Dominguez-Garcia, "Robust distributed average consensus via exchange of running sums," *IEEE Trans. Auto. Contr.*, vol. 31, no. 6, pp. 1492–1507, 2016.

[31] A. Nedić and A. Ozdaglar, "Convergence rate for consensus with delays," *J. Glob. Optim.*, vol. 47, no. 3, pp. 437–456, 2010.

[32] P. Lin and W. Ren, "Constrained consensus in unbalanced networks with communication delays," *IEEE Trans. Auto. Contr.*, vol. 59, no. 3, pp. 775–781, 2013.

[33] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[34] Y. Sun, G. Scutari, and D. Palomar, "Distributed nonconvex multiagent optimization over time-varying networks," in *Proc. of Asilomar 2016*. IEEE, pp. 788–794.

[35] G. Scutari and Y. Sun, "Distributed nonconvex constrained optimization over time-varying digraphs," *Math. Prog.*, vol. 176, no. 1-2, pp. 497–544, July 2019.

[36] Y. Sun, A. Daneshmand, and G. Scutari, "Convergence rate of distributed optimization algorithms based on gradient tracking," *arXiv:1905.02637*, 2019.

[37] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. of FOCS 2003*. IEEE, pp. 482–491.

[38] N. Bof, R. Carli, and L. Schenato, "Average consensus with asynchronous updates and unreliable communication," in *Proc. of the IFAC Word Congress 2017*, pp. 601–606.

[39] T. S. Rappaport, *Wireless Communications: Principles & Practice*. Prentice Hall, 2002.

[40] S. M. Kay, *Fundamentals of Statistical Signal Processing–Estimation Theory*. Prentice Hall, 1993.

[41] L. Cannelli, F. Facchinei, and G. Scutari, "Multi-agent asynchronous nonconvex large-scale optimization," in *Proc. of IEEE CAMSAP 2017*, pp. 1–5.

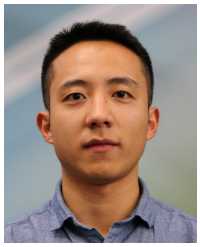
[42] L. Zhao, M. Mammadov, and J. Yearwood, "From convex to nonconvex: a loss function analysis for binary classification," in *2010 IEEE ICDM Workshops*, pp. 1281–1288.

[43] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

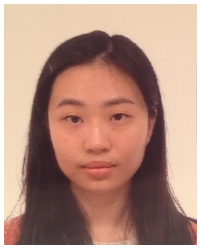
[44] M. Assran and M. Rabbat, "Asynchronous subgradient-push," *arXiv:1803.08950*, 2018.

[45] J. Zhang and K. You, "Asyspa: An exact asynchronous algorithm for convex optimization over digraphs," *arXiv:1808.04118*, 2018.

[46] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 1990.



Ye Tian received the B.S. degree in mathematics from Nanjing University, Nanjing, China, in 2016. He is currently pursuing his Ph.D. degree at the School of Industrial Engineering, Purdue University. His research interests include optimization algorithms and their applications in machine learning.



Ying Sun received the B.E. degree in electronic information from the Huazhong University of Science and Technology, Wuhan, China, in 2011, and the Ph.D. degree from the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology, Hong Kong, in 2016. She is currently a postdoctoral researcher with the School of Industrial Engineering, Purdue University. Her research interests include optimization algorithms, statistical signal processing, and machine learning.

Gesualdo Scutari (S'05-M'06-SM'11) received the Electrical Engineering and Ph.D. degrees (both with honors) from the University of Rome "La Sapienza," Rome, Italy, in 2001 and 2005, respectively. He is the Thomas and Jane Schmidt Rising Star Associate Professor with the School of Industrial Engineering, Purdue University, West Lafayette, IN, USA. He had previously held several research appointments, namely, at the University of California at Berkeley, Berkeley, CA, USA; Hong Kong University of Science and Technology, Hong Kong; and University of Illinois at Urbana-Champaign, Urbana, IL, USA. His research interests include continuous and distributed optimization, equilibrium programming, and their applications to signal processing and machine learning. He is a Senior Area Editor of the IEEE Transactions On Signal Processing and an Associate Editor of the IEEE Transactions on Signal and Information Processing over Networks. He served on the IEEE Signal Processing Society Technical Committee on Signal Processing for Communications (SPCOM). He was the recipient of the 2006 Best Student Paper Award at the IEEE ICASSP 2006, the 2013 NSF CAREER Award, the 2015 Anna Maria Molteni Award for Mathematics and Physics, and the 2015 IEEE Signal Processing Society Young Author Best Paper Award.