Asynchronous Optimization Over Graphs: Linear Convergence Under Error Bound Conditions

Loris Cannelli, *Member, IEEE*, Francisco Facchinei, Gesualdo Scutari, *Senior Member, IEEE*, and Vyacheslav Kungurtsev, *Member, IEEE*

***Abstract*—We consider convex and nonconvex constrained optimization with a partially separable objective function: Agents minimize the sum of local objective functions, each of which is known only by the associated agent and depends on the variables of that agent and those of a few others. This partitioned setting arises in several applications of practical interest. We propose what is, to the best of our knowledge, the first distributed, asynchronous algorithm with rate guarantees for this class of problems. When the objective function is nonconvex, the algorithm provably converges to a stationary solution at a sublinear rate whereas linear rate is achieved under the renowned Luo-Tseng error bound condition (which is less stringent than strong convexity). Numerical results on matrix completion and LASSO problems show the effectiveness of our method.**

***Index Terms*—Asynchronous algorithms, error bounds, linear rate, multiagent systems, nonconvex optimization.**

I. Introduction

WE STUDY distributed, nonsmooth, nonconvex optimization with a partially separable sum-cost function. Specifically, consider a set of N agents, each of them controlling/updating a subset of the n variables $\mathbf{x} \in \mathbb{R}^n$. Partitioning $\mathbf{x} = (\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T)^T$, $\mathbf{x}_i \in \mathbb{R}^{n_i}$ is the block of variables owned by agent $i \in \mathcal{N} \triangleq \{1, \ldots, N\}$, with $\sum_i n_i = n$. All agents cooperatively aim at solving the following problem:

$$\min_{\mathbf{x}_i \in \mathcal{X}_i, i \in \mathcal{N}} V(\mathbf{x}) \triangleq \underbrace{\sum_{i=1}^N f_i(\mathbf{x}_{\mathcal{N}_i})}_{\triangleq F(\mathbf{x})} + \underbrace{\sum_{i=1}^N g_i(\mathbf{x}_i)}_{\triangleq G(\mathbf{x})} \tag{P}$$

where $\mathcal{N}_i$ denotes a small subset of $\mathcal{N}$ including the index i and $\mathbf{x}_{\mathcal{N}_i} \triangleq [\mathbf{x}_j]_{j \in \mathcal{N}_i}$ denotes the column vector containing the blocks of $\mathbf{x}$ indexed by $\mathcal{N}_i$; $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is a closed convex set; f_i is a smooth (nonconvex) function that depends only on $\mathbf{x}_{\mathcal{N}_i}$; and g_i is a convex (nonsmooth) function, instrumental to encode structural constraints on the solution, such as sparsity. Both f_i and g_i are assumed to be known only by agent i.

The above formulation is motivated by a variety of applications of practical interest. For instance, loss functions arising from many machine learning problems have the "sparse" pattern of V in (P): n and N are both very large but each f_i depends only on a small number of components of $\mathbf{x}$, i.e., each subvector $\mathbf{x}_{\mathcal{N}_i}$ contains just a few components of $\mathbf{x}$. The same partitioned structure in (P) is suitable also to model networked systems wherein agents are connected through a physical communication network and can communicate only with their immediate neighbors. In this setting, often $\mathcal{N}_i$ represents the set of neighbors of agent i (including agent i itself). Examples of such applications include resource allocation problems and network utility maximization [1], state estimation in power networks [2], cooperative localization in wireless networks [3], and map building in robotic networks. Some concrete instances of Problem (P) are discussed in Section II.

A. Major Contributions

We focus on the design of distributed, asynchronous algorithms for (P), in the following sense: i) Agents can update their block variables at any time, without any coordination; and ii) when updating their own variables, agents can use a delayed out-of-sync information from the others. No constraint is imposed on the delay profiles: Delays can be arbitrary, possibly time-varying (but bounded). This model captures 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, unknown, time-varying delays.

While several forms of asynchrony have been studied in the literature—see Section I-B for an overview of most relevant results—we are not aware of any distributed scheme that is compliant to the asynchronous model (i)–(ii) and tailored to the *partitioned* (nonconvex) distributed formulation (P). This article fills this gap and proposes a general distributed, asynchronous

Manuscript received July 11, 2019; revised May 10, 2020; accepted October 4, 2020. Date of publication October 23, 2020; date of current version September 27, 2021. The work of Loris Cannelli and Gesualdo Scutari was supported in part by the USA NSF under Grant CIF 1719205 and Grant CMMI 1832688 and in part by the ARO under the Grant W911NF1810238. The work of Vyacheslav Kungurtsev was supported by the OP VVV project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics." Recommended by Associate Editor R. M. Jungers. *(Francisco Facchinei and Gesualdo Scutari contributed equally to this work.) (Corresponding author: Gesualdo Scutari.)*

Loris Cannelli is with the Istituto Dalle Molle di studi sull'Intelligenza Artificiale (IDSIA), USI/SUPSI, 6900 Lugano, Switzerland, and also with the School of Industrial Engineering, Purdue University, West-Lafayette, IN 47907 USA (e-mail: loris.cannelli@idsia.ch).

Francisco Facchinei is with the Department of Computer, Control, and Management Engineering, University of Rome La Sapienza, 00185 Rome, Italy (e-mail: facchinei@dis.uniroma1.it).

Gesualdo Scutari is with the School of Industrial Engineering, Purdue University, West-Lafayette, IN 47907 USA (e-mail: gscutari@purdue.edu).

Vyacheslav Kungurtsev is with the Department of Computer Science, Czech Technical University, 16636 Prague, Czech Republic (e-mail: vyacheslav.kungurtsev@fel.cvut.cz).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2020.3033490

algorithmic framework for convex and nonconvex instances of (P). The algorithm builds on Successive Convex Approximation (SCA) techniques: Agents solve asynchronously [in the sense (i) and (ii) above] strongly convex approximations of the original problem (P) by using (possibly) outdated information on the variables and the gradients of the other agents. No specific activation mechanism for the agents' updates, coordination, or communication protocol is assumed, but only some mild conditions ensuring that information used in the updates does not become infinitely old. For nonconvex instances of V, we prove that 1) every limit point of the sequence generated by the proposed asynchronous algorithm is a stationary solution of (P); and 2) a suitable measure of stationarity vanishes at a sublinear rate. When V further satisfies the Luo-Tseng error bound condition [4], [5], both the sequence and the objective value converge at an R-linear rate (when V is nonconvex, convergence is to stationary solutions). This error bound condition is weaker than strong convexity and it is satisfied by a variety of problems of interest, such as LASSO, Group LASSO, and Logistic Regression, just to name a few (cf. Section III-A). While linear convergence under error bounds has been proved for many *centralized* algorithms [4], [6]–[9], we are not aware of any such a result in the distributed setting; current works require strong convexity to establish linear rate of synchronous and asynchronous *distributed* algorithms (see, e.g., [10]–[12] and references therein). As a byproduct, our results provide also a positive answer to the open question whether linear convergence could be proved for *distributed* asynchronous algorithms solving highly dimensional empirical risk minimization problems, such as LASSO and Logistic Regression, a fact that was empirically observed but, to our knowledge, never proved.

B. Related Works

Since the seminal work [13], asynchronous parallelism has been applied to several centralized solution methods, including (randomized) block-coordinate descent schemes [6], [13]–[17], and stochastic gradient algorithms [18], [19]. However, those methods are not applicable to Problem (P) , since they would require each agent to know the entire objective function V.

Distributed schemes exploring (some form of) asynchrony have been studied in [20]–[39]; next, we group them based upon the asynchrony features (i) and (ii).

1) *Random Activation and No Delays [20]–[27], [40]:* While substantially different in the form of the updates performed by the agents, these schemes are all asynchronous in the sense of feature (i) only. Agents (or edge-connected agents) are randomly activated, but when performing their computations/updates, they must use the current information from their neighbors. This means that no form of delay is allowed. Furthermore, between two activations, agents must be in idle mode (i.e., able to continuously receive information). Some form of coordination is thus needed to enforce the above conditions. All the schemes in this group but [26] can deal with convex objectives only; and none of the above works provide a convergence rate or complexity analysis.

2) *Synchronous Activation and Delays [28]–[33]:* These schemes consider synchronous activation/updates of the agents, which can tolerate fixed computation delays (e.g., outdated gradient information) [28], [29] or fixed [30], [33] or time-varying [31], [32] communication delays. However delays cannot be arbitrary, but must be such that no loss can ever occur in the network: Every agent's message must reach its intended destination within a finite time interval. Finally, all these algorithms are applicable only to convex problems.

3) *Random/Cyclic Activations and Some Form of Delay [34]–[39], [41]–[44]:* These schemes allow for random [34]–[37], [41] or deterministic uncoordinated [38], [39], [42]–[45] activation of the (edge-based) agents, together with the presence of some form of delay in the updates/computations. Specifically, [34], [35], [38] can handle link failures—the information sent by an agent to its neighbors either gets lost or received with *no delay*—but cannot deal with other forms of delay (e.g., communication delays). In [36], [37], [41] a probabilistic model is assumed whereby agents are randomly activated and update their local variables using possibly delayed information. The model requires that the random variables modeling the activation of the agents are i.i.d and independent of the delay vector used by the agent to perform its update. While this assumption makes the convergence analysis possible, in reality there is a strong dependence of the delays on the activation index, as also noted by the same authors [36], [37] (see [15] for a detailed discussion on this issue and some counter examples). Closer to our setting are the asynchronous methods in [10], [36], [39], [42]–[45]. These models, however, assume that each function f_i depends on the *entire* vector $\mathbf{x}$. As a consequence, a consensus mechanism on all the optimization variables is employed among the agents at each iteration. Because of that, a direct application of these consensus-based algorithms to the partitioned formulation (P) would lead to very inefficient schemes calling for unnecessary computation and communication overheads. Furthermore, the Alternating Direction Method of Multipliers (ADMM)-like schemes [39], [41]–[45] can be implemented only on very specific network architectures, such as star networks or hierarchical topologies with multiple master and worker nodes. Finally, notice that, with the exception of [10], [35], [39], [41]–[45] (resp. [38]), all these schemes are applicable to convex problems (resp. undirected graphs) only, with [34] further assuming that all the functions f_i have the same minimizer.

The rest of the article is organized as follows: Section II discusses some motivating applications. The proposed algorithm is introduced and analyzed in Section III. Finally, numerical results are presented in Section IV.

II. Motivating Examples

We discuss next two instances of Problem (P), which will be also used in our numerical experiments to test our algorithms (cf. Section IV). The first case study is the matrix completion problem—an example of large-scale nonconvex empirical risk minimization. We show how to exploit the sparsity pattern in the data to rewrite the problem in the form (P), so that efficient asynchronous algorithms leveraging multicore architectures can be developed. The second example deals with learning problems from networked data sets; in this setting data are distributed across multiple nodes, whose communication network is modeled as a (directed) graph.

Example #1 –Matrix Completion: The matrix completion problem consists of estimating a low-rank matrix $\mathbf{Z} \in \mathbb{R}^{M\times N}$ from a subset $\Omega \subseteq \{1,\dots,M\} \times \{1,\dots,N\}$ of its entries. Postulating the low-rank factorization $\mathbf{Z} = \mathbf{X}^T\mathbf{Y}$, with $\mathbf{X} \in \mathbb{R}^{r\times M}$ and $\mathbf{Y} \in \mathbb{R}^{r\times N}$, the optimization problem reads [46]:

$$\min_{\substack{\mathbf{X}\in\mathbb{R}^{r\times M}\\ \mathbf{Y}\in\mathbb{R}^{r\times N}}} V(\mathbf{X},\mathbf{Y}) \triangleq \frac{1}{2}\left\|(\mathbf{X}^T\mathbf{Y}-\mathbf{Z})_\Omega\right\|_F^2+\frac{\lambda}{2}\|\mathbf{X}\|_F^2+\frac{\xi}{2}\|\mathbf{Y}\|_F^2, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius norm; $(\cdot)_\Omega$ is the projection operator, defined as $[(\mathbf{X})_\Omega]_{(i,j)} = \mathbf{X}_{(i,j)}$, if $(i,j) \in \Omega$; and $[(\mathbf{X})_\Omega]_{(i,j)} = 0$ otherwise; and $\lambda, \xi > 0$ are regularization parameters. In many applications, the amount of data is so large that storage and processing from a single agent (e.g., core, machine) is not efficient or even feasible. The proposed approach is then to leverage multicore machines by first casting (1) in the form (P), and then employing the parallel asynchronous framework developed in this article.

Consider a distributed environment composed of N agents, and assume that the known entries z_{mn}, $(m,n) \in \Omega$, are partitioned among the agents. This partition along with the sparsity pattern of $(\mathbf{Z})_\Omega$ induce naturally the following splitting of the optimization variables $\mathbf{X}$ and $\mathbf{Y}$ across the agents. Let $\mathbf{x}_m$ and $\mathbf{y}_n$ denote the mth and the nth column of $\mathbf{X}$ and $\mathbf{Y}$, respectively; the agent owning z_{mn} will control/update the variables $\mathbf{x}_m$ (or $\mathbf{y}_n$), and it is connected to the agent that optimizes the column $\mathbf{y}_n$ (or $\mathbf{x}_m$). By doing so, we minimize the overlapping across the block-variables and, consequently, the communications among the agents. Problem (1) can be then rewritten in the multiagent form (P), setting

$$f_i((\mathbf{X},\mathbf{Y})_{\mathcal{N}_i}) = \frac{1}{2} \sum_{(m,n)\in\Omega_i} (\mathbf{x}_m^T \mathbf{y}_n - z_{mn})^2 \tag{2}$$

and

$$g_i\left(\{\mathbf{x}_m\}_{m\in X_i}, \{\mathbf{y}_n\}_{n\in Y_i}\right) = \frac{\lambda}{2} \sum_{m\in X_i} \|\mathbf{x}_m\|_2^2 + \frac{\xi}{2} \sum_{n\in Y_i} \|\mathbf{y}_n\|_2^2, \tag{3}$$

where $\Omega_i \subseteq \Omega$ contains the indices associated to the components of $(\mathbf{Z})_\Omega$ owned by agent i, and X_i (resp. Y_i) is the set of the column indexes of $\mathbf{X}$ (resp. $\mathbf{Y}$) controlled by agent i.

Example #2 – Empirical Risk Minimization Over Networks: Consider now a network setting where data are distributed across N geographically separated nodes. As concrete example, let us pick the renowned LASSO problem [47]:

$$\min_{\mathbf{x}=(\mathbf{x}_1^T,\ldots,\mathbf{x}_N^T)^T\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{4}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}, \mathbf{b} \in \mathbb{R}^m$, and $\lambda > 0$ is a regularization parameter. Note that (4) easily falls into Problem (P); for each $i \in \mathcal{N}$, it is sufficient to set $f_i(\mathbf{x}) = \|\mathbf{A}_i\mathbf{x} + \mathbf{b}_i\|_2^2$, with $\mathbf{A}_i \in \mathbb{R}^{m\times n}$ and $\mathbf{b}_i \in \mathbb{R}^m$, such that $\mathbf{A} = \sum_{i=1}^N \mathbf{A}_i$ and $\mathbf{b} = \sum_{i=1}^N \mathbf{b}_i$; and $g_i = \|\mathbf{x}_i\|_1$. $\mathbf{A}_i$ and $\mathbf{b}_i$ represent in fact the data stored at agent i's side. Under specific sparsity patterns in the data, the local matrices $\mathbf{A}_i$ may be (or constructed to be), such that each local function f_i depends only on some of the block variables $\mathbf{x}_i$. These dependencies will define the sets $\mathcal{N}_i$ associated to each agent i. Note that $\mathcal{N}_i$ need not coincide with the neighbors of agent i in the communication network (graph). That is, the graph modeling the dependence across the block-variables—the one with node set $\mathcal{N}$ and edge set $\mathcal{E} = \{(i,j) : j \in \mathcal{N}_i$, for some $i \in \mathcal{N}\}$—might not coincide with the communication graph. This can be desirable, e.g., when the communication graph is populated by inefficient communication links, which one wants to avoid using.

III. Distributed Asynchronous Algorithm

In the proposed asynchronous model, agents update their block-variables without any coordination. Let k be the iteration counter: The iteration $k \to k+1$ is triggered when one agent, say i, updates its own block $\mathbf{x}_i$ from $\mathbf{x}_i^k$ to $\mathbf{x}_i^{k+1}$. Hence, $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$ only differ in the ith block $\mathbf{x}_i$. To perform its update, agent i minimizes a strongly convex approximation of $\sum_{j\in\mathcal{N}_i} f_j$—the part of V that depends on $\mathbf{x}_i$—using possibly outdated information collected from the other agents $j \in \mathcal{N}_i$. To represent this situation, let $\mathbf{x}_j^{k-d_j^k(i,i)}, j \in \mathcal{N}_i\backslash\{i\}$, denote the estimate held by agent i of agent j's variable $\mathbf{x}_j^k$, where $d_j^k(i,i)$ is a nonnegative (integer) delay (the reason for the double index (i,i) in d_j^k will become clear shortly). If $d_j^k(i,i) = 0$, agent i owns the most recent information on the variable of agent j, otherwise $\mathbf{x}_j^{k-d_j^k(i,i)}$ is some delayed version of $\mathbf{x}_j^k$. We define as $\mathbf{d}^k(i,i) \triangleq [d_l^k(i,i)]_{l\in\mathcal{N}_i}$ the *delay vector* collecting these delays; for ease of notation $\mathbf{d}^k(i,i)$ contains also the value $d_i^k(i,i)$, set to zero, as each agent has always access to current values of its own variables. Using the above notation and recalling that f_i depends on $\mathbf{x}_{\mathcal{N}_i}$, agent i at iteration k solves the following strongly convex subproblem:

$$\widehat{\mathbf{x}}_i^k \triangleq \arg\min_{\mathbf{x}_i\in\mathcal{X}_i} \left\{ \tilde{f}_i\left(\mathbf{x}_i; \mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}\right) + \sum_{j\in\mathcal{N}_i\backslash\{i\}} \left\langle \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}\right), \mathbf{x}_i - \mathbf{x}_i^k \right\rangle + g_i(\mathbf{x}_i) \right\}, \tag{5}$$

where we defined $\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)} \triangleq [\mathbf{x}_l^{k-d_l^k(i,j)}]_{l\in\mathcal{N}_j}, j \in \mathcal{N}_i$.

The term $\tilde{f}_i$ in (5) is a strongly convex surrogate that replaces the nonconvex function f_i known by agent i; an outdated value of the variables of the other agents is used, $\mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}$, to build this function. Examples of valid surrogates are discussed in Section III-A. The second term in (5) approximates $\sum_{j\in\mathcal{N}_i\backslash\{i\}} f_j$ by replacing each f_j by its first-order approximation at (possibly outdated) $\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}$ (with $\nabla_{\mathbf{x}_i} f_j$ denoting the gradient of f_j with respect to the block $\mathbf{x}_i$), where $\mathbf{d}^k(i,j) \triangleq [d_l^k(i,j)]_{l\in\mathcal{N}_j}$, with $d_l^k(i,j) \geq 0$ representing the delay of the information that i knows about the gradient $\nabla_{\mathbf{x}_i} f_j$. This source of delay on the gradients is due to two facts, namely: 1) Agents $j \in \mathcal{N}_i \setminus \{i\}$ may communicate to i its gradient $\nabla_{\mathbf{x}_i} f_j$ occasionally; and 2) $\nabla_{\mathbf{x}_i} f_j$ is generally computed at some outdated point, as agent j itself may not have access of the last information of the variables of the agents in $\mathcal{N}_j \setminus \{j\}$.

Once $\widehat{\mathbf{x}}_i^k$ has been computed, agent i sets

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \gamma\left(\widehat{\mathbf{x}}_i^k - \mathbf{x}_i^k\right), \tag{6}$$

where $\gamma \in (0;1]$ is suitably chosen stepsize (cf. Section III-A).

The proposed distributed asynchronous algorithm, termed Distributed Asynchronous FLEXible parallel Algorithm (DAsyFLEXA), is formally described in Algorithm 1. We set $\mathbf{x}_i^t = \mathbf{x}_i^0$, for all $t < 0$ and $i \in \mathcal{N}$, without loss of generality.

We stress that agents need know neither the iteration counter k nor the vector of delays. No one "picks agent i^k and the delays $\{\mathbf{d}^k(i^k,j)\}_{j\in\mathcal{N}_{i^k}}$" in (S.1). This is just an *a posteriori* view of the algorithm dynamics: All agents asynchronously and continuously collect information from their neighbors and use it to update $\mathbf{x}_i$; when one agent has completed an update the iteration index k is increased and i^k is defined.

Algorithm 1: Distributed Asynchronous FLEXible parallel Algorithm (DAsyFLEXA).

Initialization: $k{=}0$; $\mathbf{x}^0{\in}\mathcal{X}{\triangleq}\prod_i \mathcal{X}_i$; $\mathbf{x}^t{=}\mathbf{x}^0$, $t{<}0$; $\gamma \in (0;1]$.
while a termination criterion is not met **do**
(S.1): Pick agent i^k and delays $\{\mathbf{d}^k(i^k,j)\}_{j\in\mathcal{N}_{i^k}}$;
(S.2): Compute $\widehat{\mathbf{x}}_{i^k}^k$ according to (5);
(S.3): Update $\mathbf{x}_{i^k}^k$ according to (6);
(S.4): Update the global iteration counter $k \leftarrow k+1$;
end while

A. Assumptions

Before studying convergence of Algorithm 1, we state the main assumptions on Problem (P) and the algorithmic choices.

1) On Problem (P): Below, we will use the following conventions: When a function is said to be differentiable on a certain domain, it is understood that the function is differentiable on an open set containing the domain. We say that f_i is block-LC^1 on a set if it is continuously differentiable on that set and $\nabla_{\mathbf{x}_j} f_i$ are locally Lipschitz. We say V is *coercive* on $\mathcal{X}=\prod_i \mathcal{X}_i$, if $\lim_{\|\mathbf{x}\|\to+\infty,\mathbf{x}\in\mathcal{X}} V(\mathbf{x})=+\infty$; this is equivalent to requiring that all level sets of V in $\mathcal{X}$ are compact.

Assumption A (On Problem (P)):

(A1) Each set $\mathcal{X}_i \subseteq \mathbb{R}^{n_i}$ is nonempty, closed, and convex;
(A2) At least one of the following conditions is satisfied
(a) $\mathcal{L}^0 \triangleq \{\mathbf{x}\in\mathcal{X} : V(\mathbf{x}) \le V(\mathbf{x}^0)\}$ is compact and all f_i are block-LC^1 on $\mathcal{X}_{\mathcal{N}_i} \triangleq \prod_{j\in\mathcal{N}_i} \mathcal{X}_j$;
(b) All f_i are C^1 and their gradients $\nabla_{\mathbf{x}_j} f_i$, $j\in\mathcal{N}_i$, are *globally* Lipschitz on $\mathcal{X}_{\mathcal{N}_i}$;
(A3) Each $g_i : \mathcal{X}_i \to \mathbb{R}$ is convex;
(A4) Problem (P) has a solution;
(A5) The communication graph $\mathcal{G}$ is connected.

The above assumptions are standard and satisfied by many practical problems. For instance, A2(a) holds if V is coercive on $\mathcal{X}$ and all f_i are block-LC^1 on $\mathcal{X}_{\mathcal{N}_i}$. Note that Example #2 satisfies A2(b); A2(a) is motivated by applications such as Example #1, which do not satisfy A2(b). A3 is a common assumption in the literature of parallel and distributed methods for the class of problems (P); two renowned examples are $g_i(\mathbf{x}_i)=\|\mathbf{x}_i\|_1$ and $g_i(\mathbf{x}_i)=\|\mathbf{x}_i\|_2$. Finally, A4 is satisfied if, for example, V is coercive or if $\mathcal{X}$ is bounded.

Remark 1: Extensions to the case of directed graphs or the case where each agent updates multiple block-variables are easy, but not discussed here for the sake of simplicity.

The aim of Algorithm 1 is to find *stationary solutions* of (P), i.e., points $\mathbf{x}^\star \in \mathcal{X}$, such that

$$\langle \nabla F(\mathbf{x}^\star) + \boldsymbol{\xi}, \mathbf{y}-\mathbf{x}^\star\rangle + G(\mathbf{y}) - G(\mathbf{x}^\star) \ge 0, \qquad \forall \mathbf{y}\in\mathcal{X}.$$

Let $\mathcal{X}^\star \subseteq \mathbb{R}^n$ denote the set of such stationary solutions.

2) On an Error Bound Condition: We prove linear convergence of Algorithm 1 under the Luo-Tseng error bound condition, which is stated next. Recall the definition: Given $\alpha>0$,

$$\mathtt{prox}_{\alpha G}(\mathbf{z}) \triangleq \arg\min_{\mathbf{y}\in\mathcal{X}} \left\{ \alpha G(\mathbf{y}) + \frac{1}{2}\|\mathbf{y}-\mathbf{z}\|_2^2 \right\}.$$

Furthermore, given $\mathbf{x}\in\mathbb{R}^n$, let

$$d(\mathbf{x},\mathcal{X}^\star) \triangleq \min_{\mathbf{x}^\star\in\mathcal{X}^\star} \|\mathbf{x}-\mathbf{x}^\star\|_2, \quad P_{\mathcal{X}^\star}(\mathbf{x}) \triangleq \operatorname*{argmin}_{\mathbf{x}^\star\in\mathcal{X}^\star} \|\mathbf{x}-\mathbf{x}^\star\|_2.$$

Note that $P_{\mathcal{X}^\star}(\mathbf{x}) \neq \emptyset$, as $\mathcal{X}^\star$ is closed.

Assumption B (Luo-Tseng error bound):

(B1) For any $\eta > \min_{\mathbf{x}\in\mathcal{X}} V(\mathbf{x})$, there exist $\epsilon,\kappa>0$, such that:

$$\left.\begin{array}{l} V(\mathbf{x}) \le \eta, \\ \|\mathbf{x}-\mathtt{prox}_G(\nabla F(\mathbf{x})-\mathbf{x})\|_2 \le \epsilon \end{array}\right\} \Rightarrow$$

$$d(\mathbf{x},\mathcal{X}^\star) \le \kappa \|\mathbf{x}-\mathtt{prox}_G(\nabla F(\mathbf{x})-\mathbf{x})\|_2,$$

(B2) there exists $\delta>0$ such that

$$\left.\begin{array}{l} \mathbf{x},\mathbf{y}\in\mathcal{X}^*, \\ V(\mathbf{x})\neq V(\mathbf{y}) \end{array}\right\} \Rightarrow \|\mathbf{x}-\mathbf{y}\|_2 \ge \delta.$$

B1 is a local Lipschitzian error bound: The distance of $\mathbf{x}$ from $\mathcal{X}^\star$ is of the same order of the norm of the residual $\mathbf{x}-\mathtt{prox}_G(\nabla F(\mathbf{x})-\mathbf{x})$ at $\mathbf{x}$. It is not difficult to check, that $\mathbf{x}\in\mathcal{X}^\star$ if and only if $\mathbf{x}-\mathtt{prox}_G(\nabla F(\mathbf{x})-\mathbf{x})=0$. Error bounds of this kind have been extensively studied in the literature, see [4], [7] and references therein. Examples of problems satisfying Assumption B include: LASSO, Group LASSO, Logistic Regression, unconstrained optimization with smooth nonconvex quadratic objective or $F(\mathbf{A}\mathbf{x})$, with F being strongly convex and $\mathbf{A}$ being arbitrary. B2 states that the level curves of V restricted to $\mathcal{X}^\star$ are "properly separated". B2 is trivially satisfied, e.g., if V is convex, if $\mathcal{X}$ is bounded, or if (P) has a finite number of stationary solutions.

3) On the Subproblems (5): The surrogate functions $\tilde{f}_i$ satisfy the following fairly standard conditions ($\nabla\tilde{f}_i$ denotes the partial gradient of $\tilde{f}_i$ w.r.t. the first argument).

Assumption C: Each $\tilde{f}_i : \mathcal{X}_i \times \mathcal{X}_{\mathcal{N}_i} \to \mathbb{R}$ is chosen, so that

(C1) $\tilde{f}_i(\cdot;\mathbf{y})$ is C^1 and τ-strongly convex on $\mathcal{X}_i$, for all $\mathbf{y}\in\mathcal{X}_{\mathcal{N}_i}$;
(C2) $\nabla\tilde{f}_i(\mathbf{y}_i;\mathbf{y}_{\mathcal{N}_i}) = \nabla_{\mathbf{y}_i} f_i(\mathbf{y}_{\mathcal{N}_i})$, for all $\mathbf{y}\in\mathcal{X}$;
(C3) $\nabla\tilde{f}_i(\mathbf{y};\cdot)$ is L_i-Lipschitz continuous on $\mathcal{X}_{\mathcal{N}_i}$, for all $\mathbf{y}\in\mathcal{X}_i$.

A wide array of surrogate functions $\tilde{f}_i$ satisfying Assumption C can be found in [48]; three examples are discussed next.

1) It is always possible to choose $\tilde{f}_i$ as the first-order approximation of f_i: $\tilde{f}_i(\mathbf{x}_i;\mathbf{y}_{\mathcal{N}_i}) = \langle\nabla_{\mathbf{x}_i} f(\mathbf{y}_{\mathcal{N}_i}), \mathbf{x}_i-\mathbf{y}_i\rangle + c\|\mathbf{x}_i-\mathbf{y}_i\|_2^2$, where c is a positive constant.

2) If f_i is block-wise uniformly convex, instead of linearizing f_i one can exploit a second-order approximation and set $\tilde{f}_i(\mathbf{x}_i;\mathbf{y}_{\mathcal{N}_i}) = f_i(\mathbf{y}_{\mathcal{N}_i}) + \langle\nabla_{\mathbf{x}_i} f_i(\mathbf{y}_{\mathcal{N}_i}), \mathbf{x}_i-\mathbf{y}_i\rangle + \frac{1}{2}(\mathbf{x}_i-\mathbf{y}_i)^T\nabla^2_{\mathbf{x}_i\mathbf{x}_i} f_i(\mathbf{y}_{\mathcal{N}_i})(\mathbf{x}_i-\mathbf{y}_i) + c\|\mathbf{x}_i-\mathbf{y}_i\|_2^2$, for any $\mathbf{y}\in\mathcal{X}$, where c is a positive constant.

3) In the same setting as above, one can also better preserve the partial convexity of f_i and choose $\tilde{f}_i(\mathbf{x}_i;\mathbf{y}_{\mathcal{N}_i}) = f_i(\mathbf{x}_i,\mathbf{y}_{\mathcal{N}_i\setminus\{i\}}) + c\|\mathbf{x}_i-\mathbf{y}_i\|_2^2$, for any $\mathbf{y}\in\mathcal{X}$.

4) On the Asynchronous/Communication Model: The way agent i builds its own estimates $\mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}$ and $\nabla_{\mathbf{x}_i} f_j(\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)})$, $j\in\mathcal{N}_i\setminus\{i\}$, depends on the particular asynchronous model and communication protocol under consideration and it is immaterial to the convergence of Algorithm 1. This is a major departure from previous works, such as [20], [22], [26], which instead enforce specific asynchrony and communication protocols. We only require the following mild conditions.

Assumption D (On the asynchronous model).

(D1) Every block variable of $\mathbf{x}$ is updated at most every $B\ge N$ iterations, i.e., $\cup_{t=k}^{k+B-1} i^t = \mathcal{N}$, for all k;

(D2) $\exists\, D \in [0, B]$, such that every component of $\mathbf{d}^k(i,j)$, $i \in \mathcal{N}$, $j \in \mathcal{N}_i$, is not greater than D, for any $k \geq 0$.[1]

Assumption D is satisfied virtually in all practical scenarios. D1 controls the frequency of the updates and is satisfied, for example, by any *essentially cyclic rules*. In practice, it is automatically satisfied, e.g., if each agent wakes up and performs an update whenever some internal clock ticks, without any centralized coordination. D2 imposes a mild condition on the communication protocol employed by the agents: Information used in the agents' updates can not become infinitely old. While this implies agents communicate sufficiently often, it does not enforce any specific protocol on the activation/idle time/communication. For instance, differently from several asynchronous schemes in the literature [20]–[23], [26], [27], [34], agents need not be always in "idle mode" to continuously receive messages from their neighbors. Notice that time varying delays satisfying D2 model also packet losses.

B. Convergence Analysis

We are now in the position to state the main convergence results for DAsyFLEXA. For nonconvex instances of (P), an appropriate measure of optimality is needed to evaluate the progress of the algorithm toward stationarity. In order to define such a measure, we first introduce the following quantities: For any $k \geq 0$ and $i \in \mathcal{N}$,

$$\widehat{\widehat{\mathbf{x}}}_i^k \triangleq \arg\min_{\mathbf{x}_i \in \mathcal{X}_i} \left\{ \tilde{f}_i\left(\mathbf{x}_i; \mathbf{x}_{\mathcal{N}_i}^k\right) + \sum_{j \in \mathcal{N}_i \setminus \{i\}} \left\langle \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^k\right), \mathbf{x}_i - \mathbf{x}_i^k \right\rangle + g_i(\mathbf{x}_i) \right\}, \quad (7)$$

where $\widehat{\widehat{\mathbf{x}}}_i^k$ is a "synchronous" instance of $\widehat{\mathbf{x}}_i^k$ [cf. (5)] wherein all $\mathbf{d}^k(i,j) = \mathbf{0}$. Convergence to stationarity is monitored by the following merit function:

$$M_V(\mathbf{x}^k) \triangleq \|\widehat{\widehat{\mathbf{x}}}^k - \mathbf{x}^k\|_2^2, \quad \text{with} \quad \widehat{\widehat{\mathbf{x}}}^k \triangleq \left[\widehat{\widehat{\mathbf{x}}}_i^k\right]_{i \in \mathcal{N}}. \quad (8)$$

Note that M_V is a valid measure of stationarity, as M_V is continuous and $M_V(\mathbf{x}^k) = 0$ if and only if $\mathbf{x}^k \in \mathcal{X}^\star$.

The following theorem shows that, when agents use a sufficiently small stepsize, the sequence of the iterates produced by DAsyFLEXA converges to a stationary solution of (P), driving $M_V(\mathbf{x}^k)$ to zero at a sublinear rate. In the theorem we use two positive constants, L and C_1, whose definition is given in Appendix A and C3 [cf. (28)], respectively. Suffices to say, here, that L is essentially a Lipschitz constant for the partial gradients $\nabla_{\mathbf{x}_i} f_i$ whose definition varies according to whether A2(a) or A2(b) holds. In the latter case, L is simply the largest global Lipschitz constant for all $\nabla_{\mathbf{x}_j} f_i$'s. In the former case, the sequences $\{\mathbf{x}^k\}$ and $\{\widehat{\mathbf{x}}^k\}$, with $\widehat{\mathbf{x}}^k \triangleq [\widehat{\mathbf{x}}_i^k]_{i \in \mathcal{N}}$, are proved to be bounded [cf. Theorem 2(c)]; L is then the Lipschitz constant of all $\nabla_{\mathbf{x}_j} f_i$'s over the compact set confining these sequences.

Theorem 2: Given Problem (P) under Assumption A; let $\{\mathbf{x}^k\}$ be the sequence generated by DAsyFLEXA, under Assumptions C and D. Choose $\gamma \in (0,1]$, such that $\gamma < \frac{2\tau}{L(2+\rho^2 D^2)}$, with $\rho \triangleq \max_{i \in \mathcal{N}} |\mathcal{N}_i|$. Then, there hold the following:

a) Any limit point of $\{\mathbf{x}^k\}$ is a stationary solution of (P).
b) In at most T_ϵ iterations, DAsyFLEXA drives the stationarity measure $M_V(\mathbf{x}^k)$ below ϵ, $\epsilon > 0$, where
$$T_\epsilon = \left\lceil C_1 \left(V(\mathbf{x}^0) - \min_{\mathbf{x} \in \mathcal{X}} V(\mathbf{x}) \right) \cdot \frac{1}{\epsilon} \right\rceil,$$
where $C_1 > 0$ is a constant defined in the Appendix [cf. (28)], which depends on $\rho, L_i, i \in \mathcal{N}, L, \tau, \gamma, N, B$, and D.
c) If, in particular A2(a) is satisfied, $\{\mathbf{x}^k\}$ is bounded.

Proof: See the Appendix C. ■

Theorem 2 provides a unified set of convergence conditions for several algorithms, asynchronous models, and communication protocols. Note that when $D = 0$, the condition on γ, reduces to the renowned condition used in the synchronous proximal-gradient algorithm. The term D^2 in the denominator of the upper-bound on γ should then be seen as the price to pay for asynchrony: The larger the possible delay D, the smaller γ, to make the algorithm robust to asynchrony/delays.

Theorem 3 improves on the convergence of DAsyFLEXA, when V satisfies the error bound condition in Assumption B. Specifically, convergence of the whole sequence $\{\mathbf{x}^k\}$ to a stationary solution $\mathbf{x}^\star$ is established [in contrast with subsequence convergence in Theorem 2 (b)], and suitable subsequences that converge *linearly* are identified.

Theorem 3: Given Problem (P) under Assumptions A and B, let $\{\mathbf{x}^k\}$ be the sequence generated by DAsyFLEXA, under Assumptions C and D. Suppose that $\gamma/\tau > 0$ is sufficiently small. Then, $\{\mathbf{x}^{t+kB}\}$ and $\{V(\mathbf{x}^{t+kB})\}$, $t \in \{0, \ldots, B-1\}$, converge at least R-linearly to some $\mathbf{x}^\star \in \mathcal{X}^\star$ and $V^\star \triangleq V(\mathbf{x}^\star)$, respectively, that is

$$V(\mathbf{x}^{t+kB}) - V^\star = \mathcal{O}\left(\lambda^{t+kB}\right),$$
$$\|\mathbf{x}^{t+kB} - \mathbf{x}^\star\| = \mathcal{O}\left(\sqrt{\lambda^{t+kB}}\right),$$

where $\lambda \in (0,1)$ is a constant defined in the Appendix [cf. (38)], which depends on $\rho, L_i, i \in \mathcal{N}, L, \tau, \gamma, N, B$, and D.

Proof: See the Appendix D. ■

In essence, the theorem proves a B-steps linear convergence rate. To the best of our knowledge, this is the first (linear) convergence rate result in the literature for an asynchronous algorithm in the setting considered in this article.

IV. Numerical Results

In this section we report some numerical results on the two problems described in Section II. All our experiments were run on the Archimedes1 cluster computer at Purdue University, equipped with two 22-cores Intel E5-2699Av4 processors (44 cores in total) and 512GB of RAM. Code for the LASSO problem was written in MATLAB R2019a; code for the Matrix Completion problem was written in C++ using the OpenMPI library for parallel and asynchronous operations.

A. Distributed LASSO

1) Problem Setting: We simulate the (convex) LASSO problem stated in (4). The underlying sparse linear model is generated as follows: $\mathbf{b} = \mathbf{A}\mathbf{x}^\star + \mathbf{e}$, where $\mathbf{A} \in \mathbb{R}^{15000 \times 30000}$. $\mathbf{A}$, $\mathbf{x}^\star$, and $\mathbf{e}$ have i.i.d. elements, drawn from a Gaussian $\mathcal{N}(0, \sigma^2)$

[1] While (S.2) in Algorithm 1 is defined once $\mathbf{d}^k(i^k, j)$, $j \in \mathcal{N}_{i^k}$ is given, here we extend the definition of the delay vectors $\mathbf{d}^k(i,j)$ to all $i, j \in \mathcal{N}$, whose values are set to the delays of the information known by the associated agent on the variables and gradients of the others, at the time agent i^k performs its update. This will simplify the notation in some of the technical derivations.

distribution, with $\sigma = 1$ for $\mathbf{A}$ and $\mathbf{x}^\star$, and $\sigma = 0.1$ for the noise vector $\mathbf{e}$. Entries of $\mathbf{A}$ are then normalized by $\|\mathbf{A}\|$. To impose sparsity on $\mathbf{x}^\star$ and $\mathbf{A}$, we randomly set to zero 95% of their components. Finally, in (4), we set $\lambda = 1$.

2) Network Setting: We consider a fixed, undirected network composed of 50 agents; $\mathbf{x} \in \mathbb{R}^{30000}$ is partitioned in 50 block-variables $\mathbf{x}_i \in \mathbb{R}^{600}$, $i \in \{1, \ldots, 50\}$, each of them controlled by one agent. We define the local functions f_i and g_i as described in Section II (cf. Example #2); each $\mathbf{A}_i$ (resp. $\mathbf{b}_i$) is all zeros but its ith row (resp. component), which coincides with that of $\mathbf{A}$ (resp. $\mathbf{b}$). This induces the following communication pattern among the agents: Each agent i is connected only to the agents js owning the $\mathbf{x}_j$s corresponding to the nonzero column-entries of $\mathbf{A}_i$.

Algorithms: We simulated the following algorithms.

• *DAsyFLEXA*: We used the surrogate functions

$$\tilde{f}_i\left(\mathbf{x}_i; \mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}\right) = \left\langle \nabla_{\mathbf{x}_i} f_i\left(\mathbf{x}_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}\right), \mathbf{x}_i - \mathbf{x}_i^k \right\rangle + \frac{\tau_i}{2}\|\mathbf{x}_i - \mathbf{x}_i^k\|_2^2, \quad (9)$$

where $\tau_i > 0$ is a tunable parameter, which is updated following the same heuristic used in [49]. The stepsize γ is set to 0.9. Note that, using (9), problem (5) has a closed-form solution via the renowned soft-thresholding operator.

• *PrimalDual* asynchronous algorithm [36]: This seems to be the distributed asynchronous scheme closest to DAsyFLEXA. Note that there are some important differences between the two algorithms. First, the PrimalDual algorithm [36] does not exploit the sparsity pattern of the objective function V; every agent instead controls and updates a local copy of the *entire* vector $\mathbf{x}$, which requires employing a consensus mechanism to enforce an agreement on such local copies. This leads to an unnecessary communication overhead among the agents. Second, no explicit estimate of the gradients of the other agents is employed; the lack of this knowledge is overcome by introducing additional communication variables, which lead to contribute to increase the communication cost. Third, the PrimalDual algorithm does not have convergence guarantees in the nonconvex case. In our simulations we tuned the stepsizes of [36] by hand in order to obtain the best performances; specifically we set $\alpha = 0.9$, and $\eta_i = 1.5$ for $i = 1, \ldots, 50$ (see [36] for details on these parameters).

• *AsyBADMM*: This is a block-wise asynchronous ADMM, introduced in [41] to solve nonconvex and nonsmooth optimization problems. Since AsyBADMM requires the presence of master and worker nodes in the network, to implement it on a meshed network, we selected uniformly at random 5 nodes of the network as servers while the others acting as workers. The parameters of the algorithm (see [41] for details) are tuned by hand in order to obtain the best performances; specifically we set $\gamma = 0.06$, $C = 10^4$, and $\rho_{ij} = 50$, for all (i, j).

All the algorithms are initialized from the same randomly chosen point, drawn from $\mathcal{N}(0, 1)$.

Asynchronous Model: We simulate the following asynchronous model. Each agent is activated periodically, every time a local clock triggers. The agents' local clocks have the same frequency but different phase shift, which are selected uniformly at random within $[5, 50]$. Based upon its activation, each agent: 1) performs its update and then broadcasts its gradient vector $\nabla_{\mathbf{x}_i} f_i$ together with its own block-variable $\mathbf{x}_i$ to the agents in $\mathcal{N}_i \backslash \{i\}$; and 2) modifies the phase shift of its local clock by selecting uniformly at random a new value.

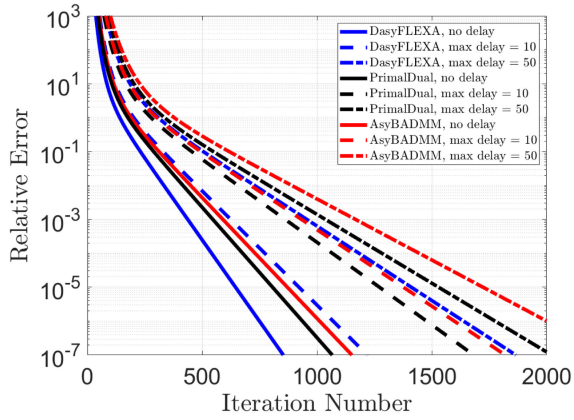

Fig. 1. LASSO problem: Relative error versus # of iterations.

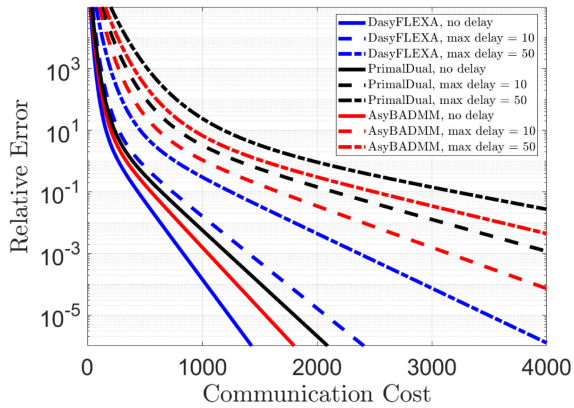

Fig. 2. LASSO problem: Relative error versus # of message exchanges.

Fig. 1 plots relative error $(V(\mathbf{x}^k) - V^\star)/V^\star$ of the different methods versus the number of iterations. Fig. 2 shows the same function versus the number of message exchanges per agent; each scalar variable sent from an agent to one of its neighbors is counted as one message exchanged. All the curves are averaged over ten independent realizations.

DAsyFLEXA outperforms the PrimalDual scheme [36] and AsyBADMM [41]. Also, as anticipated, PrimalDual requires much more communications than DAsyFLEXA.

B. Distributed Matrix Completion

In this section we consider the Distributed Matrix Completion problem (1). We generate a 2200×2200 matrix $\mathbf{Z}$ with samples drawn from $\mathcal{N}(0, 1)$; and we set $\lambda = \xi = 1$ and $r = 4$. Each core of our cluster computer represents a different agent; the columns of $\mathbf{X}$ and $\mathbf{Y}$ are equally partitioned across the 22 cores, and those of $\mathbf{Y}$ uniformly among the other 11 cores; and all cores access a shared memory where the data are stored. We sampled uniformly at random 10% of the entries of $\mathbf{Z}$, and distributed these samples z_{mn} to the agents owing the corresponding column $\mathbf{x}_m$ of $\mathbf{X}$ or $\mathbf{y}_n$ of $\mathbf{Y}$, choosing randomly between the two.

We applied the following instance of DAsyFLEXA to (1). Consider one of the agents that optimizes some columns of $\mathbf{X}$, say agent i. Since each f_i is biconvex in $\mathbf{X}$ and $\mathbf{Y}$, the following

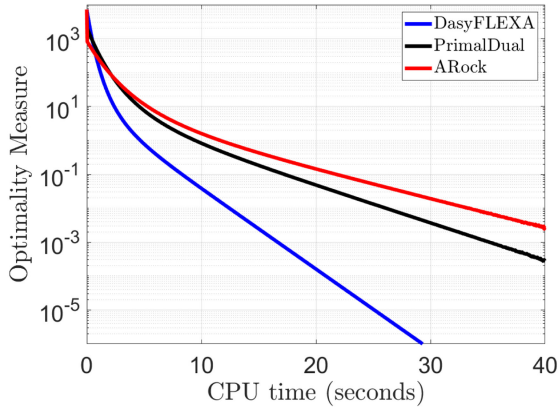

Fig. 3. Matrix completion: Stationarity distance versus # CPU time (in seconds).

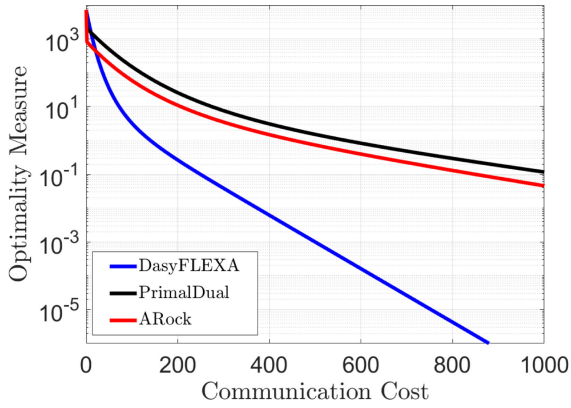

Fig. 4. Matrix completion: Stationarity distance versus # of message exchanges.

surrogate function satisfies Assumption C:

$$\tilde{f}_i\left(\{\mathbf{x}_m\}_{m\in X_i};(\mathbf{X},\mathbf{Y})_{\mathcal{N}_i}^{k-\mathbf{d}^k(i,i)}\right)$$
$$=\frac{1}{2}\sum_{(m,n)\in\Omega_i}\left(\mathbf{x}_m^T\mathbf{y}_n^{k-d^k_{j(n)}(i,i)}-z_{mn}\right)^2+\frac{\tau_i}{2}\sum_{(m,n)\in\Omega_i}\|\mathbf{x}_m-\mathbf{x}_m^k\|_2^2, \tag{10}$$

where $j(n)$ is the index $j\in\mathcal{N}_i$ of the agent that controls $\mathbf{y}_n$, and $\tau_i>0$ is updated following the same heuristic used in [49] [the surrogate function for the agents that update columns of $\mathbf{Y}$ is the same as (10), with the obvious change of variables]. Note that (10) preserves the block-wise convexity present in the original function f_i, which contrasts with the common approach in the literature based on the linearization of f_i. Problem (5) with the surrogate (10) has a closed-form solution.

We compare our algorithm with the decentralized ADMM version of ARock, as presented in [37]. Even if this method has convergence guarantees for convex problems only, its performances on this experiment appeared to be good. For ARock we fixed $\eta^k=0.9$, for all k, and $\gamma=10$, which are the values that gave us the best performances in the experiments.

The rest of the setup is the same as that described for the LASSO problem. Figs. 3 and 4 plot $\|\widehat{\mathbf{x}}^k-\mathbf{x}^k\|_\infty$ (a valid measure of stationarity), with $\widehat{\mathbf{x}}_i$ defined as in (5), obtained by DAsyFLEXA and the PrimalDual algorithm [36] versus the CPU time (measured in seconds) and message exchanges per agent. On our tests, we observed that all the algorithms converged to the same stationary solution. The results confirm the behavior observed in the previous section for convex problems: DAsyFLEXA has better performances than PrimalDual, and the difference is mostly significant is terms of communication cost. DAsyFLEXA is also more efficient than ARock, which suffers from a similar drawback of PrimalDual for what concerns the number of message exchanges; this is due to the fact that ARock requires the use of dual variables, which cause a communication overhead.

APPENDIX

In this Appendix we prove Theorems 2 and 3.

A. Notation

Vectors $\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}$ have different length. It is convenient to replace them with equal-length vectors retaining of course the same information. This is done introducing the following (column) vectors $\mathbf{x}^k(i,j)\triangleq(\mathbf{x}_l^k(i,j))_{l=1}^N\in\mathcal{X}$, defined as:

$$[\mathbf{x}_l^k(i,j)]_{l\in\mathcal{N}_j}\triangleq\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}, \tag{11a}$$
$$\mathbf{x}_l^k(i,j)=\mathbf{x}_l^k,\quad l\notin\mathcal{N}_j. \tag{11b}$$

In other words, the blocks of $\mathbf{x}^k(i,j)$ indexed by $\mathcal{N}_j$ coincide with $\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}$ whereas the other block-components, irrelevant to the proofs, are conveniently set to their most up-to-date values. We will use the shorthand $\mathbf{x}_{\mathcal{N}_j}^k(i,j)\triangleq[\mathbf{x}_l^k(i,j)]_{l\in\mathcal{N}_j}$.

Since at each iteration $k\geq 0$ only one block of $\mathbf{x}^k$ is updated, and because of Assumption D2, it is not difficult to check that the delayed vector $\mathbf{x}^k(i,j)$ can be written as

$$\mathbf{x}^k(i,j)=\mathbf{x}^k+\sum_{l\in\mathcal{K}^k(i,j)}(\mathbf{x}^l-\mathbf{x}^{l+1}), \tag{12}$$

where $\mathcal{K}^k(i,j)$ is a subset of $\{k-D,\dots,k-1\}$ whose elements depend on which block variables have been updated in the window $[\max\{0,k-D\},\max\{0,k-1\}]$. Recall that it is assumed $\mathbf{x}^t=\mathbf{x}^0$, for $t<0$.

Finally, notice that the notation $\widehat{\mathbf{x}}_i^k$ for the best-response map (5) is a shorthand for the formal expression $\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(i))$, where $\tilde{\mathbf{x}}^k(i)\triangleq[\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}]_{j\in\mathcal{N}_i}$. Similarly, $\widehat{\bar{\mathbf{x}}}_i^k$ (resp. $\widehat{\bar{\mathbf{x}}}^k$) in (7) is a shorthand for $\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k(i))$ (resp. $\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k)$), where $\bar{\mathbf{x}}^k(i)\triangleq[\mathbf{x}_{\mathcal{N}_j}^k]_{j\in\mathcal{N}_i}$ (resp. $\bar{\mathbf{x}}^k\triangleq[\bar{\mathbf{x}}^k(i)]_{i\in\mathcal{N}}$). We also define the following shorthands:

$$\Delta\widehat{\mathbf{x}}^k\triangleq\left[\Delta\widehat{\mathbf{x}}_i^k\right]_{i\in\mathcal{N}},\quad \Delta\widehat{\mathbf{x}}_i^k\triangleq\widehat{\mathbf{x}}_i^k-\mathbf{x}_i^k. \tag{13}$$

Table I summarizes the main notation used in the article.

On the constant L: The proofs rely on some Lipschitz properties of $\nabla_{\mathbf{x}_j}f_i$'s. To provide a unified proof under either A2(a) or A2(b), we introduce a constant $L>0$ whose value depends on whether A2(a) or A2(b) hold. Specifically:

1) *A2(a) holds:* The gradients $\nabla_{\mathbf{x}_j}f_i$'s are not globally Lipschitz on the sets $\mathcal{X}_{\mathcal{N}_i}$'s; our approach to study convergence is to ensure that they are Lipschitz continuous on suitably defined sets containing the sequences generated by Algorithm 1. We define these sets as follows. Define first the set $\texttt{Cube}\triangleq\{\mathbf{w}\in\mathcal{X}:\|\mathbf{w}\|_\infty\leq U\}$, where U is positive constant that ensures $\mathcal{L}^0\subseteq\texttt{Cube}$ (note that $U<+\infty$ because $\mathcal{L}^0$ is bounded). Then,

TABLE I
TABLE OF NOTATION

Symbol	Definition
$V(\mathbf{x})$, cf. (P)	$F(\mathbf{x})+G(\mathbf{x})$
$F(\mathbf{x})$, cf. (P)	$\sum_{i=1}^{N} f_i(\mathbf{x}_{\mathcal{N}_i})$
$G(\mathbf{x})$, cf. (P)	$\sum_{i=1}^{N} g_i(\mathbf{x}_i)$
$\mathbf{x}$, cf. (P)	Optimization variable
$\mathbf{x}_i$, cf. (P)	Block-variable of agent i
$\mathbf{x}_{\mathcal{N}_i}$, cf. (P)	Block-variables of agent i's set of neighbors: $[\mathbf{x}_j]_{j\in\mathcal{N}_i}$
$\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}$	Agent i's local copy of agent j's vector $\mathbf{x}_{\mathcal{N}_j}^k$, possibly delayed
$\mathbf{x}^k(i,j)$, cf. (11)	Same as $\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}$, with the addition of slack elements to fix dimensionality
$\widehat{\mathbf{x}}_i^k/\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^k(i))$, cf. (5)	Solution of subproblem (5)
$\tilde{\mathbf{x}}^k(i)$	Agent i's local copies of his neighbors vectors $[\mathbf{x}_{\mathcal{N}_j}^k]_{j\in\mathcal{N}_i}$, possibly delayed: $\left[\mathbf{x}_{\mathcal{N}_j}^{k-\mathbf{d}^k(i,j)}\right]_{j\in\mathcal{N}_i}$
$\bar{\mathbf{x}}^k$	Collection of all the $\tilde{\mathbf{x}}^k(i)$'s: $\left[\bar{\mathbf{x}}^k(i)\right]_{i\in\mathcal{N}}$
$\widehat{\bar{\mathbf{x}}}_i^k/\widehat{\mathbf{x}}_i(\bar{\mathbf{x}}^k(i))$, cf. (7)	Solution of subproblem (5) wherein all the delays are set to 0
$\bar{\mathbf{x}}^k(i)$	Same structure of $\tilde{\mathbf{x}}^k(i)$ wherein all the delays are set to 0
$\bar{\mathbf{x}}^k$	Collection of all the $\bar{\mathbf{x}}^k(i)$'s: $\left[\bar{\mathbf{x}}^k(i)\right]_{i\in\mathcal{N}}$

we define a proper widening $\bar{\mathcal{L}}^0$ of $\mathcal{L}^0$: $\bar{\mathcal{L}}^0 \triangleq (\mathcal{L}^0+\psi\mathcal{B})\cap\mathcal{X}$, where $\mathcal{B}$ is the unitary ball centered in the origin, and $\psi>0$ is a finite positive constant defined as

$$\psi \triangleq \max_{i\in\mathcal{N}} \max_{\substack{\tilde{\mathbf{w}}(i)\triangleq[\mathbf{w}_{\mathcal{N}_j}(j)]_{j\in\mathcal{N}_i}\\ \mathbf{w}(j)\in\mathtt{Cube}}} \|\widehat{\mathbf{x}}_i(\tilde{\mathbf{w}}(i))-\mathbf{w}_i(i)\|_2. \tag{14}$$

Note that $\bar{\mathcal{L}}^0$ is compact, because $\mathcal{L}^0$ is bounded and $\psi<+\infty$ [given that Cube is bounded and $\widehat{\mathbf{x}}(\cdot)$ is continuous, due to (5), A2, A3, and C3]. Consider now any vector $\mathbf{x}\in\bar{\mathcal{L}}^0$. A2(a) and compactness of $\bar{\mathcal{L}}^0$ imply that the gradients $\nabla_{\mathbf{x}_j}f_i$'s are globally Lipschitz over the sets containing the subvectors $\mathbf{x}_{\mathcal{N}_i}$'s, with L being the maximum value of the Lipschitz constant of all the gradients over these sets.

2) *A2(b) holds:* In this case, L is simply the global Lipschitz constant $\nabla_{\mathbf{x}_i}f_i$ over the whole space.

Remark 4: To make sense of the complicated definition of L under A2(a), we anticipate how this constant will be used. Our proof leverages the descent lemma to majorize $V(\mathbf{x}^{k+1})$. To do so, each $\nabla_{\mathbf{x}_j}f_i$ needs to be globally Lipschitz on a convex set containing $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$. This is what the convex set $\bar{\mathcal{L}}^0$ is meant for: $\mathbf{x}^k$ and $\mathbf{x}^{k+1}$ belong to $\bar{\mathcal{L}}^0$ and thus $\nabla_{\mathbf{x}_j}f_i$ is L-Lipschitz continuous.

B. Preliminaries

We summarize next some properties of the map $\widehat{\mathbf{x}}_i^k$ in (5).

Proposition 5: Given Problem (P) under Assumption A, let $\{\mathbf{x}^k\}$ be the sequence generated by DAsyFLEXA under Assumptions B and C. Suppose also that $\mathbf{x}^k\in\bar{\mathcal{L}}^0$ for all k. There hold:

a) [Optimality] For any $i\in\mathcal{N}$ and $k\geq 0$

$$\sum_{j\in\mathcal{N}_i}\left\langle\nabla_{\mathbf{x}_i}f_j(\mathbf{x}_{\mathcal{N}_j}^k(i,j)),\Delta\widehat{\mathbf{x}}_i^k\right\rangle + g_i(\widehat{\mathbf{x}}_i^k)-g_i(\mathbf{x}_i^k)\leq-\tau\|\Delta\widehat{\mathbf{x}}_i^k\|_2^2. \tag{15}$$

b) [Lipschitz continuity] For any $i\in\mathcal{N}$ and $k,h\geq 0$

$$\|\widehat{\mathbf{x}}_i^k-\widehat{\mathbf{x}}_i^h\|_2\leq\frac{L_m}{\tau}\|\mathbf{x}^k(i,i)-\mathbf{x}^h(i,i)\|_2 + \frac{L}{\tau}\sum_{j\in\mathcal{N}_i\setminus\{i\}}\|\mathbf{x}^k(i,j)-\mathbf{x}^k(i,j)\|_2, \tag{16}$$

where $L_m\triangleq\max_{i\in\mathcal{N}}L_i$.

c) [Fixed-points] $\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k)=\mathbf{x}^k$ if and only if $\mathbf{x}^k$ is a stationary solution of Problem (P) (recall the definition of $\bar{\mathbf{x}}^k$ in Table I).

d) [Error bound] For any $k\geq 0$

$$\|\mathbf{x}^k-\mathtt{prox}_G\left(\mathbf{x}^k-\nabla F(\mathbf{x}^k)\right)\|_2 \leq(1+L+NL_m)\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k)-\mathbf{x}^k\|_2. \tag{17}$$

Proof: See the technical report [50]. ■

C. Proof of Theorem 2

The proof is organized in the following steps.

Step 1–Lyapunov function & its descent: We define an appropriate Lyapunov function $\tilde{V}$ and prove that it is monotonically nonincreasing along the iterations. This also proves Theorem 2(c);

Step 2–Vanishing $\mathbf{x}$*-stationarity:* Building on the descent properties of the Lyapunov function, we prove $\lim_{k\to+\infty}\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k)-\mathbf{x}^k\|_2=0$ [Theorem 2(a)];

Step 3–Convergence rate: We prove the sublinear convergence rate of $\{M_V(\mathbf{x}^k)\}$ as stated in Theorem 2(c).

The above steps are proved under Assumptions A, C, and D.

1) Step 1–Lyapunov Function & Its Descent: Introduce the following Lyapunov-like function:

$$\tilde{V}(\mathbf{x}^k,\ldots,\mathbf{x}^{k-D}) \triangleq V(\mathbf{x}^k)+\frac{DL\rho^2}{2}\left(\sum_{l=k-D}^{k-1}(l-(k-1)+D)\|\mathbf{x}^{l+1}-\mathbf{x}^l\|_2^2\right), \tag{18}$$

where L is defined in Appendix A. Note that

$$\tilde{V}^\star\triangleq\min_{[\mathbf{y}^i\in\mathcal{X}]_{i=1}^{D+1}}\tilde{V}(\mathbf{y}^1,\ldots,\mathbf{y}^{D+1})=\min_{\mathbf{x}\in\mathcal{X}}V(\mathbf{x}).$$

The following lemma establishes the descent properties of $\tilde{V}$ and also proves Theorem 2(c).

Lemma 6: Given $\tilde{V}$ defined in (18), the following hold:

a) For any $k\geq 0$:

$$\tilde{V}(\mathbf{x}^{k+1}\ldots,\mathbf{x}^{k+1-D}) \leq\tilde{V}(\mathbf{x}^k,\ldots,\mathbf{x}^{k-D})-\gamma\left(\tau-\gamma\frac{L\left(2+D^2\rho^2\right)}{2}\right)\|\Delta\widehat{\mathbf{x}}_{ik}^k\|_2^2. \tag{19}$$

b) If, in particular, A2(a) is satisfied: $\mathbf{x}^k\in\mathcal{L}^0$, for all $k\geq 0$.

Proof: We prove the two statements by induction. For $k=0$,

$$\begin{aligned}V(\mathbf{x}^1) &= \sum_{i=1}^N f_i(\mathbf{x}^1_{\mathcal{N}_i}) + g_{i^0}(\mathbf{x}^1_{i^0}) + \sum_{i\neq i^0} g_i(\mathbf{x}^1_i)\\ &\overset{(6)}{=} \sum_{i=1}^N f_i(\mathbf{x}^1_{\mathcal{N}_i}) + g_{i^0}(\mathbf{x}^1_{i^0}) + \sum_{i\neq i^0} g_i(\mathbf{x}^0_i)\\ &\overset{(a)}{\leq} \sum_{i=1}^N f_i(\mathbf{x}^0_{\mathcal{N}_i}) + \gamma \sum_{j\in\mathcal{N}_{i^0}} \Big\langle \nabla_{\mathbf{x}_{i^0}} f_j(\mathbf{x}^0_{\mathcal{N}_j})\\ &\quad + \nabla_{\mathbf{x}_{i^0}} f_j\Big(\mathbf{x}^0_{\mathcal{N}_j}(i^0,j)\Big) - \nabla_{\mathbf{x}_{i^0}} f_j\Big(\mathbf{x}^0_{\mathcal{N}_j}(i^0,j)\Big), \Delta\widehat{\mathbf{x}}^0_{i^0}\Big\rangle\\ &\quad + \frac{\gamma^2 L}{2}\|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2^2 + g_{i^0}(\mathbf{x}^1_{i^0}) + \sum_{i\neq i^0} g_i(\mathbf{x}^0_i)\\ &\overset{\text{A3}}{\leq} \sum_{i=1}^N f_i(\mathbf{x}^0_{\mathcal{N}_i}) + \gamma\left\langle \sum_{j\in\mathcal{N}_{i^0}} \nabla_{\mathbf{x}_{i^0}} f_j\left(\mathbf{x}^0_{\mathcal{N}_j}(i^0,j)\right), \Delta\widehat{\mathbf{x}}^0_{i^0}\right\rangle\\ &\quad + \gamma\Big\langle \sum_{j\in\mathcal{N}_{i^0}} \Big(\nabla_{\mathbf{x}_{i^0}} f_j(\mathbf{x}^0_{\mathcal{N}_j})\\ &\quad - \nabla_{\mathbf{x}_{i^0}} f_j\left(\mathbf{x}^0_{\mathcal{N}_j}(i^0,j)\right)\Big), \Delta\widehat{\mathbf{x}}^0_{i^0}\Big\rangle + \frac{\gamma^2 L}{2}\|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2^2\\ &\quad + \sum_{i=1}^N g_i(\mathbf{x}^0_i) + \gamma g_{i^0}(\widehat{\mathbf{x}}^0_{i^0}) - \gamma g_{i^0}(\mathbf{x}^0_{i^0})\\ &\overset{(15),A2}{\leq} V(\mathbf{x}^0) - \gamma\left(\tau - \frac{\gamma L}{2}\right)\|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2^2\\ &\quad + \gamma L\|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2 \sum_{j\in\mathcal{N}_{i^0}} \|\mathbf{x}^0 - \mathbf{x}^0(i^0,j)\|_2\\ &\overset{(b)}{\leq} V(\mathbf{x}^0) - \gamma(\tau-\gamma L)\|\Delta\widehat{\mathbf{x}}^0_{i^0}\|_2^2\\ &\quad + \frac{L\rho}{2}\sum_{j\in\mathcal{N}_{i^0}} \underbrace{\|\mathbf{x}^0 - \mathbf{x}^0(i^0,j)\|_2^2}_{\texttt{term I}} \end{aligned} \tag{20}$$

where (a) follows from the descent lemma and the definition of L; and in (b) we used Young's inequality. Note that in (a) we used the fact that $\mathbf{x}^0$ and $\mathbf{x}^1$ belong to $\bar{\mathcal{L}}^0$ (cf. Remark 4).

We now bound `term I` in (20). It is convenient to study the more general term $\|\mathbf{x}^k - \mathbf{x}^k(i^k,j)\|_2^2$, $j\in\mathcal{N}_{i^k}$. There holds:

$$\begin{aligned}\|\mathbf{x}^k - \mathbf{x}^k(i^k,j)\|_2^2 &\overset{(12)}{\leq} \left(\sum_{l=k-D}^{k-1} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2\right)^2\\ &\leq D\sum_{l=k-D}^{k-1} \|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\\ &= D\Bigg(\sum_{l=k-D}^{k-1} (l-(k-1)+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\\ &\quad - \sum_{l=k+1-D}^{k} (l-k+D)\|\mathbf{x}^{l+1} - \mathbf{x}^l\|_2^2\Bigg) + D^2\|\mathbf{x}^{k+1} - \mathbf{x}^k\|_2^2. \end{aligned} \tag{21}$$

Combining (20) and (21) one can check that statements (a) and (b) of the lemma hold at $k=0$, that is, $\tilde{V}(\mathbf{x}^1,\ldots,\mathbf{x}^0) \leq \tilde{V}(\mathbf{x}^0,\ldots,\mathbf{x}^0)$, and $V(\mathbf{x}^1) \leq \tilde{V}(\mathbf{x}^1,\ldots,\mathbf{x}^0) \leq \tilde{V}(\mathbf{x}^0,\ldots,\mathbf{x}^0) = V(\mathbf{x}^0)$, respectively. Assume now that the two statements hold at iteration k. It is easy to check that the analogous of (20) also holds at iteration $k+1$ with the term $\sum_{j\in\mathcal{N}_{i^k}} \|\mathbf{x}^k - \mathbf{x}^k(i^k,j)\|_2$ in the analogous of `term I` at iteration k, majorized using (21). Combining (20) at $k+1$ with (21) one can check that statement (a) of the lemma holds at $k+1$. We also get: $V(\mathbf{x}^{k+1}) \leq \tilde{V}(\mathbf{x}^{k+1},\ldots,\mathbf{x}^{k+1-D}) \overset{(20)}{\leq} \tilde{V}(\mathbf{x}^k,\ldots,\mathbf{x}^{k-D}) \leq \tilde{V}(\mathbf{x}^0,\ldots,\mathbf{x}^0) = V(\mathbf{x}^0)$, which proves statement (b) of the lemma at $k+1$. This completes the proof. ■

2) Step 2 – Vanishing $\mathbf{x}$-Stationarity: It follows from A4 and Lemma 6 that, if $\gamma < \frac{2\tau}{L(2+\rho^2D^2)}$, $\{\tilde{V}(\mathbf{x}^{k-D},\ldots,\mathbf{x}^k)\}$ and thus $\{V(\mathbf{x}^k)\}$ converge. Therefore

$$\lim_{k\to+\infty} \|\Delta\widehat{\mathbf{x}}^k_{i^k}\|_2 = 0. \tag{22}$$

The next lemma extends the vanishing properties of a single block $\Delta\widehat{\mathbf{x}}^k_{i^k}$ to the entire vector $\Delta\widehat{\mathbf{x}}^k$.

Lemma 7: For any $i\in\mathcal{N}$, $k\geq 0$, and $h,t\in[k,k+B-1]$, there hold:

$$\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^t(i)) - \widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^h(i))\|_2^2 \leq C_2 \sum_{l=k-D}^{k+B-2} \|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{23}$$

$$\|\Delta\widehat{\mathbf{x}}^h\|_2^2 \leq 2(NC_2+1)\sum_{l=k-D}^{k+B-1} \|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{24}$$

with

$$C_2 \triangleq \frac{3\gamma^2(B+2D-N+1)\rho\left(L_m^2 + (\rho-1)L^2\right)}{\tau^2}.$$

Proof: See Appendix E. ■

Using (24) and (22) yields

$$\lim_{k\to+\infty} \|\Delta\widehat{\mathbf{x}}^k\|_2 = 0. \tag{25}$$

Furthermore, invoking (22), (23), and (25) together with $\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2 \leq \|\Delta\widehat{\mathbf{x}}^k\|_2 + \|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}^k\|_2$, leads to

$$\lim_{k\to+\infty} \|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2 = 0, \tag{26}$$

which, together with Proposition 5(c), proves Theorem 2(a).

3) Step 3 – Convergence Rate: We use the Lyapunov function $\tilde{V}$ to study the vanishing rate of $\{M_V(\mathbf{x}^k)\}$. Due to (26) and the definition of M_V, we know that M_V is converging to 0. Therefore T_ϵ is finite. Using $M_V(\mathbf{x}^k) > \epsilon$, for all $k\in\{0,\ldots,T_\epsilon-1\}$, we have

$$\begin{aligned}T_\epsilon\epsilon &\leq \sum_{k=0}^{T_\epsilon-1} M_V(\mathbf{x}^k) \leq 2\sum_{k=0}^{T_\epsilon-1}\left(\|\Delta\widehat{\mathbf{x}}^k\|_2^2 + \|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}^k\|_2^2\right)\\ &\overset{(16),(24)}{\leq} 2\sum_{k=0}^{T_\epsilon-1}\Bigg(2(NC_2+1)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2\\ &\quad + \sum_{i=1}^N\left(\frac{L_m^2\rho}{\tau^2}\|\mathbf{x}^k(i,i) - \mathbf{x}^k\|_2^2\right.\\ &\quad \left. + \frac{L^2\rho}{\tau^2}\sum_{j\in\mathcal{N}_i\setminus\{i\}}\|\mathbf{x}^k(i,j) - \mathbf{x}^k\|_2^2\right)\Bigg)\end{aligned}$$

$$\overset{(12)}{\leq} 2\left(2\left(NC_2+1\right)+\frac{DC_2}{3(B+2D-N+1)}\right)\cdot\sum_{k=0}^{T_\epsilon-1}\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2$$

$$\overset{(a)}{\leq} C_3\sum_{k=0}^{T_\epsilon-1}\sum_{l=k-D}^{k+B-1}\left(\tilde{V}\left(\mathbf{x}^l,\ldots,\mathbf{x}^{l-D}\right)-\tilde{V}\left(\mathbf{x}^{l+1},\ldots,\mathbf{x}^{l+1-D}\right)\right)$$

$$=C_3\sum_{k=0}^{T_\epsilon-1}\left(\tilde{V}\left(\mathbf{x}^{k-D},\ldots,\mathbf{x}^{k-2D}\right)-\tilde{V}\left(\mathbf{x}^{k+B},\ldots,\mathbf{x}^{k+B-D}\right)\right)$$

$$\leq C_3(B+D-1)\left(V(\mathbf{x}^0)-\min_{\mathbf{x}\in\mathcal{X}}V(\mathbf{x})\right), \tag{27}$$

where in (a) we used (19) and defined C_3 as

$$C_3\triangleq\frac{4\left(2\left(NC_2+1\right)+\frac{DC_2}{3(B+2D-N+1)}\right)}{\gamma\left(2\tau-\gamma L\left(2+D^2\rho^2\right)\right)}.$$

Statement (b) of the theorem follows readily by defining

$$C_1\triangleq C_3(B+D-1). \tag{28}$$

D. Proof of Theorem 3

We study now convergence of Algorithm 1 under the additional Assumption B.

First of all, note that one can always find $\eta,\epsilon,\kappa>0$, such that B1 holds. In fact, 1) by Lemma 6, there exist some η and sufficiently small γ/τ, such that $V(\mathbf{x}^k)\leq\eta$, for all $k\geq 0$; and 2) since $\|\mathbf{x}^k-\mathtt{prox}_G(\nabla F(\mathbf{x}^k)-\mathbf{x}^k)\|_2$ is asymptotically vanishing [Proposition 5(d) and (26)], one can always find some $\epsilon>0$, such that $\|\mathbf{x}^k-\mathtt{prox}_G(\nabla F(\mathbf{x}^k)-\mathbf{x}^k)\|_2\leq\epsilon$, for all $k\geq 0$.

The proof proceeds along the following steps. Step 1: We first show that the liminf of $\{V(\mathbf{x}^k)\}$ is a stationary point $V^\star$ [see (33)]. Step 2 shows that $\{V(\mathbf{x}^k)\}$ approaches $V^\star$ linearly, up to an error of the order $\mathcal{O}(\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2)$ [see (37)]. Finally, in Step 3 we show that the term $\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2$ is overall vanishing at a geometric rate, implying the convergence of $\{V(\mathbf{x}^k)\}$ to $V^\star$ at a geometric rate.

1) Step 1: Pick any vector $\mathbf{x}^\star(\mathbf{x}^k)\in P_{\mathcal{X}^\star}(\mathbf{x}^k)$, where $P_{\mathcal{X}^\star}(\mathbf{x})\triangleq\arg\min_{\mathbf{x}^\star\in\mathcal{X}^\star}\|\mathbf{x}-\mathbf{x}^\star\|_2$, $\mathbf{x}\in\mathbb{R}^n$. Note that

$$d(\mathbf{x}^k,\mathcal{X}^\star)=\|\mathbf{x}^\star(\mathbf{x}^k)-\mathbf{x}^k\|_2\overset{B1}{\leq}\kappa\|\mathbf{x}^k-\mathtt{prox}_G\big(\nabla_{\mathbf{x}}F(\mathbf{x}^k)-\mathbf{x}^k\big)\|_2. \tag{29}$$

Using (29), (26), and (17), yields

$$\lim_{k\to+\infty}\|\mathbf{x}^\star(\mathbf{x}^k)-\mathbf{x}^\star(\mathbf{x}^{k+1})\|=0. \tag{30}$$

This, together with B2, imply that there exists an index $\bar{k}\geq 0$ and a scalar $V^\star$, such that

$$V(\mathbf{x}^\star(\mathbf{x}^k))=V^\star,\quad\forall\,k\geq\bar{k}. \tag{31}$$

By the Mean Value Theorem, there exists a vector $\boldsymbol{\xi}^k=\beta^k\mathbf{x}^\star(\mathbf{x}^k)+(1-\beta^k)\mathbf{x}^k$, for some $\beta^k\in(0,1)$, such that, for any $k\geq\bar{k}$

$$\begin{aligned}V^\star-V(\mathbf{x}^k)&=\left\langle\nabla_{\mathbf{x}}F(\boldsymbol{\xi}^k),\mathbf{x}^\star(\mathbf{x}^k)-\mathbf{x}^k\right\rangle+G(\mathbf{x}^\star(\mathbf{x}^k))\\&\quad-G(\mathbf{x}^k)\leq\left\langle\nabla_{\mathbf{x}}F(\boldsymbol{\xi}^k)-\nabla_{\mathbf{x}}F(\mathbf{x}^\star(\mathbf{x}^k)),\mathbf{x}^\star(\mathbf{x}^k)-\mathbf{x}^k\right\rangle\\&\overset{(a)}{\leq}\frac{N(\rho^2L^2+1)}{2}\|\mathbf{x}^\star(\mathbf{x}^k)-\mathbf{x}^k\|_2^2\\&\overset{(17),(29)}{\leq}\frac{N\kappa(\rho^2L^2+1)(1+L+NL_m)}{2}\|\widehat{\mathbf{x}}(\bar{\mathbf{x}})^k-\mathbf{x}^k\|_2\end{aligned} \tag{32}$$

where (a) follows from A2 and $\|\boldsymbol{\xi}^k-\mathbf{x}^\star(\mathbf{x}^k)\|_2^2=\|\beta^k\mathbf{x}^\star(\mathbf{x}^k)+(1-\beta^k)\mathbf{x}^k-\mathbf{x}^\star(\mathbf{x}^k)\|_2^2\leq\|\mathbf{x}^\star(\mathbf{x}^k)-\mathbf{x}^k\|_2^2$.

By invoking (32), together with (26), we obtain

$$\liminf_{k\to+\infty}V(\mathbf{x}^k)\geq V^\star. \tag{33}$$

2) Step 2: We next show that $V(\mathbf{x}^k)$ approaches $V^\star$ at a linear rate.

To this end, consider (20) with 0 and 1 replaced by k and $k+1$, respectively; we have the following:

$$\begin{aligned}V(\mathbf{x}^{k+1})&\leq V(\mathbf{x}^k)-\gamma\left(\tau-\gamma L\right)\|\Delta\widehat{\mathbf{x}}_{i^k}\|_2^2\\&\quad+\frac{L\rho}{2}\sum_{j\in\mathcal{N}_{i^k}}\|\mathbf{x}^k-\mathbf{x}^k(i^k,j)\|_2^2\overset{(12)}{\leq}V(\mathbf{x}^k)\\&\quad-\gamma\left(\tau-\gamma L\right)\|\Delta\widehat{\mathbf{x}}_{i^k}\|_2^2+\frac{\gamma^2DL\rho^2}{2}\sum_{l=k-D}^{k-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2.\end{aligned} \tag{34}$$

It is easy to see that, for any $k\geq\bar{k}$, (34) implies

$$\begin{aligned}V(\mathbf{x}^{k+B})-V^\star&\leq V(\mathbf{x}^k)-V^\star\\&\quad-\gamma\left(\tau-\frac{\gamma L(2+BD\rho^2)}{2}\right)\sum_{l=k}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2\\&\quad+\frac{B\gamma^2DL\rho^2}{2}\sum_{l=k-D}^{k-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2.\end{aligned} \tag{35}$$

To prove the desired result we will combine next (35) with the following lemma.

Lemma 8: For any $k\geq 0$, there holds

$$\begin{aligned}V(\mathbf{x}^{k+B})-V(\mathbf{x}^\star(\mathbf{x}^k))&\leq(1-\gamma)\left(V(\mathbf{x}^k)-V(\mathbf{x}^\star(\mathbf{x}^k))\right)\\&\quad+\gamma\left(N\alpha_1+(B-N)\alpha_2\right)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2,\end{aligned} \tag{36}$$

where α_1 and α_2 are two positive constants defined in Appendix E [see (60) and (62), respectively]. ■

Proof: See Appendix E.

Multiplying the two sides of (35) and (36) by $(N\alpha_1+(B-N)\alpha_2)$ and $\tau-\gamma L(2+BD\rho^2)/2$, respectively, and adding the

two inequalities together, yields

$$V(\mathbf{x}^{k+B}) - V^\star \leq \theta \left(V(\mathbf{x}^k) - V^\star\right) + \zeta \sum_{l=k-D}^{k-1} \|\Delta \widehat{\mathbf{x}}_{i^l}^l\|_2^2, \tag{37}$$

for all $k \geq \bar{k}$, where

$$\theta \triangleq \frac{(1-\gamma)(2\tau - \gamma L(BD\mathcal{N}_m^2+2)) + 2N\alpha_1 + 2(B-N)\alpha_2}{2\tau - \gamma L(BD\mathcal{N}_m^2+2) + 2N\alpha_1 + 2(B-N)\alpha_2},$$

and

$$\zeta \triangleq \frac{(N\alpha_1 + (B-N)\alpha_2)(2\tau + \gamma L(BD\rho^2(\gamma-1)+2))}{2\tau - \gamma L(BD\mathcal{N}_m^2+2) + 2N\alpha_1 + 2(B-N)\alpha_2}.$$

3) Step 3: We can now apply Lemma 4.5 in [6] by noticing that (35), (33), and (37) correspond, respectively, to (4.21), (4.22), and to the first inequality after (4.23) in [6]. Theorem 3 readily follows, setting

$$\lambda \triangleq 1 - \frac{\gamma^2}{2} \frac{2\tau - \gamma L(BD\rho^2+2)}{2N\alpha_1 + 2(B-N)\alpha_2} \cdot \frac{}{+\gamma(2-\gamma)(2\tau - \gamma L(BD\rho^2+2))}. \tag{38}$$

E. Miscellanea Results

This section contains the proofs of Lemma 7 and Lemma 8.

Proof of Lemma 7: i) Assume without loss of generality that $t \leq h$. We have

$$\begin{aligned}
\|\widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^t(i)) - \widehat{\mathbf{x}}_i(\tilde{\mathbf{x}}^h(i))\|_2^2 &\overset{(16)}{\leq} \frac{\rho L_m^2}{\tau^2}\|\mathbf{x}^t(i,i) - \mathbf{x}^h(i,i)\|_2^2 \\
&+ \frac{\rho L^2}{\tau^2} \sum_{j\in\mathcal{N}_i\setminus\{i\}} \|\mathbf{x}^t(i,j) - \mathbf{x}^h(i,j)\|_2^2 \\
&\overset{(12),(6)}{\leq} \left(\frac{3\rho\left(L_m^2 + (\rho-1)L^2\right)}{\tau^2}\right)\left(\gamma^2(B-N+1)\right. \\
&\left.\sum_{l=t}^{h-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2 + D\gamma^2\left(\sum_{l=t-D}^{t-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2 + \sum_{l=h-D}^{h-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2\right)\right).
\end{aligned}$$

ii) Define $r_i^{h,k} \triangleq \underset{t\in[k;k+B-1]:i^t=i}{\arg\min} |t-h|$. We have:

$$\begin{aligned}
\|\Delta\widehat{\mathbf{x}}^h\|_2^2 &\leq 2\sum_{i=1}^N \left(\|\widehat{\mathbf{x}}_i^h - \widehat{\mathbf{x}}_i^{r_i^{h,k}}\|_2^2 + \|\Delta\widehat{\mathbf{x}}_i^{r_i^{h,k}}\|_2^2\right) \\
&\overset{(23)}{\leq} 2\sum_{i=1}^N \left(C_2 \sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2 + \|\Delta\widehat{\mathbf{x}}_i^{r_i^{h,k}}\|_2^2\right).
\end{aligned} \tag{39}$$

Proof of Lemma 8: Define $T_i^k + 1$ as the number of times agent i performs its update within $[k, k+B-1]$; let $l_{i,0}^k, \ldots, l_{i,T_i^k}^k$,be the iteration indexes of such updates. By the mean value theorem, there exists a vector $\boldsymbol{\xi}^k = \beta^k \mathbf{x}^\star(\mathbf{x}^k) + (1-\beta^k)\mathbf{x}^k$, for some $\beta^k \in (0,1)$, such that

$$\begin{aligned}
V(\mathbf{x}^{k+B}) - V(\mathbf{x}^\star(\mathbf{x}^k)) &= \left\langle \nabla_{\mathbf{x}} F(\boldsymbol{\xi}^k), \mathbf{x}^{k+B} - \mathbf{x}^\star(\mathbf{x}^k)\right\rangle \\
&+ G(\mathbf{x}^{k+B}) - G(\mathbf{x}^\star(\mathbf{x}^k)) \\
&= \sum_{i=1}^N \Bigg(\underbrace{\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k)\right\rangle}_{\texttt{term II}} \\
&+ \sum_{t=1}^{T_i^k-1} \underbrace{\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{l_{i,t+1}^k} - \mathbf{x}_i^{l_{i,t}^k}\right\rangle}_{\texttt{term III}} \\
&+ \underbrace{\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{k+B} - \mathbf{x}_i^{l_{i,T_i^k}^k}\right\rangle}_{\texttt{term IV}} \Bigg) \\
&+ G(\mathbf{x}^{k+B}) - G(\mathbf{x}^\star(\mathbf{x}^k)).
\end{aligned} \tag{40}$$

To prove (36), it is then sufficient show that `term II`, `term III`, and `term IV` in (40) converge at a geometric rate up to an error of the order $\mathcal{O}(\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2)$. To do this, we first show that `term II`, `term III`, and `term IV` converges at a geometric rate up to the error terms $a_{i,4}^k$, $b_{i,t,4}^k$, and $c_{i,4}^k$, respectively [see (41), (44), and (47)]. Then, we prove that each of these errors is of the order $\mathcal{O}(\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{i^l}^l\|_2^2)$, as desired [see (59) and (61)].

`Term II` can be upper bounded as

$$\begin{aligned}
&\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k)\right\rangle \\
&\overset{A2}{\leq} \left\langle \nabla_{\mathbf{x}_i} F\left(\widehat{\mathbf{x}}^{l_{i,0}^k}\right), \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k)\right\rangle \\
&\quad + \underbrace{\rho L \left\|\widehat{\mathbf{x}}^{l_{i,0}^k} - \boldsymbol{\xi}^k\right\|_2 \left\|\mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k)\right\|_2}_{\triangleq a_{i,1}^k} \\
&\overset{A2,C2,C3}{\leq} \left\langle \nabla \tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right)\right. \\
&\quad \left. + \sum_{j\in\mathcal{N}_i\setminus\{i\}} \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j)\right), \mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k)\right\rangle \\
&\quad + \left\|\mathbf{x}_i^{l_{i,1}^k} - \mathbf{x}_i^\star(\mathbf{x}^k)\right\|_2 \left(L_i \left\|\widehat{\mathbf{x}}_{\mathcal{N}_i}^{l_{i,0}^k} - \mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right\|_2\right. \\
&\quad \left. + L \sum_{j\in\mathcal{N}_i\setminus\{i\}} \left\|\widehat{\mathbf{x}}_{\mathcal{N}_j}^{l_{i,0}^k} - \mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j)\right\|_2\right) + a_{i,1}^k \\
&\overset{(a)}{\leq} (\gamma-1)\left\langle \nabla \tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}; \mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right)\right. \\
&\quad \left. + \sum_{j\in\mathcal{N}_i\setminus\{i\}} \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j)\right), \Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\rangle + g_i\left(\mathbf{x}_i^\star(\mathbf{x}^k)\right) \\
&\quad - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right) + a_{i,2}^k \overset{C2}{\leq} g_i(\mathbf{x}_i^\star(\mathbf{x}^k)) - g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right)
\end{aligned}$$

$$
\begin{aligned}
&+(\gamma-1)\left\langle \sum_{j\in\mathcal{N}_i}\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j)\right),\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\rangle\\
&+(1-\gamma)\left\|\nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right)-\nabla\tilde{f}_i\left(\mathbf{x}_i^{l_{i,0}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right)\right\|_2\\
&\cdot\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2+a_{i,2}^k \overset{(b)}{\leq} g_i\left(\mathbf{x}_i^\star(\mathbf{x}^k)\right)-g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right)\\
&+\frac{1-\gamma}{\gamma}\left(V\left(\mathbf{x}^{l_{i,0}^k}\right)-V\left(\mathbf{x}^{l_{i,0}^k+1}\right)\right)\\
&+(1-\gamma)\left\|\sum_{j\in\mathcal{N}_i}\left(\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}\right)-\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j)\right)\right)\right\|_2\\
&\cdot\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2+\frac{L\gamma(1-\gamma)}{2}\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2^2+a_{i,3}^k\\
&+(1-\gamma)\left(g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right)-g_i\left(\mathbf{x}_i^{l_{i,0}^k}\right)\right)\\
&\overset{(c)}{=}\frac{1-\gamma}{\gamma}\left(V\left(\mathbf{x}^{l_{i,0}^k}\right)-V\left(\mathbf{x}^{l_{i,0}^k+1}\right)\right)+g_i\left(\mathbf{x}_i^\star(\mathbf{x}^k)\right)\\
&\quad+(\gamma-1)g_i\left(\mathbf{x}_i^{l_{i,0}^k}\right)-\gamma g_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right)+a_{i,4}^k, \qquad (41)
\end{aligned}
$$

where the quantities $a_{i,2}^k$ in (a), and $a_{i,3}^k$ in (b) are defined in (42) and (43) shown at the bottom of this page, respectively; furthermore in (b) we used the descent lemma, and in (c) we defined

$$
\begin{aligned}
a_{i,4}^k \triangleq\ & a_{i,3}^k+\frac{L\gamma(1-\gamma)}{2}\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2^2+(1-\gamma)\\
&\cdot\underbrace{\left\|\sum_{j\in\mathcal{N}_i}\left(\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}\right)-\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j)\right)\right)\right\|_2\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2}_{\texttt{term VII}}.
\end{aligned}
$$

`Term III` can be upper bounded as: for any i and $t\in[1,T_i^k-1]$

$$
\begin{aligned}
&\left\langle\nabla_{\mathbf{x}_i}F(\boldsymbol{\xi}^k),\mathbf{x}_i^{l_{i,t+1}^k}-\mathbf{x}_i^{l_{i,t}^k}\right\rangle\\
&\overset{A2}{\leq}\left\langle\nabla_{\mathbf{x}_i}F\left(\widehat{\mathbf{x}}^{l_{i,t}^k}\right),\mathbf{x}_i^{l_{i,t+1}^k}-\mathbf{x}_i^{l_{i,t}^k}\right\rangle\\
&\quad+\underbrace{\rho L\left\|\widehat{\mathbf{x}}^{l_{i,t}^k}-\boldsymbol{\xi}^k\right\|_2\left\|\mathbf{x}_i^{l_{i,t}^k}-\mathbf{x}_i^{l_{i,t+1}^k}\right\|_2}_{\triangleq b_{i,t,1}^k}
\end{aligned}
$$

$$
\begin{aligned}
&\overset{A2,C2,C3}{\leq}\left\langle\nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i)\right)\right.\\
&\quad\left.+\sum_{j\in\mathcal{N}_i\setminus\{i\}}\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j)\right),\mathbf{x}_i^{l_{i,t+1}^k}-\mathbf{x}_i^{l_{i,t}^k}\right\rangle\\
&\quad+\left\|\mathbf{x}_i^{l_{i,t}^k}-\mathbf{x}_i^{l_{i,t+1}^k}\right\|_2\left(L_i\left\|\widehat{\mathbf{x}}_{\mathcal{N}_i}^{l_{i,t}^k}-\mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i)\right\|_2\right.\\
&\quad\left.+L\sum_{j\in\mathcal{N}_i\setminus\{i\}}\left\|\widehat{\mathbf{x}}_{\mathcal{N}_j}^{l_{i,t}^k}-\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j)\right\|_2\right)+b_{i,t,1}^k\\
&\overset{(a)}{\leq}(\gamma-1)\left\langle\nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i)\right)\right.\\
&\quad\left.+\sum_{j\in\mathcal{N}_i\setminus\{i\}}\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j)\right),\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\rangle+g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right)\\
&\quad-g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right)+b_{i,t,2}^k \overset{C2}{\leq} g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right)\\
&+(\gamma-1)\left\langle\sum_{j\in\mathcal{N}_i}\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j)\right),\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\rangle\\
&+(1-\gamma)\left\|\nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i)\right)-\nabla\tilde{f}_i\left(\mathbf{x}_i^{l_{i,t}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,t}^k}(i,i)\right)\right\|_2\\
&\cdot\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\|_2+b_{i,t,2}^k \overset{(b)}{\leq} g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right)\\
&+\frac{1-\gamma}{\gamma}\left(V\left(\mathbf{x}^{l_{i,t}^k}\right)-V\left(\mathbf{x}^{l_{i,t}^k+1}\right)\right)\\
&+(1-\gamma)\left\|\sum_{j\in\mathcal{N}_i}\left(\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}\right)-\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}_{\mathcal{N}_j}^{l_{i,t}^k}(i,j)\right)\right)\right\|_2\\
&\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\|_2+\frac{L\gamma(1-\gamma)}{2}\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right\|_2^2+b_{i,t,3}^k\\
&+(1-\gamma)\left(g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right)-g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right)\right)\\
&=\frac{1-\gamma}{\gamma}\left(V\left(\mathbf{x}^{l_{i,t}^k}\right)-V\left(\mathbf{x}^{l_{i,t}^k+1}\right)\right)\\
&\quad+\gamma\left(g_i\left(\mathbf{x}_i^{l_{i,t}^k}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l_{i,t}^k}\right)\right)+b_{i,t,4}^k \qquad (44)
\end{aligned}
$$

$$
a_{i,2}^k \triangleq a_{i,1}^k+\underbrace{\left\|\mathbf{x}_i^{l_{i,1}^k}-\mathbf{x}_i^\star(\mathbf{x}^k)\right\|_2\left(L_i\left\|\widehat{\mathbf{x}}_{\mathcal{N}_i}^{l_{i,0}^k}-\mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right\|_2+L\sum_{j\in\mathcal{N}_i\setminus\{i\}}\left\|\widehat{\mathbf{x}}_{\mathcal{N}_j}^{l_{i,0}^k}-\mathbf{x}_{\mathcal{N}_j}^{l_{i,0}^k}(i,j)\right\|_2\right)}_{\texttt{term V}} \qquad (42)
$$

$$
a_{i,3}^k \triangleq a_{i,2}^k+(1-\gamma)\underbrace{\left\|\nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l_{i,0}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right)-\nabla\tilde{f}_i\left(\mathbf{x}_i^{l_{i,0}^k};\mathbf{x}_{\mathcal{N}_i}^{l_{i,0}^k}(i,i)\right)\right\|_2\left\|\Delta\widehat{\mathbf{x}}_i^{l_{i,0}^k}\right\|_2}_{\texttt{term VI}} \qquad (43)
$$

$$b^k_{i,t,2} \triangleq b^k_{i,t,1} + \underbrace{\left\|\mathbf{x}_i^{l^k_{i,t}} - \mathbf{x}_i^{l^k_{i,t+1}}\right\|_2 \left(L_i \left\|\widehat{\mathbf{x}}^{l^k_{i,t}}_{\mathcal{N}_i} - \mathbf{x}^{l^k_{i,t}}_{\mathcal{N}_i}(i,i)\right\|_2 + L \sum_{j\in\mathcal{N}_i\setminus\{i\}} \left\|\widehat{\mathbf{x}}^{l^k_{i,t}}_{\mathcal{N}_j} - \mathbf{x}^{l^k_{i,t}}_{\mathcal{N}_j}(i,j)\right\|_2 \right)}_{\texttt{term VIII}} \tag{45}$$

$$b^k_{i,t,3} \triangleq b^k_{i,t,2} + (1-\gamma) \underbrace{\left\|\nabla \tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,t}}; \mathbf{x}^{l^k_{i,t}}_{\mathcal{N}_i}(i,i)\right) - \nabla \tilde{f}_i\left(\mathbf{x}_i^{l^k_{i,t}}; \mathbf{x}^{l^k_{i,t}}_{\mathcal{N}_i}(i,i)\right)\right\|_2 \left\|\Delta\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right\|_2}_{\texttt{term VI}}. \tag{46}$$

where the quantities $b^k_{i,t,2}$ in (a), and $b^k_{i,t,3}$ in (b) are defined in (45) and (46) shown at the top of this page, respectively; furthermore in (b) we used the descent lemma, and in (c) we defined

$$b^k_{i,t,4} \triangleq b^k_{i,t,3} + \frac{L\gamma(1-\gamma)}{2}\left\|\Delta\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right\|_2^2 + (1-\gamma) \cdot \underbrace{\left\|\sum_{j\in\mathcal{N}_i}\left(\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}^{l^k_{i,t}}_{\mathcal{N}_j}\right) - \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}^{l^k_{i,t}}_{\mathcal{N}_j}(i,j)\right)\right)\right\|_2 \left\|\Delta\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right\|_2}_{\texttt{term VII}}.$$

Following similar steps, we can bound term IV as (we omit the details because of the space limit, see [50])

$$\left\langle \nabla_{\mathbf{x}_i} F(\boldsymbol{\xi}^k), \mathbf{x}_i^{k+B} - \mathbf{x}_i^{l^k_{i,T^k_i}}\right\rangle \le \frac{1-\gamma}{\gamma}\left(V\left(\mathbf{x}^{l^k_{i,T^k_i}}\right) - V\left(\mathbf{x}^{l^k_{i,T^k_i}+1}\right)\right) + \gamma\left(g_i\left(\mathbf{x}_i^{l^k_{i,T^k_i}}\right) - g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T^k_i}}\right)\right) + c^k_{i,4}, \tag{47}$$

with

$$c^k_{i,4} \triangleq c^k_{i,3} + \frac{L\gamma(1-\gamma)}{2}\left\|\Delta\widehat{\mathbf{x}}_i^{l^k_{i,T^k_i}}\right\|_2^2 + (1-\gamma) \cdot \underbrace{\left\|\sum_{j\in\mathcal{N}_i}\left(\nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}^{l^k_{i,T^k_i}}_{\mathcal{N}_j}\right) - \nabla_{\mathbf{x}_i} f_j\left(\mathbf{x}^{l^k_{i,T^k_i}}_{\mathcal{N}_j}(i,j)\right)\right)\right\|_2 \left\|\Delta\widehat{\mathbf{x}}_i^{l^k_{i,T^k_i}}\right\|_2}_{\texttt{term VII}},$$

where $c^k_{i,3}$ is defined in (48) shown at the bottom of this page. We now show that the error terms $a^k_{i,4}$, $b^k_{i,t,4}$, and $c^k_{i,4}$, are of the order $\mathcal{O}(\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i_l}\|_2^2)$. To do so, in the following we properly upper bound each term inside $a^k_{i,4}$, $b^k_{i,t,4}$, and $c^k_{i,4}$.

We begin noticing that, by the definition of $\boldsymbol{\xi}^k$, it follows

$$\begin{aligned}\|\widehat{\mathbf{x}}^h - \boldsymbol{\xi}^k\|_2 &= \|(1-\beta^k)\mathbf{x}^k + \beta^k\mathbf{x}^\star(\mathbf{x}^k) - \widehat{\mathbf{x}}^h\|_2 \\ &\le \|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2 + \|\widehat{\mathbf{x}}^h - \mathbf{x}^k\|_2 \\ &\le \|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2 + \|\Delta\widehat{\mathbf{x}}^h\| + \|\mathbf{x}^h - \mathbf{x}^k\|,\end{aligned} \tag{49}$$

for all $h \in [k, k+B-1]$

1) Bounding $a^k_{i,1}$: There holds

$$a^k_{i,1} \overset{(a)}{\le} \frac{3\rho L}{2}\left(2\|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2^2 + (1+\gamma^2)\|\Delta\widehat{\mathbf{x}}^{l^k_{i,0}}\|_2^2 + 2\|\mathbf{x}^{l^k_{i,0}} - \mathbf{x}^k\|_2^2\right) \overset{(b)}{\le} 3\rho L\Bigg(\kappa^2(1+L+NL_m)^2\left(\|\Delta\widehat{\mathbf{x}}^k\|_2^2 + C_2\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2\right) + (NC_2+1)(1+\gamma^2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2 + \gamma^2(B-N+1)\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2\Bigg) \overset{(c)}{\le} \rho L\beta_1\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{50}$$

where in (a) we used (49) and the Young's inequality; (b) follows from (23), (24), and the fact that, for any $k \ge 0$

$$\begin{aligned}&\|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2 \\ &\overset{B1}{\le} \kappa\|\mathbf{x}^k - \mathtt{prox}_G\left(\nabla_{\mathbf{x}}F(\mathbf{x}^k) - \mathbf{x}^k\right)\|_2 \\ &\overset{(17)}{\le} \kappa(1+L+NL_m)\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \mathbf{x}^k\|_2 \\ &\le \kappa(1+L+NL_m)\left(\|\widehat{\mathbf{x}}(\bar{\mathbf{x}}^k) - \widehat{\mathbf{x}}^k\|_2 + \|\Delta\widehat{\mathbf{x}}^k\|_2\right),\end{aligned} \tag{51}$$

and in (c) we used (24) and defined

$$\beta_1 \triangleq C_2\left(\kappa^2(1+L+NL_m)^2(2N+1) + N(1+\gamma^2)\right) + \kappa^2(1+L+NL_m)^2 + 1 + \gamma^2(B-N+2).$$

$$c^k_{i,3} \triangleq \underbrace{\rho L\left\|\widehat{\mathbf{x}}^{l^k_{i,T^k_i}} - \boldsymbol{\xi}^k\right\|_2\left\|\mathbf{x}_i^{l^k_{i,T^k_i}} - \mathbf{x}_i^{k+B}\right\|_2}_{c^k_{i,1}} + (1-\gamma)\underbrace{\left\|\nabla\tilde{f}_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T^k_i}}; \mathbf{x}^{l^k_{i,T^k_i}}_{\mathcal{N}_i}(i,i)\right) - \nabla\tilde{f}_i\left(\mathbf{x}_i^{l^k_{i,T^k_i}}; \mathbf{x}^{l^k_{i,T^k_i}}_{\mathcal{N}_i}(i,i)\right)\right\|_2\left\|\Delta\widehat{\mathbf{x}}_i^{l^k_{i,T^k_i}}\right\|_2}_{\texttt{term VI}}$$
$$+ \underbrace{\left\|\mathbf{x}_i^{l^k_{i,T^k_i}} - \mathbf{x}_i^{k+B}\right\|_2\left(L_i\left\|\widehat{\mathbf{x}}^{l^k_{i,T^k_i}}_{\mathcal{N}_i} - \mathbf{x}^{l^k_{i,T^k_i}}_{\mathcal{N}_i}(i,i)\right\|_2 + L\sum_{j\in\mathcal{N}_i\setminus\{i\}}\left\|\widehat{\mathbf{x}}^{l^k_{i,T^k_i}}_{\mathcal{N}_j} - \mathbf{x}^{l^k_{i,T^k_i}}_{\mathcal{N}_j}(i,j)\right\|_2\right)}_{\texttt{term IX}}. \tag{48}$$

2) Bounding $b^k_{i,t,1}$ and $c^k_{i,1}$: for $t \in [1, T^k_i - 1]$

$$b^k_{i,t,1} \overset{(a)}{\leq} \frac{\rho L}{2}\Big(3\|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2^2 + (3+\gamma^2)\|\Delta\widehat{\mathbf{x}}^{l^k_{i,t}}_i\|_2^2 + 3\|\mathbf{x}^{l^k_{i,t}}_i - \mathbf{x}^k\|_2^2\Big) \overset{(b)}{\leq} \frac{\rho L}{2}\Bigg(6\kappa^2(1+L+NL_m)^2\Bigg(\|\Delta\widehat{\mathbf{x}}^k\|_2^2 + C_2\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2\Bigg) + 2(NC_2+1)(3+\gamma^2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2 + 3\gamma^2(B-N+1)\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2\Bigg) \overset{(c)}{\leq} \rho L\beta_2\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{52}$$

where in (a) we used (49) and the Young's inequality; (b) follows from (23), (24), (51); and in (c) we used (24) and defined

$$\beta_2 \triangleq C_2\Big(3\kappa^2(1+L+NL_m)^2(2N+1) + N(3+\gamma^2)\Big) + 6\kappa^2(1+L+NL_m)^2 + 3 + \frac{\gamma^2}{2}(3B-3N+5).$$

Following the same steps as in (52), it is not difficult to prove

$$c^k_{i,1} \leq \rho L\beta_2\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2. \tag{53}$$

3) Bounding `term V` : There holds

$$\texttt{term V} \overset{(a)}{\leq} 2\|\mathbf{x}^k - \mathbf{x}^\star(\mathbf{x}^k)\|_2^2 + 2\gamma^2\Bigg(\Big\|\Delta\widehat{\mathbf{x}}^{l^k_{i,0}}_i\Big\|_2^2 + (L_i^2 + L^2(\rho-1))\Bigg(\|\Delta\widehat{\mathbf{x}}^{l^k_{i,0}}\|_2^2 + D\gamma^2\sum_{l=l^k_{i,0}-D}^{l^k_{i,0}-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2\Bigg)\Bigg) \overset{(b)}{\leq} \beta_4\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{54}$$

where in (a) we used (12) and the Young's inequality; and in (b) we used (23), (24), (51), and defined

$$\beta_4 \triangleq 2C_2\Big(2\kappa^2(1+L+NL_m)^2(2N+1) + N\left(L_m^2 + L^2(\rho-1)\right)\Big) + 2\kappa^2(1+L+NL_m)^2 + \left(L_m^2 + L^2(\rho-1)\right)(1+D\gamma^2) + 2\gamma^2.$$

4) Bounding `term VI` : For $t \in [0, T^k_i]$,

$$\texttt{term VI} \overset{(a)}{\leq} (L^2+L_i^2)\|\mathbf{x}^{l^k_{i,t}}(i,i) - \widehat{\mathbf{x}}^{l^k_{i,t}}\|_2^2 + \frac{1}{2}\Big\|\Delta\widehat{\mathbf{x}}^{l^k_{i,t}}_i\Big\|_2^2 \overset{(12)}{\leq} (L^2+L_i^2)\Bigg(2\Big\|\Delta\widehat{\mathbf{x}}^{l^k_{i,t}}\Big\|_2^2 + 2D\gamma^2\sum_{h=l^k_{i,t}-D}^{l^k_{i,t}-1}\|\Delta\widehat{\mathbf{x}}^h_{i^h}\|_2^2\Bigg) + \frac{1}{2}\Big\|\Delta\widehat{\mathbf{x}}^{l^k_{i,t}}_i\Big\|_2^2 \overset{(b)}{\leq} \beta_3\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{55}$$

where in (a) we used A2, B2, B3, and the Young's inequality; and in (b) we used (24) and defined

$$\beta_3 \triangleq 2(L^2+L_m^2)\left(2NC_2 + D\gamma^2 + 1\right) + \frac{1}{2}.$$

5) Bounding `term VII` : For $t \in [0, T^k_i]$,

$$\texttt{term VII} \overset{(a)}{\leq} \frac{1}{2}\Bigg(\rho L^2\sum_{j\in\mathcal{N}_i}\Big\|\mathbf{x}^{l^k_{i,t}} - \mathbf{x}^{l^k_{i,t}}(i,j)\Big\|_2^2 + \Big\|\Delta\widehat{\mathbf{x}}^{l^k_{i,t}}_i\Big\|_2^2\Bigg) \overset{(12)}{\leq} \frac{1}{2}\Bigg(\rho^2L^2D^2\gamma^2\sum_{l=l^k_{i,t}-D}^{l^k_{i,t}-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2 + \Big\|\Delta\widehat{\mathbf{x}}^{l^k_{i,t}}_i\Big\|_2^2\Bigg) \leq \frac{\rho^2L^2D^2\gamma^2+1}{2}\sum_{l=k-D}^{k+B-2}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{56}$$

where in (a) we used A2 and the Young's inequality.

6) Bounding `term VIII` *and* `term IX` : For $t \in [1, T^k_i - 1]$

$$\texttt{term VIII} \overset{(a)}{\leq} \gamma^2\Big\|\Delta\widehat{\mathbf{x}}^{l^k_{i,t}}_i\Big\|_2^2 + (L_i^2 + L^2(\rho-1))\Bigg(\Big\|\Delta\widehat{\mathbf{x}}^{l^k_{i,t}}\Big\|_2^2 + D\gamma^2\sum_{l=l^k_{i,t}-D}^{l^k_{i,t}-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2\Bigg)\Bigg) \overset{(b)}{\leq} \beta_5\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{57}$$

where in (a) we used (12) and the Young's inequality; and in (b) we used (24), and defined

$$\beta_5 \triangleq \left(L_m^2 + L^2(\rho-1)\right)\left(2NC_2 + D\gamma^2 + 2\right) + \gamma^2.$$

As done in (57), it is easy to prove that

$$\texttt{term IX} \leq \beta_5\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2. \tag{58}$$

Using the above results, we can bound $a^k_{i,4}$, $b^k_{i,t,4}$, and $c^k_{i,4}$. According to definition of $a^k_{i,4}$, we have

$$a^k_{i,4} \overset{(50),(55)-(54)}{\leq} \alpha_1\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{59}$$

where

$$\alpha_1 \triangleq \Bigg((1-\gamma)\left(\beta_3 + \frac{L\gamma(\rho^2LD^2\gamma+1)+1}{2}\right) + \rho L\beta_1 + \beta_4\Bigg). \tag{60}$$

For $b^k_{i,t,4}$ and $c^k_{i,4}$, we have: $t \in [1, T^k_i - 1]$,

$$b^k_{i,i,4};\, c^k_{i,4} \overset{(52)-(56),(57),(58)}{\leq} \alpha_2\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}^l_{i^l}\|_2^2, \tag{61}$$

where

$$\alpha_2 \triangleq \Bigg(\left(1-\gamma\right)\left(\beta_3 + \frac{L\gamma(\rho^2LD^2\gamma+1)+1}{2}\right) + \rho L\beta_2 + \beta_5\Bigg). \tag{62}$$

Combining (40), (41), (44), (47), (59), and (61) yields:

$$V(\mathbf{x}^{k+B}) - V(\mathbf{x}^\star(\mathbf{x}^k)) \leq \frac{1-\gamma}{\gamma}\left(V(\mathbf{x}^k) - V(\mathbf{x}^{k+B})\right)$$

$$+\sum_{i=1}^{N}\left(\gamma\left(g_i\left(\mathbf{x}_i^{l^k_{i,T^k_i}}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T^k_i}}\right)\right)\right.$$
$$+\gamma\sum_{t=1}^{T^k_i-1}\left(g_i\left(\mathbf{x}_i^{l^k_{i,t}}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right)\right)+(\gamma-1)g_i\left(\mathbf{x}_i^{l^k_{i,0}}\right)$$
$$\left.-\gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,0}}\right)+g_i\left(\mathbf{x}_i^{k+B}\right)\right)+(N\alpha_1$$
$$+(B-N)\alpha_2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{il}^{l}\|_2^2\overset{A3}{\leq}\frac{1-\gamma}{\gamma}\left(V(\mathbf{x}^k)\right.$$
$$\left.-V(\mathbf{x}^{k+B})\right)+\sum_{i=1}^{N}\left(\gamma\left(g_i\left(\mathbf{x}_i^{l^k_{i,T^k_i}}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T^k_i}}\right)\right)\right.$$
$$+\gamma\sum_{t=1}^{T^k_i-1}\left(g_i\left(\mathbf{x}_i^{l^k_{i,t}}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right)\right)+(\gamma-1)g_i\left(\mathbf{x}_i^{l^k_{i,0}}\right)$$
$$\left.-\gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,0}}\right)+(1-\gamma)g_i\left(\mathbf{x}_i^{l^k_{i,T^k_i}}\right)+\gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T^k_i}}\right)\right)$$
$$+(N\alpha_1+(B-N)\alpha_2))\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{il}^{l}\|_2^2$$
$$=\frac{1-\gamma}{\gamma}\left(V(\mathbf{x}^k)-V(\mathbf{x}^{k+B})\right)+\sum_{i=1}^{N}\left(g_i\left(\mathbf{x}_i^{l^k_{i,T^k_i}}\right)\right.$$
$$+\gamma\sum_{t=1}^{T^k_i-1}\left(g_i\left(\mathbf{x}_i^{l^k_{i,t}}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right)\right)$$
$$+(\gamma-1)g_i\left(\mathbf{x}_i^{l^k_{i,0}}\right)-\gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,0}}\right)+(N\alpha_1$$
$$+(B-N)\alpha_2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{il}^{l}\|_2^2\overset{A3}{\leq}\frac{1-\gamma}{\gamma}\left(V(\mathbf{x}^k)\right.$$
$$\left.-V(\mathbf{x}^{k+B})\right)+\sum_{i=1}^{N}\left((1-\gamma)g_i\left(\mathbf{x}_i^{l^k_{i,T^k_i-1}}\right)+\gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,T^k_i-1}}\right)\right.$$
$$+\gamma\sum_{t=1}^{T^k_i-1}\left(g_i\left(\mathbf{x}_i^{l^k_{i,t}}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right)\right)$$
$$+(\gamma-1)g_i\left(\mathbf{x}_i^{l^k_{i,0}}\right)-\gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,0}}\right)$$
$$+(N\alpha_1+(B-N)\alpha_2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{il}^{l}\|_2^2$$
$$=\frac{1-\gamma}{\gamma}\left(V(\mathbf{x}^k)-V(\mathbf{x}^{k+B})\right)+\sum_{i=1}^{N}\left(g_i\left(\mathbf{x}_i^{l^k_{i,T^k_i-1}}\right)\right.$$
$$+\gamma\sum_{t=1}^{T^k_i-2}\left(g_i\left(\mathbf{x}_i^{l^k_{i,t}}\right)-g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,t}}\right)\right)$$
$$+(\gamma-1)g_i\left(\mathbf{x}_i^{l^k_{i,0}}\right)-\gamma g_i\left(\widehat{\mathbf{x}}_i^{l^k_{i,0}}\right)$$
$$+(N\alpha_1+(B-N)\alpha_2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{il}^{l}\|_2^2\leq\frac{1-\gamma}{\gamma}\left(V(\mathbf{x}^k)\right.$$
$$\left.-V(\mathbf{x}^{k+B})\right)+(N\alpha_1+(B-N)\alpha_2)\sum_{l=k-D}^{k+B-1}\|\Delta\widehat{\mathbf{x}}_{il}^{l}\|_2^2.$$

REFERENCES

[1] R. Carli and G. Notarstefano, "Distributed partition-based optimization via dual decomposition," in *Proc. IEEE 52nd Conf. Decis. Control*, 2013, pp. 2979–2984.

[2] V. Kekatos and G. B. Giannakis, "Distributed robust power system state estimation," *IEEE Trans. Power Syst.*, vol. 28, no. 2, pp. 1617–1626, May 2013.

[3] T. Erseghe, "A distributed and scalable processing method based upon ADMM," *IEEE Signal Process. Lett.*, vol. 19, no. 9, pp. 563–566, Sep. 2012.

[4] Z.-Q. Luo and P. Tseng, "Error bounds and convergence analysis of feasible descent methods: A general approach," *Ann. Oper. Res.*, vol. 46, no. 1, pp. 157–178, 1993.

[5] Z.-Q. Luo and P. Tseng, "On the linear convergence of descent methods for convex essentially smooth minimization," *SIAM J. Control Optim.*, vol. 30, no. 2, pp. 408–425, 1992.

[6] P. Tseng, "On the rate of convergence of a partially asynchronous gradient projection algorithm," *SIAM J. Optimiz.*, vol. 1, no. 4, pp. 603–619, 1991.

[7] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Math. Program.*, vol. 117, no. 1\2, pp. 387–423, 2009.

[8] H. Zhang, J. Jiang, and Z.-Q. Luo, "On the linear convergence of a proximal gradient method for a class of nonsmooth convex minimization problems," *J. Oper. Res. Soc. China*, vol. 1, no. 2, pp. 163–186, 2013.

[9] D. Drusvyatskiy and A. S. Lewis, "Error bounds, quadratic growth, and linear convergence of proximal methods," *Math. Oper. Res.*, vol. 43, no. 3, pp. 919–948, 2018.

[10] Y. Tian, Y. Sun, and G. Scutari, "Achieving linear convergence in distributed asynchronous multi-agent optimization," *IEEE Trans. Autom. Control*, to be published.

[11] Y. Sun, A. Daneshmand, and G. Scutari, "Distributed optimization based on gradient-tracking revisited: Enhancing convergence rate via surrogation," 2020, *arXiv:1905.02637*.

[12] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM J. Optim.*, vol. 25, no. 2, pp. 944–966, 2015.

[13] J. Tsitsiklis, D. Bertsekas, and M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms," *IEEE Trans. Autom. Control*, vol. 31, no. 9, pp. 803–812, Sep. 1986.

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

[15] L. Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari, "Asynchronous parallel algorithms for nonconvex optimization," *Math. Program.*, vol. 184, pp. 121–154, 2020.

[16] D. Davis, B. Edmunds, and M. Udell, "The sound of APALM clapping: Faster nonsmooth nonconvex optimization with stochastic asynchronous PALM," in *Proc. 30th Conf. Adv. Neural Inf. Process. Syst. 29*, 2016, pp. 226–234.

[17] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Englewood Cliffs, NJ, USA: Prentice Hall, vol. 23, 1989.

[18] F. Niu, B. Recht, C. Ré, and S. J. Wright, "HOGWILD: A lock-free approach to parallelizing stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst. 24*, 2011, pp. 693–701.

[19] X. Lian, Y. Huang, Y. Li, and J. Liu, "Asynchronous parallel stochastic gradient for nonconvex optimization," in *Proc. Adv. Neural Inf. Process. Syst. 28*, 2015, pp. 2719–2727.

[20] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *Proc. IEEE 52nd Conf. Decis. Control*, 2013, pp. 3671–3676.

[21] E. Wei and A. Ozdaglar, "On the o (1= k) convergence of asynchronous distributed alternating direction method of multipliers," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2013, pp. 551–554.
[22] P. Bianchi, W. Hachem, and F. Iutzeler, "A coordinate descent primal-dual algorithm and application to distributed asynchronous optimization," *IEEE Trans. Autom. Control*, vol. 61, no. 10, pp. 2947–2957, Oct. 2016.
[23] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 4, pp. 772–790, Aug. 2011.
[24] I. Notarnicola and G. Notarstefano, "Asynchronous distributed optimization via randomized dual proximal gradient," *IEEE Trans. Autom. Control*, vol. 62, no. 5, pp. 2095–2106, May 2017.
[25] J. Xu, S. Zhu, Y. C. Soh, and L. Xie, "Convergence of asynchronous distributed gradient methods over stochastic networks," *IEEE Trans. Autom. Control*, vol. 63, no. 2, pp. 434–448, Feb. 2018.
[26] I. Notarnicola and G. Notarstefano, "A randomized primal distributed algorithm for partitioned and big-data non-convex optimization," *IEEE 55th Conf. Decis. Control*, 2016, pp. 153–158.
[27] A. Nedić, "Asynchronous broadcast-based convex optimization over a network," *IEEE Trans. Autom. Control*, vol. 56, no. 6, pp. 1337–1351, Jun. 2011.
[28] 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, Feb. 2015.
[29] 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.
[30] K. I. Tsianos and M. G. Rabbat, "Distributed dual averaging for convex optimization under communication delays," *IEEE Amer. Control Conf.*, 2012, pp. 1067–1072.
[31] K. I. Tsianos and M. G. Rabbat, "Distributed consensus and optimization under communication delays," *IEEE 49th Allerton Conf. Commun., Control, Comput.*, pp. 974–982, 2011.
[32] 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.
[33] T. T. Doan, C. L. Beck, and R. Srikant, "Impact of communication delays on the convergence rate of distributed optimization algorithms," 2017, *arXiv:1708.03277*.
[34] 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, Feb. 2015.
[35] 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, Mar. 2017.
[36] 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. 4, no. 2, pp. 293–307, Jun. 2018.
[37] 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.
[38] N. Bof, R. Carli, G. Notarstefano, L. Schenato, and D. Varagnolo, "Newton-Raphson consensus under asynchronous and lossy communications for peer-to-peer networks," 2017, *arXiv:1707.09178*.
[39] M. Hong, "A distributed, asynchronous and incremental algorithm for nonconvex optimization: An ADMM approach," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 3, pp. 935–945, Sep. 2018.
[40] S. M. Shah and K. E. Avrachenkov, "Linearly convergent asynchronous distributed ADMM via Markov sampling," 2020, *arXiv:1810.05067*.
[41] R. Zhu, D. Niu, and Z. Li, "A block-wise, asynchronous and distributed ADMM algorithm for general form consensus optimization," 2018, *arXiv:1802.08882*.
[42] T.-H. Chang, M. Hong, W.-C. Liao, and X. Wang, "Asynchronous distributed ADMM for large-scale optimization part I: Algorithm and convergence analysis," *IEEE Trans. Signal Process*, vol. 64, no. 12, pp. 3118–3130, 2016.
[43] M. Ma, J. Ren, G. B. Giannakis, and J. Haup, "Fast asynchronous decentralized optimization: Allowing multiple masters," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2018, pp. 633–637.
[44] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proc. 31st Int. Conf. Mach. Learn.*, 2014, pp. 1701–1709.
[45] S. Jiang, Y. Lei, S. Wang, and D. Wang, "An asynchronous ADMM algorithm for distributed optimization with dynamic scheduling strategy," in *Proc. IEEE 21st Int. Conf. HPCC; IEEE 17th Int. Conf. SmartCity; IEEE 5th Int. Conf. DSS*, 2019.
[46] N. Srebro, J. Rennie and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst. 17*, 2005, pp. 1329–1336.
[47] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. Ser. B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.
[48] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel selective algorithms for nonconvex big data optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1874–1889, Apr. 2015.
[49] L. Cannelli, F. Facchinei, V. Kungurtsev, and G. Scutari, "Asynchronous parallel algorithms for nonconvex big-data optimization Part II: Complexity and numerical results," 2017, *arXiv:1701.04900*.
[50] L. Cannelli, F. Facchinei, G. Scutari, and V. Kungurtsev, "Asynchronous optimization over graphs: Linear convergence under error bound conditions," 2020, *arXiv:2010.09057*.

Loris Cannelli (Member, IEEE) received the B.S. and M.S. degrees in electrical and telecommunication engineering from the University of Perugia, Italy, in 2010 and 2013, respectively, the M.S. degree in electrical engineering from State University of New York at Buffalo, NY, USA, in 2015, and the Ph.D. degree in industrial engineering from the Purdue University, West Lafayette, IN, USA, in 2019.

His research interests include optimization algorithms, machine learning, and big-data analytics.

Francisco Facchinei received the Ph.D. degree in system engineering from the University of Rome, "La Sapienza," Rome, Italy, in 1990.

He is a Full Professor of operations research, Engineering Faculty, University of Rome, "La Sapienza." His research interests include theoretical and algorithmic issues related to nonlinear optimization, variational inequalities, complementarity problems, equilibrium programming, and computational game theory.

Gesualdo Scutari (Senior Member, IEEE) 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. His research interests include continuous and distributed optimization, equilibrium programming, and their applications to signal processing and machine learning.

Dr. Scutari was the recipient of the 2006 Best Student Paper Award at the IEEE ICASSP 2006, the 2013 NSF CAREER Award, and the 2015 IEEE Signal Processing Society Young Author Best Paper Award. He is a Senior Area Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and an Associate Editor for *SIAM Journal on Optimization*.

Vyacheslav Kungurtsev (Member, IEEE) received the B.S. degree in mathematics from Duke University, Durham, NC, USA, in 2007, and the Ph.D. degree in mathematics with a specialization in computational science from the University of California - San Diego, La Jolla, CA, USA, in 2013.

He spent one year as Postdoctoral Researcher with the KU Leuven, Leuven, Belgium, for the Optimization for Engineering Center, and since 2014, he has been a Researcher at Czech Technical University in Prague, Czechia, working on various aspects of continuous optimization.