The Distributed Dual Ascent Algorithm is Robust to Asynchrony

Bianchi, Mattia; Ananduta, Wicak; Grammatico, Sergio

DOI
10.1109/LCSYS.2021.3084883

Publication date
2022

Document Version
Final published version

Published in
IEEE Control Systems Letters

The Distributed Dual Ascent Algorithm is Robust to Asynchrony

Mattia Bianchi, *Member, IEEE*, Wicak Ananduta, and Sergio Grammatico, *Senior Member, IEEE*

***Abstract*—The distributed dual ascent is an established algorithm to solve strongly convex multi-agent optimization problems with separable cost functions, in the presence of coupling constraints. In this letter, we study its asynchronous counterpart. Specifically, we assume that each agent only relies on the outdated information received from some neighbors. Differently from the existing randomized and dual block-coordinate schemes, we show convergence under heterogeneous delays, communication and update frequencies. Consequently, our asynchronous dual ascent algorithm can be implemented without requiring any coordination between the agents.**

***Index Terms*—Optimization algorithms, networked control systems, variational methods.**

I. Introduction

DISTRIBUTED multi-agent optimization is well suited for modern large-scale and data-intensive problems, where the volume and spatial scattering of the information render centralized processing and storage inefficient or infeasible. Engineering applications arise in power systems [1], communication networks [2], machine learning [3] and robotics [4], just to name a few. A prominent role is played by asynchronous algorithms, where communication and updates of the local processors are not coordinated. The asynchronous approach is advantageous in several ways: it eliminates the need for synchronization, which is costly in large networks; it reduces the idle time, when distinct processors have different computational capabilities; it enhances robustness with respect to unreliable, lossy and delayed, communication; and it alleviates transmission and memory-access congestion. On this account, in this letter we study a completely asynchronous implementation of the distributed dual ascent, a fundamental algorithm for constrained optimization.

Literature review: The dual ascent consists of solving the dual problem via the gradient method. Its major advantage with respect to augmented Lagrangian methods (e.g., method of multipliers and alternating direction method of multipliers (ADMM)) is decomposability: for separable problems, the update breaks down into decentralized subproblems, allowing for distributed and parallel (as opposed to sequential) implementation. Although this "dual decomposition" is an old idea [5], distributed algorithms based on the dual ascent are still very actively researched [6], [7], [8].

Block-coordinate versions of the dual ascent, where only part of the variables is updated at each iteration, are also explored in the literature [9]. More generally, a variety of distributed algorithms has been proposed to solve constraint-coupled optimization problems, possibly with block-updates and time-varying communication [10], [11], [12]. Nonetheless, in all the cited works, a common clock is employed to synchronize the communication and update frequencies.

On the contrary, a global clock is superfluous for asynchronous methods. Since the seminal work [13], distributed asynchronous optimization algorithms have received increasing attention, especially with regards to primal schemes [14], [15]. Most of the available asynchronous dual approaches leverage randomized activation [16], [17], [18], [19], where the update of each agent is triggered by a local timer or by signals received from the neighbors. However, no delay is tolerated, i.e., the agents perform their computations using the most recent information. This requires some coordination, since each agent must wait for its neighbors to complete their tasks before starting a new update. To deal with delays caused by imperfect communication or non-negligible computational time, primal-dual [20] and ADMM-type [21], [22] algorithms have been analyzed. Yet, the method in [21] requires the presence of a master node; meanwhile, in [20], [22], the delays are assumed to be independent of the activation sequences, which is not realistic [15], [20].

Contributions: In this letter, we propose an asynchronous implementation of the distributed dual ascent, according to the celebrated *partially asynchronous* model, devised by Bertsekas and Tsitsiklis [13, Sec. 7], where: (a) agents perform updates and send data at any time, without any need for coordination signals; (b) the agents use outdated information from their neighbors to perform their updates. In particular, we allow some agents to transmit more frequently, process faster and execute more iterations than others, based exclusively on their local clocks. The model also encompasses the presence of heterogeneous delays or dropouts in the

Manuscript received March 4, 2021; revised May 4, 2021; accepted May 21, 2021. Date of publication May 28, 2021; date of current version June 28, 2021. This work was supported in part by Netherlands Organization for Scientific Research (NWO) through Research Project OMEGA under Grant 613.001.702, and in part by the European Research Council (ERC) through Research Project COSMOS under Grant 802348. Recommended by Senior Editor F. Dabbene. *(Corresponding author: Mattia Bianchi.)*

The authors are with the Delft Center for Systems and Control, TU Delft, 2624 CD Delft, The Netherlands (e-mail: m.bianchi@tudelft.nl; w.ananduta@tudelft.nl; s.grammatico@tudelft.nl).

Digital Object Identifier 10.1109/LCSYS.2021.3084883

communication. Differently from [21], only peer-to-peer communication is required. Moreover, we do not postulate a stochastic characterization of delays or activation sequences. Instead, we merely assume bounds on the communication and update frequencies. In this partially asynchronous scenario, Low and Lapsley [23] studied a dual ascent algorithm, for a network utility maximization (NUM) problem, with affine inequality constraints modeling link capacity limits.

Here we consider a more general setting, and address the presence of convex inequality and equality constraints. Our main contribution is to prove the convergence of the sequence generated by the asynchronous distributed dual ascent to an optimal primal solution, under assumptions that are standard for its synchronous counterpart and provided that small-enough uncoordinated step sizes are employed. In fact, we drop some of the technical conditions in [23] (see Section II), and relax the need for a global step. Our strategy is to relate the algorithm to a perturbed projected scaled gradient scheme. With respect to asynchronous primal methods [13, Sec. 7.5] and to the general fixed-point algorithm in [24], the main technical challenge is that the agents are not able to compute the partial gradients of the dual function locally; as a consequence, we have to consider two layers of delays. To validate our results, we provide a numerical simulation on the optimal power flow (OPF) problem.

Notation: $\mathbb{N}$ is the set of natural numbers, including 0. $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ is the set of extended real numbers. $\mathbf{0}_n \in \mathbb{R}^n$ is the vector with all elements equal to 0; $I_n \in \mathbb{R}^{n\times n}$ is an identity matrix; we may omit the subscripts if there is no ambiguity. For an extended value function $f : \mathbb{R}^n \to \overline{\mathbb{R}}$, $\mathrm{dom}(f) := \{x \in \mathbb{R}^n \mid f(x) < \infty\}$; f is ρ-strongly convex if $x \mapsto f(x) - \frac{\rho}{2}\|x\|^2$ is convex, coercive if $\lim_{\|x\|\to\infty} f(x) = \infty$. Given a positive definite matrix $P \succ 0$, $\|x\|_P =: \sqrt{\langle x, Px\rangle}$ is the P-weighted norm; we omit the subscript if $P = I$. $\mathrm{int}(S)$ is the interior of a set S.

II. Problem Setup and Mathematical Background

Let $\mathcal{I} = \{1, 2, \ldots, N\}$ be a set of agents. The agents can communicate over an undirected network $\mathcal{G}(\mathcal{I}, \mathcal{E})$; the pair (i, j) belongs to the set of edges $\mathcal{E} \subseteq \mathcal{I} \times \mathcal{I}$ if and only if agents i and j can *occasionally* exchange information, with the convention $(i, i) \in \mathcal{E}$ for all $i \in \mathcal{I}$. We denote by $\mathcal{N}_i = \{j \mid (i, j) \in \mathcal{E}\}$ the neighbors set of agent i. The agents' common goal is to solve the following convex monotropic optimization problem, where the decision vector $x_i \in \mathbb{R}^{n_i}$ of agent i is coupled to the decisions of the neighbors $\mathcal{N}_i$ via convex shared constraints:

$$\min_{x_i\in\mathbb{R}^{n_i}, i\in\mathcal{I}} \quad \sum_{i\in\mathcal{I}} f_i(x_i) \tag{1a}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}_i} c_{i,j}(x_j) \le \mathbf{0}_{p_i}, \quad \forall i \in \mathcal{I} \tag{1b}$$

$$\sum_{j\in\mathcal{N}_i} a_{i,j}(x_j) = \mathbf{0}_{r_i}, \quad \forall i \in \mathcal{I}, \tag{1c}$$

Here, the cost $f_i : \mathbb{R}^{n_i} \to \overline{\mathbb{R}}$, the functions $\{c_{j,i} : \mathbb{R}^{n_i} \to \mathbb{R}^{p_j}, j \in \mathcal{N}_i\}$, and the affine functions $\{a_{j,i} : \mathbb{R}^{n_i} \to \mathbb{R}^{r_j} : x_i \mapsto A_{j,i}x_i - b_{j,i}, j \in \mathcal{N}_i\}$ are local data kept by agent i.

Remark 1: Problems in the form (1) arise naturally in resource allocation [2] and network flow problems [25], e.g., NUM for communication networks [7] or OPF in energy systems [1]. More generally, a well-known approach to solve a (cost-coupled) distributed optimization problem is to recast it as (1), by introducing slack variables to decouple the costs and additional consensus constraints for consistency. For instance, the problem $\{\min_{z\in\mathbb{R}} \sum_{i\in\mathcal{I}} f_i(z)\}$ is equivalent to

$$\min_{x_i\in\mathbb{R}, i\in\mathcal{I}} \sum_{i\in\mathcal{I}} f_i(x_i) \text{ s.t. } L_{-1}(\mathrm{col}((x_i)_{i\in\mathcal{I}})) = \mathbf{0}_{N-1}, \tag{2}$$

where L_{-1} is the (full row rank, see Assumption 1 below) matrix obtained by removing the first row from the Laplacian of a connected graph; indeed, (2) is an instance of (1).

In the following, we use the compact notation $x := \mathrm{col}((x_i)_{i\in\mathcal{I}}) \in \mathbb{R}^n$ and $f(x) := \sum_{i\in\mathcal{I}} f_i(x_i)$, where $n := \sum_{i\in\mathcal{I}} n_i$. Let us also define $c_{i,j}(x_j) := \mathbf{0}_{p_i}$, $A_{i,j} := \mathbf{0}_{r_i\times n_j}$ and $a_{i,j}(x_j) := \mathbf{0}_{r_i}$ for all $i \in \mathcal{I}$, $j \notin \mathcal{N}_i$; $m_i := p_i + r_i$, $g_{i,j}(x_j) := \mathrm{col}(c_{i,j}(x_j), a_{i,j}(x_j))$, $g_i(x_i) := \mathrm{col}((g_{j,i}(x_i))_{j\in\mathcal{N}})$, and $\Omega_i := \mathbb{R}^{p_i}_{\ge 0} \times \mathbb{R}^{r_i}$, for all $i, j \in \mathcal{I}$; $g(x) := \sum_{i\in\mathcal{I}} g_i(x_i)$, $m := \sum_{i\in\mathcal{I}} m_i$ and $\Omega := \prod_{i\in\mathcal{I}} \Omega_i$. We assume the following conditions throughout the paper.

Assumption 1 (Regularity and Convexity):

(i) For all $i \in \mathcal{I}$, f_i is proper, closed, and ρ_i-strongly convex, for some $\rho_i > 0$.

(ii) For all $i, j \in \mathcal{I}$, $g_{i,j}$ is componentwise convex and $\theta_{i,j}$-Lipschitz continuous on $\mathrm{dom}(f_j)$, for some $\theta_{i,j} > 0$.

(iii) There exists $\bar{x} \in \mathrm{int}(\mathrm{dom}(f))$ such that, for all $i \in \mathcal{I}$, $\sum_{j\in\mathcal{N}_i} c_{i,j}(\bar{x}_j) < \mathbf{0}_{p_i}$ and $\sum_{j\in\mathcal{N}_i} a_{i,j}(\bar{x}_j) = \mathbf{0}_{r_i}$.

(iv) The matrix $A := [A_{i,j}]_{i,j\in\mathcal{I}}$ has full row rank.

Under Assumption 1(iii), problem (1) is feasible. In addition, the strong convexity in Assumption 1(i) ensures that there exists a unique solution $x^\star$, with finite optimal value $f^\star := f(x^\star) \in \mathbb{R}$; this condition is standard for dual gradient methods [6, Asm. 2.1], [9, Asm. 1], and commonly found in the problems mentioned in Remark 1. Differently from [23, Asm. C1], we do not assume differentiability of f, nor that the functions f_i's are increasing (e.g., quadratic cost functions are allowed here). We note that local constraints can be enforced in (1) by opportunely choosing the domains of the f_i's (which need not be bounded, see [23]). We also remark that Assumption 1(ii) is automatically satisfied in the most common case of affine inequality constraints [6], [7].

Given the information available to each agent, the natural way of distributedly solving (1) is resorting to dual methods. By Assumption 1(iii), strong duality holds [26, Th. 28.2], i.e.,

$$f^\star = q^\star := \max_{y\in\Omega} q(y), \tag{3}$$

where $q : \Omega \to \mathbb{R}$ is the concave dual function,

$$q(y) := \min_{x\in\mathbb{R}^n} f(x) + \langle g(x), y\rangle, \tag{4}$$

with dual variable y, and the maximum $q^\star$ is attained in (3); we denote by $\mathcal{Y}^\star$ the convex nonempty set of dual solutions (namely, solutions of (3)). Moreover, Assumption 1 rules out the case of redundant equality constraints; together with Assumption 1(iii), it guarantees that the (convex) function $-q$ is coercive on Ω, and hence that $\mathcal{Y}^\star$ is bounded (for similar arguments, see [27, Sec. VII, Th. 2.3.2], [8, Lem. 1]). For this reason, Assumption 1(iii) and 1(iv) have been exploited for particular instances of problem (1) [6, Asm. 2.1], [8, Asm. 1].

Algorithm 1 Asynchronous Distributed Dual Ascent

Initialization: $\forall i \in \mathcal{I}$, $y_i(0) = \mathbf{0}_{m_i}$, $x_i(0) = \operatorname{argmin}_{x_i \in \mathbb{R}^{n_i}} f_i(x_i)$.

Local variables update: For all $k \in \mathbb{N}$, each agent $i \in \mathcal{I}$ does:

if $k \in T_i$:

$$\begin{cases} x_i(k+1) = \operatorname*{argmin}_{x_i \in \mathbb{R}^{n_i}} \left(f_i(x_i) + \sum_{j \in \mathcal{N}_i} \langle g_{j,i}(x_i), y_j(\tau_{i,j}^k) \rangle \right) & \text{(7a)} \\ y_i(k+1) = \operatorname{proj}_{\Omega_i} \left(y_i(k) + \gamma_i \sum_{j \in \mathcal{N}_i} g_{i,j}(x_j(\tau_{i,j}^k)) \right) & \text{(7b)} \end{cases}$$

otherwise:

$$\begin{cases} x_i(k+1) = x_i(k) & \text{(8a)} \\ y_i(k+1) = y_i(k). & \text{(8b)} \end{cases}$$

A. Synchronous Distributed Dual Ascent Algorithm

Under Assumption 1(i) and 1(ii), the concave dual function q in (4) is differentiable with Lipschitz gradient [7, Lem. II.1]. Thus, for a $\gamma > 0$ small enough, the dual ascent iteration

$$y(k+1) = \operatorname{proj}_{\Omega}(y(k) + \gamma \nabla q(y(k))) \tag{5}$$

converges to a dual solution. By the envelope theorem, $\nabla q(y) = g(x^\star(y))$, where $x^\star(y) \coloneqq \operatorname{argmin}_{x \in \mathbb{R}^n} f(x) + \langle g(x), y \rangle$; therefore, by assigning to agent i the Lagrangian multipliers $y_i \in \mathbb{R}^{m_i}$, with $y = \operatorname{col}((y_i)_{i \in \mathcal{I}})$, the update in (5) can be written, for the single agent i, for all $i \in \mathcal{I}$, as

$$x_i(k+1) = \operatorname*{argmin}_{x_i \in \mathbb{R}^{n_i}} \left(f_i(x_i) + \sum_{j \in \mathcal{N}_i} \langle g_{j,i}(x_i), y_j(k) \rangle \right) \tag{6a}$$

$$y_i(k+1) = \operatorname{proj}_{\Omega_i} \left(y_i(k) + \gamma \sum_{j \in \mathcal{N}_i} g_{i,j}(x_j(k+1)) \right), \tag{6b}$$

where the argmin is single valued by Assumption 1, and the sequence $(x(k))_{k \in \mathbb{N}}$ converges to the primal solution $x^\star$. We emphasize that computing the update in (6) requires each agent to receive information *from all of its neighbors twice* per iteration: the first time because computing $x_i(k+1)$ requires the knowledge of $\{y_j(k), j \in \mathcal{N}_i\}$; and the second time to collect the quantities $\{g_{i,j}(x_j(k+1)), j \in \mathcal{N}_i\}$.

III. Asynchronous Distributed Dual Ascent

Let us now introduce an asynchronous version of the distributed dual ascent method (6), according to the *partially asynchronous model* [13, Sec. 7.1]. The iteration is shown in Algorithm 1, and it is determined by:

- a nonempty sequence $T_i \subseteq \mathbb{N}$, for each $i \in \mathcal{I}$. Agent i performs an update only for $k \in T_i$.
- an integer variable $\tau_{i,j}^k$, with $0 \leq \tau_{i,j}^k \leq k$, for each $i \in \mathcal{I}$, $j \in \mathcal{N}_i$, $k \in \mathbb{N}$, which represents the number of steps by which the information used in the updates of (x_i, y_i) at step k is outdated. For example, $\tau_{i,j}^k = k - \delta$ means that the variable $y_j(\tau_{i,j}^k)$ that agent i uses to compute $x_i(k+1)$ is outdated by δ steps.

In particular, the following variables are available to agent i when performing the update at $k \in T_i$:

$$y_i(k), \ \{y_j(\tau_{i,j}^k) \mid j \in \mathcal{N}_i\}, \ \{g_{i,j}(x_j(\tau_{i,j}^k)) \mid j \in \mathcal{N}_i\}. \tag{9}$$

For mathematical convenience, $\tau_{i,j}^k$ is defined also for $k \notin T_i$, even if these variables are immaterial to Algorithm 1. Note that $\tau_{i,j}^k \geq 0$ implies that at $k = 0$ all the agents have updated information on the neighbors' variables; this is not restrictive and it is only meant to ease the notation (the same holds for the initialization in Algorithm 1; see also [13, Sec. 7.1]).

Remark 2: The update in (7) differs from (6) even if $\tau_{i,j}^k = k$ for all $i \in \mathcal{I}$, $j \in \mathcal{N}_i$, $k \in \mathbb{N}$, as in (6b) the agents are exploiting the most recent information $x_j(k+1)$.

In Algorithm 1, k should be regarded as an event counter, an iteration index as seen by an external observer, which is increased every time one or more agents complete an update. It is introduced to give a global description of the algorithm, but it is *not known* by the agents, who can compute and communicate *without any form of coordination*. We present a simple example in Figure 1. Indeed, the partially asynchronous model captures a plethora of asynchronous protocols (for different choices of the parameters T_i's, $\tau_{i,j}^k$'s), encompassing several scenarios where: (a) the communication links of the network $\mathcal{G}$ are lossy or active intermittently; (b) there are heterogeneous delays on the transmission; (c) the local computation time cannot be neglected; (d) the agents update their local variables at different frequencies. We refer to [13] for an exhaustive discussion.

Assumption 2 (Partial asynchronism, [13, Sec. 7.1, Asm. 1.1]): There exists a positive integer Q such that:

1) *Bounded inter-update intervals*: for all $k \in \mathbb{N}$, for all $i \in \mathcal{I}$, it holds that $\{k, k+1, \ldots, k+Q-1\} \cap T_i \neq \varnothing$;
2) *Bounded delays*: for all $k \in \mathbb{N}$, for all $i \in \mathcal{I}$ and for all $j \in \mathcal{N}_i$, it holds that $k - Q \leq \tau_{i,j}^k \leq k$.

Remark 3: Assumption 2 is mild and easily satisfied in distributes computation; for instance, it holds for the scenario in Figure 1, see [13, Sec. 7.1, Ex. 1.1], [15, Sec. III.A].

We are ready to enunciate our main result.

Theorem 1: Let Assumptions 1-2 hold. For all $i \in \mathcal{I}$, let

$$\theta_i \coloneqq \sqrt{\textstyle\sum_{j \in \mathcal{N}_i} \theta_{j,i}^2} \tag{10}$$

$$\phi_i \coloneqq \textstyle\sum_{j \in \mathcal{N}_i} \theta_j^2 / \rho_i \tag{11}$$

$$\ell_i \coloneqq \textstyle\sum_{j \in \mathcal{N}_i} \theta_{i,j} \, \theta_j / \rho_j \tag{12}$$

$$\xi_i \coloneqq \textstyle\sum_{j \in \mathcal{N}_i} \sum_{l \in \mathcal{N}_j} \theta_{l,j} \, \theta_j / \rho_j \tag{13}$$

with $\theta_{j,i}$ and ρ_j as in Assumption 1, for all $i, j \in \mathcal{I}$. Assume that, for all $i \in \mathcal{I}$, the step size $\gamma_i > 0$ is chosen such that

$$\gamma_i^{-1} > 1/2\phi_i + 3/2Q(\ell_i + \xi_i), \tag{14}$$

with Q as in Assumption 2. Then, the sequence $(x(k))_{k \in \mathbb{N}}$ generated by Algorithm 1 converges to the solution $x^\star$ of the optimization problem in (1).

Remark 4: Each agent i can locally compute the parameters ϕ_i, ℓ_i, ξ_i in Theorem 1, only based on some information from its direct neighbors. Thus, the choice of the step sizes γ_i's is decentralized, provided that the agents have access to (an upper bound for) the asynchronism bound Q (otherwise, vanishing step sizes can be considered).

Remark 5: As usual for partially asynchronous optimization algorithms [13], [15], the upper bound in (14) decreases if Q grows. This is a structural issue: we can construct a problem satisfying Assumption 1 (similarly to [13, Sec. 7.1, Ex. 1.3]) such that, for any fixed positive

γ_i's, there is a large enough Q and some sequences $\tau_{i,j}^k$'s , T_i's satisfying Assumption 2, for which Algorithm 1 diverges.

IV. Convergence Analysis

In this section we prove Theorem 1. Our idea is to relate Algorithm 1 to a perturbed scaled block coordinate version of (5), and to show that, for small-enough steps sizes, the error caused by the outdated information is also small and does not compromise convergence. With respect to the asynchronous gradient method in [13], the main technical complication is that, for each update in (5), the agents communicate twice; in turn, the update of agent i depends on the variables of its neighbors' neighbors (or *second order neighbors*). In the following, we denote by $\tilde{y}_i := \mathrm{col}((y_j)_{j\in\mathcal{N}_i})$ and $\tilde{\tilde{y}}_i := \mathrm{col}((\tilde{y}_j)_{j\in\mathcal{N}_i})$ the dual variables of the neighbors and second order neighbors of agent i, respectively.

To compare Algorithm 1 and (5), the first step is to get rid of the primal variables x_i's in Algorithm 1. Importantly, we have to take into account that the x_j's in (7b) are not only outdated, but also computed using outdated information. In fact, for any $i \in \mathcal{I}$, $k \in T_i$, the variable $x_j(\tau_{i,j}^k)$ is computed by agent $j \in \mathcal{N}_i$, according to (7a), as

$$x_j(\tau_{i,j}^k) = \operatorname*{argmin}_{x_j\in\mathbb{R}^{n_j}} f_j(x_j) + \sum_{l\in\mathcal{N}_j} \langle g_{l,j}(x_j), y_l(\bar{\tau}_{i,j,l}^k)\rangle \tag{15}$$

$$=: \mathbb{x}_j^\star(\tilde{y}_j(\bar{\tau}_{i,j}^k)), \tag{16}$$

where, for all $l \in \mathcal{N}_j$, $\bar{\tau}_{i,j,l}^k := \tau_{j,l}^p$, and p is the last time agent j performed an update prior to $\tau_{i,j}^k$ (see Figure 1), i.e.,

$$p = p(i,j,k) := \max(\{0\} \cup \{t \in T_j \mid t \le \tau_{i,j}^k - 1\}), \tag{17}$$

(and (15) also holds if $p(i,j,k) = 0$ because of the initial conditions in Algorithm 1). For brevity of notation, in (16), we define $\bar{\tau}_{i,j}^k := \mathrm{col}((\bar{\tau}_{i,j,l}^k)_{l\in\mathcal{N}_j})$ and $\tilde{y}_j(\bar{\tau}_{i,j}^k) := \mathrm{col}((y_l(\bar{\tau}_{i,j,l}^k)_{l\in\mathcal{N}_j})$. By replacing (16) in (7a), we obtain that, for all $i \in \mathcal{I}$, $k \in T_i$

$$y_i(k+1) = \mathrm{proj}_{\Omega_i}\Big(y_i(k) + \gamma_i \sum\nolimits_{j\in\mathcal{N}_i} g_{i,j}(\mathbb{x}_j^\star(\tilde{y}_j(\bar{\tau}_{i,j}^k)))\Big)$$

$$=: \mathrm{proj}_{\Omega_i}\Big(y_i(k) + \gamma_i \boldsymbol{F}_i(\tilde{\tilde{y}}_i(\bar{\tau}_i^k))\Big), \tag{18}$$

where $\bar{\tau}_i^k := \mathrm{col}((\bar{\tau}_{i,j}^k)_{j\in\mathcal{N}_i})$, $\tilde{\tilde{y}}_i(\bar{\tau}_i^k) := \mathrm{col}((\tilde{y}_j(\bar{\tau}_{i,j}^k))_{j\in\mathcal{N}_i})$. Hence, (18) expresses the update in (7b) as a function of the outdated dual variables of the second order neighbors of agent i. Since (18) makes use of second order information, the maximum delay (in this notation) is not bounded by Q anymore: instead, by (17) and Assumption 2, it holds that $p(i,j,k) - Q \le \tau_{j,l}^{p(i,j,k)} = \bar{\tau}_{i,j,l}^k \le p(i,j,k)$, $\tau_{i,j}^k - Q \le p(i,j,k) \le \max(0, \tau_{i,j}^k - 1)$, $k - Q \le \tau_{i,j}^k \le k$; thus,

$$k - 3Q \le \bar{\tau}_{i,j,l}^k \le k - 1, \tag{19}$$

for all $k \ge 1$, and $\bar{\tau}_{i,j,l}^0 = 0$. Figure 1 illustrates how the lower bound is obtained, with $l = i$.

We emphasize that the mapping $\boldsymbol{F}_i$ is not a partial gradient of the dual function (differently from the asynchronous gradient method in [13]), – e.g., its argument lies in a different space. In particular, we note that $\tilde{\tilde{y}}_i(\bar{\tau}_i^k)$ can contain multiple instances of y_l, for some $l \in \mathcal{I}$ (including y_i), with

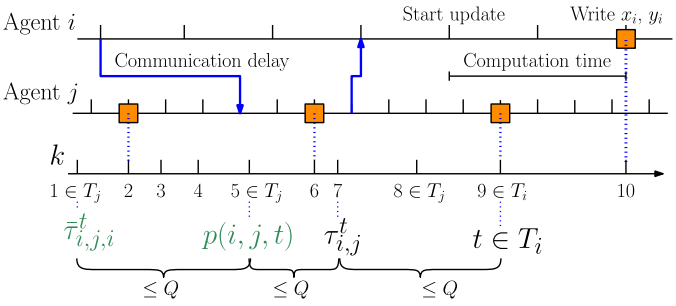

Fig. 1. An example of asynchronous updates and communication. We consider a simple scenario, where each agent periodically broadcasts its variables and computes an update using the latest data received from its neighbors, according to its *local* clock. The communication is subject to bounded delays. The figure only considers two neighboring agents i and j. The squares indicates the instants when the local variables change value; the arrows indicates data transmission. The global counter k increases every time an agent in the network completes its update. Here, agent i completes an update using the outdated information $x_j(7)$, $y_j(7)$ and k is increased to 10: hence agent i is viewed by an external observer as performing the update at $k = 9$, and $\tau_{i,j}^9 = 7$. The quantities in green are defined in Section IV.

different delays. Nonetheless, by the definition in (18), for any $y(k) \in \mathbb{R}^m$, we have

$$\boldsymbol{F}_i(\tilde{\tilde{y}}_i(k)) = \nabla_{y_i} q(y(k)). \tag{20}$$

In one word, if there is no delay, (18) corresponds to the y_i-update of the synchronous dual ascent (5) (however, there is always delay in Algorithm 1, see (19) and Remark 2).

Finally, we can also rewrite Algorithm 1 as

$$(\forall k \in \mathbb{N})(\forall i \in \mathcal{I}) \quad y_i(k+1) = y_i(k) + \gamma_i s_i(k), \tag{21}$$

where

$$s_i(k) := \frac{1}{\gamma_i}(\mathrm{proj}_{\Omega_i}(y_i(k) + \gamma_i \boldsymbol{F}_i(\tilde{\tilde{y}}_i(\bar{\tau}_i^k))) - y_i(k)), \tag{22}$$

if $k \in T_i$, $s_i(k) = \mathbf{0}_{m_i}$ otherwise.

Before proceeding with the analysis of Theorem 1, we recall the following results. The proof of Lemma 1 is standard and omitted here (see, e.g., [9, Lem. 1]).

Lemma 1: For all $j \in \mathcal{I}$, the mapping $\mathbb{x}_j^\star$ in (16) is $\frac{\theta_j}{\rho_j}$-Lipschitz continuous, with θ_j as in (10).

Lemma 2 (Weighted descent lemma, [6, Lemma 2.2]): Let $\Phi = \mathrm{diag}((\phi_i \otimes I_{m_i})_{i\in\mathcal{I}})$, ϕ_i as in (11), for all $i \in \mathcal{I}$. Let q be the dual function in (4). For any $y, z \in \mathbb{R}^m$, it holds that $q(y) \ge q(z) + \langle y - z, \nabla q(z)\rangle - 1/2\|y - z\|_\Phi^2$.

Lemma 3 [13, Lem. 5.1]: Let $s_i(k)$ be as in (22), for all $i \in \mathcal{I}$. Then, for all $i \in \mathcal{I}$, $k \in \mathbb{N}$, it holds that $\langle s_i(k), \boldsymbol{F}_i(\tilde{\tilde{y}}_i(\bar{\tau}_i^k))\rangle \ge \|s_i(k)\|^2$.

A. Proof of Theorem 1

Let $\Gamma := \mathrm{diag}((\gamma_i \otimes I_{m_i})_{i\in\mathcal{I}})$, $s(k) = \mathrm{col}((s_i(k))_{k\in\mathbb{N}})$, $\boldsymbol{F}(\tilde{\tilde{y}}(\bar{\tau}^k)) := \mathrm{col}((\boldsymbol{F}_i(\tilde{\tilde{y}}_i(\bar{\tau}_i^k)))_{i\in\mathcal{I}})$. By Lemma 2, we have

$$q(y(k+1)) = q(y(k) + \Gamma s(k))$$

$$\ge q(y(k)) - \frac{1}{2}\|\Gamma s(k)\|_\Phi^2 + \langle \Gamma s(k), \nabla q(y(k))\rangle$$

$$= q(y(k)) - \frac{1}{2}\|\Gamma s(k)\|_\Phi^2 + \langle \Gamma s(k), \boldsymbol{F}(\tilde{\tilde{y}}(\bar{\tau}^k))\rangle$$

$$
\begin{aligned}
&+\langle \Gamma s(k), \nabla q(y(k)) - \boldsymbol{F}(\tilde{\tilde{y}}(\bar{\tau}^k))\rangle \\
&= q(y(k)) - \frac{1}{2}\|\Gamma s(k)\|^2_{\Phi} + \langle \Gamma s(k), \boldsymbol{F}(\tilde{\tilde{y}}(\bar{\tau}^k))\rangle \\
&\quad +\langle \Gamma s(k), \boldsymbol{F}(\tilde{\tilde{y}}(k)) - \boldsymbol{F}(\tilde{\tilde{y}}(\bar{\tau}^k))\rangle \\
&\geq q(y(k)) - \frac{1}{2}\|s(k)\|^2_{\Phi\Gamma^2} + \|s(k)\|^2_{\Gamma} \\
&\quad +\langle \Gamma s(k), \boldsymbol{F}(\tilde{\tilde{y}}(k)) - \boldsymbol{F}(\tilde{\tilde{y}}(\bar{\tau}^k))\rangle, \qquad (23)
\end{aligned}
$$

where in the last equality we used (20) and the last inequality follows by Lemma 3 (we recall that Φ, Γ are diagonal matrices). We next bound the last addend in (23). By definition of $\boldsymbol{F}_i$ in (18) and the Cauchy–Schwartz inequality, we have

$$
\begin{aligned}
&\langle \Gamma s(k), \boldsymbol{F}(\tilde{\tilde{y}}(k)) - \boldsymbol{F}(\tilde{\tilde{y}}(\bar{\tau}^k))\rangle \\
&\geq -\sum_{i\in\mathcal{I}} \|\gamma_i s_i(k)\| \|\boldsymbol{F}_i(\tilde{\tilde{y}}_i(k)) - \boldsymbol{F}_i(\tilde{\tilde{y}}_i(\bar{\tau}_i^k))\| \\
&\geq -\sum_{i\in\mathcal{I}} \|\gamma_i s_i(k)\| \sum_{j\in\mathcal{N}_i} \|g_{i,j}(\mathbb{x}_j^\star(\tilde{\tilde{y}}_j(k))) - g_{i,j}(\mathbb{x}_j^\star(\tilde{\tilde{y}}_j(\bar{\tau}_{i,j}^k)))\| \\
&\geq -\sum_{i\in\mathcal{I}} \|\gamma_i s_i(k)\| \sum_{j\in\mathcal{N}_i} \theta_{i,j}\frac{\theta_j}{\rho_j} \|\tilde{\tilde{y}}_j(k) - \tilde{\tilde{y}}_j(\bar{\tau}_{i,j}^k)\| \\
&\geq -\sum_{i\in\mathcal{I}} \|\gamma_i s_i(k)\| \sum_{j\in\mathcal{N}_i} \theta_{i,j}\frac{\theta_j}{\rho_j} \sum_{\tau=k-3Q}^{k-1} \|\mathrm{col}((\gamma_l s_l(\tau))_{l\in\mathcal{N}_j})\|,
\end{aligned}
$$

where in the third inequality we used Lemma 1 and Assumption 1(ii), and the last follows by (21) and (19) (without loss of generality, we let $s(k) := \mathbf{0}_m$, for all $k < 0$). Therefore, by the elementary relation $2|a||b| \leq a^2 + b^2$, we obtain

$$
\begin{aligned}
&\langle \Gamma s(k), \boldsymbol{F}(\tilde{\tilde{y}}(k)) - \boldsymbol{F}(\tilde{\tilde{y}}(\bar{\tau}^k))\rangle \\
&\quad \geq -\sum_{\tau=k-3Q}^{k-1} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{N}_i} \theta_{i,j}\frac{\theta_j}{\rho_j} \frac{1}{2}\|\gamma_i s_i(k)\|^2 \\
&\qquad - \sum_{\tau=k-3Q}^{k-1} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{N}_i} \theta_{i,j}\frac{\theta_j}{\rho_j} \sum_{l\in\mathcal{N}_j} \frac{1}{2}\|\gamma_l s_l(\tau)\|^2. \qquad (24)
\end{aligned}
$$

The first term on the right-hand side of (24) equals $-\|s(k)\|^2_{\frac{3}{2}Q\Gamma^2 L}$, with $L := \mathrm{diag}((\ell_i \otimes I_{m_i})_{i\in\mathcal{I}})$ and ℓ_i as in (12). For the second term, since $\mathcal{G}$ is undirected, we reorder the addends as

$$
\begin{aligned}
&\sum_{\tau=k-3Q}^{k-1} \sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{N}_i} \theta_{i,j}\frac{\theta_j}{\rho_j} \sum_{l\in\mathcal{N}_j} \frac{1}{2}\|\gamma_l s_l(\tau)\|^2 \\
&\quad = \sum_{\tau=k-3Q}^{k-1} \sum_{i\in\mathcal{I}} \frac{1}{2}\|\gamma_i s_i(\tau)\|^2 \left(\sum_{j\in\mathcal{N}_i} \sum_{l\in\mathcal{N}_j} \theta_{l,j}\frac{\theta_j}{\rho_j}\right) \\
&\quad = \sum_{\tau=k-3Q}^{k-1} \|s(\tau)\|^2_{\frac{1}{2}\Gamma^2\Xi},
\end{aligned}
$$

where $\Xi =: \mathrm{diag}((\xi_i \otimes I_{m_i})_{i\in\mathcal{I}})$ and ξ_i as in (13). Then, by substituting in (23), we obtain

$$
\begin{aligned}
q(y(k+1)) &\geq q(y(k)) + \|s(k)\|^2_{\Gamma(I-\frac{1}{2}\Phi\Gamma-\frac{3}{2}Q L\Gamma)} \\
&\quad - \sum_{\tau=k-3Q}^{k-1} \|s(\tau)\|^2_{\frac{1}{2}\Xi\Gamma^2}, \qquad (25)
\end{aligned}
$$

with $I - \frac{1}{2}\Phi\Gamma - \frac{3}{2}QL\Gamma \succ 0$ by the assumption on Γ in (14). Since (25) holds for any k, summing over k finally yields

$$
\begin{aligned}
&q(y(k+1)) - q(y(0)) \\
&\quad \geq \sum_{\tau=0}^{k} \|s(\tau)\|^2_{\Gamma(I-\frac{1}{2}\Phi\Gamma-\frac{3}{2}QL\Gamma)} - 3Q\sum_{\tau=0}^{k} \|s(\tau)\|^2_{\frac{1}{2}\Xi\Gamma^2} \\
&\quad = \sum_{\tau=0}^{k} \|s(\tau)\|^2_{\Gamma(I-\frac{1}{2}\Phi\Gamma-\frac{3}{2}Q(\Xi+L)\Gamma)}, \qquad (26)
\end{aligned}
$$

and $I - \frac{1}{2}\Phi\Gamma - \frac{3}{2}Q(\Xi + L)\Gamma \succ 0$ by assumption (14).

We know that q is bounded above on Ω by (3), and $y(k) \in \Omega$ for all $k \in \mathbb{N}$; then, by taking the limit in (26), we have

$$
\lim_{k\to\infty} s(k) = \mathbf{0}_m. \qquad (27)
$$

Hence, by the updates in (21) and by (27), we also have

$$
\lim_{k\to\infty} \|y(k+1) - y(k)\| = 0. \qquad (28)
$$

Similarly, since, by Assumption 2(ii) and (21), $\|y_j(k) - y_j(\tau_{i,j}^k)\| \leq \sum_{\tau=k-Q}^{k-1} \gamma_j \|s_j(\tau)\|$, it also follows that

$$
\lim_{k\to\infty} \|y_j(k) - y_j(\tau_{i,j}^k)\| = 0, \ \forall i \in \mathcal{I}, j \in \mathcal{N}_i. \qquad (29)
$$

For any $i \in \mathcal{I}$, consider the *subsequence* $(s_i(k))_{k\in T_i}$, which converges to $\mathbf{0}$ by (27). In view of (22), $(\mathrm{proj}_{\Omega_i}(y_i(k) + \gamma_i \boldsymbol{F}_i(\tilde{\tilde{y}}_i(\bar{\tau}_i^k))) - y_i(k))_{k\in T_i} \to \mathbf{0}$. Therefore, by leveraging the continuity of $\boldsymbol{F}_i$ (which directly follows by the definition in (18) and Lemma 1) and of the projection [13, Sec. 3.3, Prop. 3.2], (29) and (20) yield $(\mathrm{proj}_{\Omega_i}(y_i(k) + \gamma_i \nabla_{y_i} q(y(k)) - y_i(k)))_{k\in T_i} \to \mathbf{0}$. However, again by continuity, (28) and Assumption 2(i), we can also infer convergence of the whole sequence, $\lim_{k\to\infty} \mathrm{proj}_{\Omega_i}(y_i(k) + \gamma_i \nabla_{y_i} q(y(k)) - y_i(k) = \mathbf{0}_{m_i}$, or

$$
\lim_{k\to\infty} \left(\mathrm{proj}_{\Omega_i}(\, y(k) + \Gamma \nabla q(y(k)) - y(k)\right) = \mathbf{0}_m. \qquad (30)
$$

We note that $q(y(k)) \geq q(y(0))$ for all $k \in \mathbb{N}$, by (26); moreover, $-q$ is coercive on Ω by Assumption 1(iii) and 1(iv). We conclude that the sequence $(y(k))_{k\in\mathbb{N}}$ is bounded; in turn, (30) implies that $(y(k))_{k\in\mathbb{N}}$ converges to the set of dual solutions $\mathcal{Y}^\star$.

We can finally turn our attention to the primal problem (1). By (7a), for any $i \in \mathcal{I}$, $k \in T_i$, we have $x_i(k+1) = \mathbb{x}_i^\star(\tilde{\tilde{y}}_i(\tau_i^k))$, where $\mathbb{x}_i^\star$ is defined in (16) and $\tau_i^k := \mathrm{col}((\tau_{i,j}^k)_{j\in\mathcal{N}_i})$. Moreover, by strong duality, for any $y^\star = \mathrm{col}((y_i^\star)_{i\in\mathcal{I}}) \in \mathcal{Y}^\star$, it holds that $\mathbb{x}_i^\star(\tilde{y}_i^\star) = x_i^\star$, with $\tilde{y}_i^\star := \mathrm{col}((y_j^\star)_{j\in\mathcal{N}_i})$ and $\mathrm{col}((x_i^\star)_{i\in\mathcal{I}}) = x^\star$. Therefore we can exploit (29), the fact that $(y(k))_k$ is converging to $\mathcal{Y}^\star$, and Lipschitz continuity of $\mathbb{x}_i^\star$ in Lemma 1, to conclude that $(x_i(k+1))_{k\in T_i} \to x_i^\star$. The conclusion follows because the convergence also holds for the whole sequence, i.e., $(x_i(k))_{k\in\mathbb{N}} \to x_i^\star$, by (8a).

V. Numerical Simulation

We consider an OPF problem on the IEEE 14-bus network [9, Fig. 2]. Each bus $i \in \mathcal{I} = \{1, \ldots, 14\}$ has a decision variable $x_i = \mathrm{col}(P_i, \psi_i) \in \mathbb{R}^2$, where $P_i \geq 0$ is the power generated, bounded by generation capacities, and ψ_i is the voltage phase of bus i. The goal is to minimize the sum

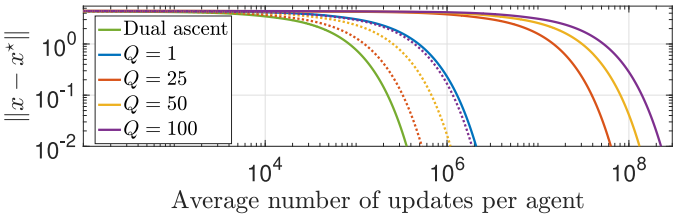

Fig. 2. Distance from the optimum, with step sizes chosen to satisfy their theoretical upper bounds (solid lines) and 100 times larger (dotted lines).

of strongly convex quadratic local costs, subject to the coupling flow constraints $\{P_i - P_i^{\mathrm{d}} = \sum_{j\in\mathcal{N}_i} B_{i,j}(\psi_i - \psi_j), \forall i \in \mathcal{I}\}$, where $P_i^{\mathrm{d}} \geq 0$ is the power demand at bus i and $B_{i,j}$ is the susceptance of line (i, j), which represent the direct current approximation of the power flow equations. We simulate Algorithm 1 for the setup in Figure 1, with randomly chosen delays and local clock frequencies. We compare four scenarios, resulting in different values for Q, with the synchronous dual ascent (6), in Figure 2. The case $Q = 1$ corresponds to a synchronous algorithm, where all the agents update their variables at every iteration. For $Q > 1$, the agents perform updates asynchronously, according to their own clocks. To compare synchronous and asynchronous implementation, we take into account the overall computation burden, i.e., the average number of updates performed per agent. For large values of Q, the upper bounds on the step sizes γ_i's in (14) decrease, resulting in slower convergence. However, the bounds can be conservative. In fact, Algorithm 1 still converges with step sizes set 100 times larger than their theoretical upper bounds for $Q = 25, 50, 100$ (but not for $Q = 1$).

VI. Conclusion

The distributed dual ascent retains its convergence properties even if the updates are carried out completely asynchronously and using delayed information, provided that small enough uncoordinated step sizes are chosen. Convergence rates for primal-dual methods in this general asynchronous scenario are currently unknown.

References

[1] D. K. Molzahn *et al.*, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Nov. 2017.

[2] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Trans. Commun.*, vol. 52, no. 7, pp. 1136–1144, Jul. 2004.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[4] M. G. Rabbat and R. D. Nowak, "Decentralized source localization and tracking [wireless sensor networks]," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 3, 2004, pp. 921–924.

[5] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, no. 3, pp. 399–417, 1963.

[6] I. Necoara and V. Nedelcu, "On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems," *Automatica*, vol. 55, pp. 209–216, May 2015.

[7] A. Beck, A. Nedić, A. Ozdaglar, and M. Teboulle, "An $o(1/k)$ gradient method for network resource allocation problems," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 64–73, Mar. 2014.

[8] A. Nedić and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1757–1780, 2009.

[9] W. Ananduta, C. Ocampo-Martinez, and A. Nedić, "Accelerated multi-agent optimization method over stochastic networks," in *Proc. 59th IEEE Conf. Decis. Control (CDC)*, 2020, pp. 2961–2966.

[10] A. Camisa, F. Farina, I. Notarnicola, and G. Notarstefano, "Distributed constraint-coupled optimization over random time-varying graphs via primal decomposition and block subgradient approaches," in *Proc. 58th IEEE Conf. Decis. Control*, 2019, pp. 6374–6379.

[11] A. Falsone and M. Prandini, "A distributed dual proximal minimization algorithm for constraint-coupled optimization problems," *IEEE Control Syst. Lett.*, vol. 5, no. 1, pp. 259–264, Jan. 2021.

[12] X. Li, G. Feng, and L. Xie, "Distributed proximal algorithms for multiagent optimization with coupled inequality constraints," *IEEE Trans. Autom. Control*, vol. 66, no. 3, pp. 1223–1230, Mar. 2021.

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

[14] K. Srivastava and A. Nedić, "Distributed asynchronous constrained stochastic optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 4, pp. 772–790, Aug. 2011.

[15] Y. Tian, Y. Sun, and G. Scutari, "Achieving linear convergence in distributed asynchronous multiagent optimization," *IEEE Trans. Autom. Control*, vol. 65, no. 12, pp. 5264–5279, Dec. 2020.

[16] I. Notarnicola, R. Carli, and G. Notarstefano, "Distributed partitioned big-data optimization via asynchronous dual decomposition," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 4, pp. 1910–1919, Dec. 2018.

[17] N. Bastianello, R. Carli, L. Schenato, and M. Todescato, "Asynchronous distributed optimization over lossy networks via relaxed ADMM: Stability and linear convergence," *IEEE Trans. Autom. Control*, vol. 66, no. 6, pp. 2620–2635, Jun. 2021.

[18] Y. Lin, I. Shames, and D. Nesic, "Asynchronous distributed optimization via dual decomposition and block coordinate ascent," in *Proc. 58th IEEE Conf. Decis. Control (CDC)*, 2019, pp. 6380–6385.

[19] F. Farina, A. Garulli, A. Giannitrapani, and G. Notarstefano, "A distributed asynchronous method of multipliers for constrained nonconvex optimization," *Automatica*, vol. 103, pp. 243–253, May 2019.

[20] 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.

[21] T.-H. Chang, M.-C. Hong, W. 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, Jun. 2016.

[22] 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.

[23] S. H. Low and D. E. Lapsley, "Optimization flow control. I. Basic algorithm and convergence," *IEEE/ACM Trans. Netw.*, vol. 7, no. 6, pp. 861–874, Dec. 1999.

[24] R. Hannah and W. Yin, "On unbounded delays in asynchronous parallel fixed-point algorithms," *J. Sci. Comput.*, vol. 76, no. 5, pp. 299–326, 2018.

[25] R. T. Rockafellar, *Network Flows and Monotropic Optimization*. Belmont, MA, USA: Athena Sci., 1998.

[26] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1970.

[27] J. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*. Heidelberg, Germany: Springer, 2012.