Convergence Analysis of Dual Decomposition Algorithm in Distributed Optimization

Su, Yifan; Wang, Zhaojian; Cao, Ming; Jia, Mengshuo; Liu, Feng

Published in:
IEEE Transactions on Automatic Control

DOI:
10.1109/TAC.2022.3213608

Convergence Analysis of Dual Decomposition Algorithm in Distributed Optimization: Asynchrony and Inexactness

Yifan Su, Zhaojian Wang, *Member, IEEE*, Ming Cao, *Fellow, IEEE*, Mengshuo Jia, *Member, IEEE*, and Feng Liu, *Senior Member, IEEE*

***Abstract*—Dual decomposition is widely utilized in the distributed optimization of multiagent systems. In practice, the dual decomposition algorithm is desired to admit an asynchronous implementation due to imperfect communication, such as time delay and packet drop. In addition, computational errors also exist when the individual agents solve their own subproblems. In this article, we analyze the convergence of the dual decomposition algorithm in the distributed optimization when both the communication asynchrony and the subproblem solution inexactness exist. We find that the interaction between asynchrony and inexactness slows down the convergence rate from $\mathcal{O}(1/k)$ to $\mathcal{O}(1/\sqrt{k})$. Specifically, with a constant step size, the value of the objective function converges to a neighborhood of the optimal value, and the solution converges to a neighborhood of the optimal solution. Moreover, the violation of the constraints diminishes in $\mathcal{O}(1/\sqrt{k})$. Our result generalizes and unifies the existing ones that only consider either asynchrony or inexactness. Finally, numerical simulations validate the theoretical results.**

***Index Terms*—Asynchronous algorithm, distributed optimization, dual decomposition, inexact algorithm, multiagent system (MAS).**

Manuscript received 1 April 2022; revised 11 August 2022; accepted 22 September 2022. Date of publication 11 October 2022; date of current version 28 July 2023. This work was supported by the Joint Research Fund in Smart Grid under Grant U1966601 under cooperative agreement between the National Natural Science Foundation of China (NSFC) and State Grid Corporation of China. Recommended by Associate Editor T. Faulwasser. *(Corresponding author: Feng Liu.)*

Yifan Su and Feng Liu are with the State Key Laboratory of Power System, Department of Electrical Engineering, Tsinghua University, Beijing 100084, China (e-mail: deansu97@foxmail.com; lfeng@tsinghua.edu.cn).

Zhaojian Wang is with the Key Laboratory of System Control, and Information Processing, Ministry of Education of China, Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wangzhaojian@sjtu.edu.cn).

Ming Cao is with the Faculty of Science, and Engineering, University of Groningen, 9747 AG Groningen, The Netherlands (e-mail: ming.cao@ieee.org).

Mengshuo Jia is with the Power Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland (e-mail: jia@eeh.ee.ethz.ch).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TAC.2022.3213608.

Digital Object Identifier 10.1109/TAC.2022.3213608

I. Introduction

Dual decomposition is widely utilized in solving distributed optimization problems of multiagent systems (MASs), such as communication networks [1], [2], [3], computer vision [4], [5], and power systems [6], [7], [8]. A dual decomposition algorithm usually involves two phases in each iteration: a coordinator updates the dual variables (Lagrangian multipliers) and individual agents solve their subproblems locally [9], [10], [11]. Then, the dual variables and the subproblem solutions are exchanged between the coordinator and the agents via the communication network to execute the next iteration. During the iterative process, the communication asynchrony and the subproblem solution inexactness may undermine the convergence of the algorithm. In the literature, these two issues are addressed separately even though they always coexist. In this regard, this article analyzes the convergence of the dual-decomposition-based distributed optimization (DD-DO) algorithm considering asynchrony and inexactness simultaneously.

A. Related Works

Dual decomposition is commonly regarded as a first-order (sub)gradient ascent method with respect to the dual problem. Under ideal conditions, the convergence of dual decomposition has been thoroughly studied. For a diminishing step size α_k satisfying $\sum_{k=0}^{\infty} \alpha_k \to \infty, \sum_{k=0}^{\infty} \alpha_k^2 < \infty$, the gradient and subgradient algorithms converge to the optimal value [12, Prop. 8.2.4]. For a constant step size, the gradient algorithm still converges to the optimal value [13, Prop. 3.4], while the subgradient method converges to a neighborhood of the optimal value [12, Prop. 8.2.2]. However, if the communication asynchrony and the subproblem solution inexactness are considered, the convergence of dual decomposition requires further studies. Next, we give a short review of these two aspects.

1) Communication Asynchrony: In practice, the implementation of the DD-DO algorithm usually suffers from asynchrony due to packet drop, time delay in communications, nonidentical computational rates, etc. In this situation, the synchronous DD-DO algorithm may cause longer idle time since the coordinator and the agents have to wait for the latest information from their neighbors in order to execute the next iteration [14]. To circumvent this problem, the asynchronous DD-DO algorithm is proposed by updating the dual variables and solving the subproblems immediately using the previously stored information, if the latest information happens to be unavailable. In [15], the convergence of the asynchronous DD-DO algorithm is studied, showing that the algorithm converges to the optimal solution under a bounded time delay. Magnússon et al. [16] develop a fully distributed dual decomposition-based method in a radial communication network and proves the convergence of the proposed algorithm, considering the asynchrony under a novel communication structure. Notarnicola et al. [17] propose an asynchronous partitioned dual decomposition algorithm for fully distributed optimization over peer-to-peer networks, where the algorithm converges in probability with the independent and identically distributed (i.i.d.) delays.

Besides the dual decomposition method, the communication asynchrony has also been considered in other distributed optimization algorithms, e.g., the alternating direction method of multipliers (ADMM) [18], [19], [20], the consensus algorithm [21], the primal-dual gradient method [22], [23], and the distributed Newton method [24]. Chang et al. [20] prove that the asynchronous distributed ADMM has a linear convergence rate for the consensus optimization problem. In [23], the dynamic of the proposed partial primal-dual gradient method converges to the equilibrium point exponentially even with a nonsmooth objective function. Mansoori and Wei [24] develop an asynchronous Newton approach, which converges with a global linear rate and a local superlinear rate in expectation.

2) Subproblem Solution Inexactness: In the dual decomposition algorithm, the subproblem solutions of the individual agents will inevitably deviate from the optimal solutions, depending on the preset error tolerances of solvers, the types of problems, and the accuracy of parameters. The inexactness issue may lead to considerable error or even divergence of the algorithm due to the accumulation of subproblem errors during iterations. To address the issue, the averaging scheme is suggested in recent years by taking the average of the decision variables over the iteration horizon. Devolder et al. [25] utilize the inexact oracle to study the dual decomposition algorithm. In [26], the inexact dual decomposition is proved to have an $\mathcal{O}(1/k)$ rate of convergence. An inexact DD-DO algorithm to solve a Laplacian consensus problem is studied in [27], where the deviation of solution diminishes exponentially considering the exponentially decayed error. In [28], the iteration complexity of the inexact augmented Lagrangian method for constrained convex programming is studied, where the convergence rate is $\mathcal{O}(1/k)$ even with a nonsmooth objective function. Mehyar et al. [29] analyze the convergence of dual decomposition with inexact updating of dual variables, where the choice of step size is presented to help the algorithm enter an attraction region in finite steps.

The inexactness has also been considered in other distributed optimization algorithms, e.g., the ADMM [30], [31], the primal gradient algorithm [32], the primal-dual gradient method [33], the proximal gradient method [34], [35], [36], and the distributed Newton method [37], [38]. Chang et al. [30] prove that the ADMM with the proximal subproblems converges linearly under certain convexity assumptions, while Svaiter [31] show that the ADMM with the σ-approximate solutions of subproblems converges in $\mathcal{O}(1/k)$. In [32], a novel primal-dual gradient projection method is proposed with an inexact computation of the projection, which has an $\mathcal{O}(1/k)$ rate of ergodic convergence. Wei et al. [38] develop an inexact distributed Newton method considering the errors of the Newton direction and the step size calculation, which achieves a local quadratic convergence rate.

B. Contributions

It should be noted that the works mentioned previously only consider either the asynchrony [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24] or the inexactness [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], although they always coexist in distributed optimization methods. It is still unknown how the distributed optimization methods perform in the asynchronous and inexact case, which hinders these algorithms from being applied to real systems. In this article, we analyze the convergence of the DD-DO algorithm considering asynchrony and inexactness simultaneously. Specifically, under mild conditions, we prove the convergence of the asynchronous and inexact DD-DO algorithm, whose characteristics include the following.

1) *Sublinear convergence rate:* Under ideal conditions, the DD-DO algorithm converges in $\mathcal{O}(1/k)$ as given in [39] and [40]. We prove that the interaction of asynchrony and inexactness slows down the convergence rate to $\mathcal{O}(1/\sqrt{k})$. We also show that a *constant* step size is enough to obtain the aforementioned convergence performance, which is more applicable in practice than using a diminishing step size.
2) *Suboptimality and feasibility:* We show that the value of the primal variable converges to a neighborhood of the optimal solution to the primal problem, while the value of the primal (dual) objective converges to a neighborhood of the optimal value of the primal (dual) problem, both in $\mathcal{O}(1/\sqrt{k})$. We also give upper bounds of these neighborhoods, which are positively correlated to the degrees of asynchrony and inexactness. Moreover, the violation of the constraints diminishes in an $\mathcal{O}(1/\sqrt{k})$ rate of convergence, even though the subproblem solutions are inexact in each iteration.
3) *Generality:* Our convergence results generalize and unify the existing works that only consider asynchrony [15] or inexactness [26]. By simply setting the inexactness or asynchrony parameter as zero, our result reduces to that given in [15] or [26], respectively. Our work also first gives an $\mathcal{O}(1/k)$ rate of convergence of the *asynchronous* DD-DO algorithm, which, to the best of our knowledge, has not been presented in the existing literature [6], [15], [16].

C. Organization

The rest of this article is organized as follows. Section II formulates the optimal MAS operation problem and solves it by the synchronous and exact DD-DO algorithm. In Section III, the asynchrony and inexactness are formulated and analyzed in the DD-DO algorithm. Section IV proves the convergence of the asynchronous and inexact algorithm. Section V gives numerical results, and finally, Section VI concludes this article.

Notations: In this article, we use $\mathbb{R}^n$ ($\mathbb{R}^n_+$) to denote the n-dimensional (nonnegative) Euclidean space. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, we denote the inner product by $< \boldsymbol{x}, \boldsymbol{y} >= \boldsymbol{x}^T \boldsymbol{y}$, and the 2-norm by $\|\boldsymbol{x}\| = \sqrt{< \boldsymbol{x}, \boldsymbol{x} >}$. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, x_i stands for the ith entry. $\mathrm{col}\{\boldsymbol{x}_i\}_{i \in \mathcal{I}}$ stacks the vectors $\boldsymbol{x}_i$ as a new column vector in the order of the index set $\mathcal{I}$. For a vector $\boldsymbol{x} \in \mathbb{R}^n$, $[\boldsymbol{x}]^+ := \mathrm{col}\{\max\{x_i, 0\}\}_{i \in \{1,2,\ldots,n\}}$ stands for the projection onto $\mathbb{R}^n_+$. For $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, $\boldsymbol{x} \le \boldsymbol{y}$ (or $\boldsymbol{y} \ge \boldsymbol{x}$) are meant to be component wise, i.e., $x_i \le y_i \, \forall i = 1, 2, \ldots, n$. For a matrix $A \in \mathbb{R}^{m \times n}$, $\|A\|$ stands for the 2-norm. For a set Ω, $|\Omega|$ stands for its cardinality.

II. DD-DO Algorithm in MASs

In this section, we formulate the optimal operation problem of the MAS, and solve it by the conventional DD-DO algorithm.

A. Primal Problem

We focus on the large-scale MAS with a set of agents denoted by $\mathcal{N}$. Each agent $i \in \mathcal{N}$ can make its decision $\boldsymbol{x}_i \in \mathbb{R}^{n_i}$ in a local feasible region $\mathcal{X}_i$, and meanwhile causes a cost $f_i(\boldsymbol{x}_i)$. Our objective is to minimize the aggregate cost with restrictions on global constraints and local feasible regions, i.e., solve the following optimization problem, called the primal problem

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) = \sum_{i \in \mathcal{N}} f_i(\boldsymbol{x}_i) \tag{1a}$$

$$\text{s.t.}\ \boldsymbol{x}_i \in \mathcal{X}_i, \ \forall i \in \mathcal{N} \tag{1b}$$

$$A\boldsymbol{x} \le \boldsymbol{b} \tag{1c}$$

where $\boldsymbol{x} = \mathrm{col}\{\boldsymbol{x}_i\}_{i \in \mathcal{N}} \in \mathbb{R}^n$ is the aggregate decision vector; $\mathcal{X} = \Pi_{i \in \mathcal{N}} \mathcal{X}_i$ is the aggregate feasible region; (1b) represents the local feasible regions of the agents; (1c) is the global constraints. The matrix $A \in \mathbb{R}^{m \times n}$ and the vector $\boldsymbol{b} \in \mathbb{R}^m$ are constants. Let $A_i \in \mathbb{R}^{m \times n_i}$ denote the ith sliced block of $A = (A_1, A_2, \ldots, A_{|\mathcal{N}|})$. Then, (1c) can be replaced by

$$\sum_{i \in \mathcal{N}} A_i \boldsymbol{x}_i \le \boldsymbol{b}.$$

Throughout this article, we make the following assumptions on the primal problem.

Assumption A1:

1) The cost function $f_i(\cdot)$ is c_i strongly convex and differentiable over $\mathcal{X}_i$. Hence, the objective function $F(\cdot)$ is strongly convex with $c_F = \min_{i \in \mathcal{N}} c_i$ and differentiable over $\mathcal{X}$.
2) The feasible region $\mathcal{X}_i$ is a nonempty, compact, and convex set. Hence, $\mathcal{X}$ is also nonempty, compact, and convex.
3) There exists a strictly feasible interior point in $\mathcal{X}$ that satisfies (1c).

Under Assumption A1, problem (1) enjoys a *unique* primal optimal solution denoted by $\boldsymbol{x}^* \in \mathcal{X}$. Denote by

$$F^* = F(\boldsymbol{x}^*)$$

the optimal value of (1).

B. Dual Problem

Define the Lagrangian of (1)

$$\mathcal{L}(\boldsymbol{x}; \boldsymbol{\lambda}) = F(\boldsymbol{x}) + \langle \boldsymbol{\lambda}, A\boldsymbol{x} - \boldsymbol{b} \rangle$$

where $\boldsymbol{\lambda} \in \mathbb{R}^m_+$ is the Lagrangian multiplier of (1c). Throughout this article, we call $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ the primal and dual variables, respectively.

Then, the dual problem of (1) is given as

$$\max_{\boldsymbol{\lambda} \ge 0} \min_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}; \boldsymbol{\lambda}). \tag{2}$$

From the strong convexity of $F(\boldsymbol{x})$, given $\boldsymbol{\lambda} \ge 0$, the Lagrangian $\mathcal{L}(\boldsymbol{x}; \boldsymbol{\lambda})$ is also c_F strongly convex in $\boldsymbol{x}$. Denote by $D(\boldsymbol{\lambda})$ and $\boldsymbol{x}(\boldsymbol{\lambda})$ the optimal value and the *unique* optimal solution to the inner minimization problem of (2), i.e.,

$$D(\boldsymbol{\lambda}) = \min_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}; \boldsymbol{\lambda}) \tag{3}$$

$$\boldsymbol{x}(\boldsymbol{\lambda}) = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \mathcal{L}(\boldsymbol{x}; \boldsymbol{\lambda}). \tag{4}$$

Let D^* denote the optimal value of (2). Under Assumption A1, the Slater condition of the problem (1) holds from [42, Sec. 5.2.3]. It indicates that the duality gap is zero, i.e., $F^* = D^*$. Define the optimal set of (2) as $\Lambda^* := \{\boldsymbol{\lambda} \in \mathbb{R}^m_+ \,|\, D(\boldsymbol{\lambda}) = D^*\}$ and denote by $\boldsymbol{\lambda}^* \in \Lambda^*$ an arbitrary optimal solution to (2). It follows that

$$D^* = D(\boldsymbol{\lambda}^*) = \mathcal{L}(\boldsymbol{x}(\boldsymbol{\lambda}^*); \boldsymbol{\lambda}^*), \ \forall \boldsymbol{\lambda}^* \in \Lambda^*.$$

From [42], the Slater condition guarantees that Λ^* is nonempty and bounded. According to the dual theory [42, Sec. 5.1.2], the dual function $D(\boldsymbol{\lambda})$ is concave. Thus, Λ^* is convex from [42, Sec. 4.2.1].

Remark 1 (Multiple optimal solution $\boldsymbol{\lambda}^*$): Although the optimal solution to (2) may not be unique, i.e., $|\Lambda^*| \ge 1$, our main results hold for any $\boldsymbol{\lambda}^* \in \Lambda^*$. To tighten our convergence results, we can pick the *unique* optimal solution with the minimal 2-norm defined as

$$\boldsymbol{\lambda}^{**} := \arg\min_{\boldsymbol{\lambda}^* \in \Lambda^*} \|\boldsymbol{\lambda}^*\|. \tag{5}$$

C. Dual Decomposition

The basic idea of dual decomposition is to solve the dual problem in a distributed manner. Note that the Lagrangian is separable, i.e.,

$$\mathcal{L}(\boldsymbol{x}; \boldsymbol{\lambda}) = \sum_{i \in \mathcal{N}} \mathcal{L}_i(\boldsymbol{x}_i; \boldsymbol{\lambda}) - < \boldsymbol{\lambda}, \boldsymbol{b} >$$

$$\mathcal{L}_i(\boldsymbol{x}_i; \boldsymbol{\lambda}) = f_i(\boldsymbol{x}_i) + < A_i^T \boldsymbol{\lambda}, \boldsymbol{x}_i >.$$

From the strong convexity of $f_i(\boldsymbol{x}_i)$, given any $\boldsymbol{\lambda}$, $\mathcal{L}_i(\boldsymbol{x}_i; \boldsymbol{\lambda})$ is also c_i-strongly convex in $\boldsymbol{x}_i$, and hence, is minimized over $\mathcal{X}_i$ at a *unique* point. For $i \in \mathcal{N}$, we define

$$D_i(\boldsymbol{\lambda}) = \min_{\boldsymbol{x}_i \in \mathcal{X}_i} \mathcal{L}_i(\boldsymbol{x}_i; \boldsymbol{\lambda}) \tag{6}$$

$$\boldsymbol{x}_i(\boldsymbol{\lambda}) = \arg\min_{\boldsymbol{x}_i \in \mathcal{X}_i} \mathcal{L}_i(\boldsymbol{x}_i; \boldsymbol{\lambda}). \tag{7}$$

Algorithm 1: Synchronous and Exact DD-DO Algorithm.

Input: Accuracy tolerance $\epsilon > 0$, step size $\alpha > 0$, initial dual variable $\boldsymbol{\lambda}^0 \geq 0$, and iteration index $k = 0$.

Output: Optimal solution $\boldsymbol{x}^*$.

S1 (Solving subproblems): Agent i attains the optimal solution, $\boldsymbol{x}_i^k$, to the following subproblem:

$$\min_{\boldsymbol{x}_i \in \mathcal{X}_i} \mathcal{L}_i \left(\boldsymbol{x}_i; \boldsymbol{\lambda}^k\right) \tag{8}$$

i.e., $\boldsymbol{x}_i^k = \boldsymbol{x}_i(\boldsymbol{\lambda}^k)$.

S2 (Updating dual variable): The central coordinator updates the dual variable as

$$\boldsymbol{\lambda}^{k+1} = \left[\boldsymbol{\lambda}^k + \alpha \left(A\boldsymbol{x}^k - \boldsymbol{b}\right)\right]^+ \tag{9}$$

where $\boldsymbol{x}^k = \mathrm{col}\{\boldsymbol{x}_i^k\}_{i\in\mathcal{N}}$.

S3: If $\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\| \leq \epsilon$, $\boldsymbol{x}^k$ is recognized as the optimal solution and the algorithm terminates. Otherwise, set $k = k + 1$ and go to **S1**.

Hence, we have

$$D(\boldsymbol{\lambda}) = \sum_{i\in\mathcal{N}} D_i(\boldsymbol{\lambda}) - <\boldsymbol{\lambda}, \boldsymbol{b}>$$

$$\boldsymbol{x}(\boldsymbol{\lambda}) = \mathrm{col}\{\boldsymbol{x}_i(\boldsymbol{\lambda})\}_{i\in\mathcal{N}}.$$

The entire distributed algorithm is shown in Algorithm 1. According to [13, Prop. 3.4], [15], the dual variable ultimately converges to some optimal point $\boldsymbol{\lambda}^* \in \Lambda^*$, and meanwhile, agents attain the optimal solution $\boldsymbol{x}^* = \boldsymbol{x}(\boldsymbol{\lambda}^*) = \mathrm{col}\{\boldsymbol{x}_i(\boldsymbol{\lambda}^*)\}_{i\in\mathcal{N}}$ and the optimal value $D^* = D(\boldsymbol{\lambda}^*) = \sum_{i\in\mathcal{N}} D_i(\boldsymbol{\lambda}^*) - <\boldsymbol{\lambda}^*, \boldsymbol{b}>$ by solving the subproblems (8).

Remark 2 (Distributed Implementation): The dual decomposition algorithm can be partially distributed or fully distributed up to the structure of the MAS. Dual decomposition is commonly implemented in a partially distributed manner as Algorithm 1, where the dual variable is computed by a central coordinator. The algorithm can also be fully distributed depending on the particular sparse communication network, for instance, Internet networks [15] and radial distribution grids [16]. Without a central coordinator, each agent communicates with its neighbors and updates the dual variable locally. Both the partially and fully distributed algorithms share the same iterative procedure as Algorithm 1. Hence, we follow the partially distributed framework throughout the rest of this article.

For later theoretical analyses, we turn to studying the basic properties of the dual decomposition algorithm. Under Assumption A1, we have the following lemma of the Lipschitz continuity of $\boldsymbol{x}_i(\boldsymbol{\lambda})$.

Lemma 1: Suppose Assumption A1 holds. $\boldsymbol{x}_i(\boldsymbol{\lambda})$ is $\|A_i\|/c_i$ - Lipschitz continuous in $\boldsymbol{\lambda} \in \mathbb{R}_+^m$.

The proof of Lemma 1 can be found in Appendix A. Then, we have the following corollary of the existence and Lipschitz continuity of $\nabla D(\boldsymbol{\lambda})$.

Corollary 2: Supposing Assumption A1 holds, we have the following properties.

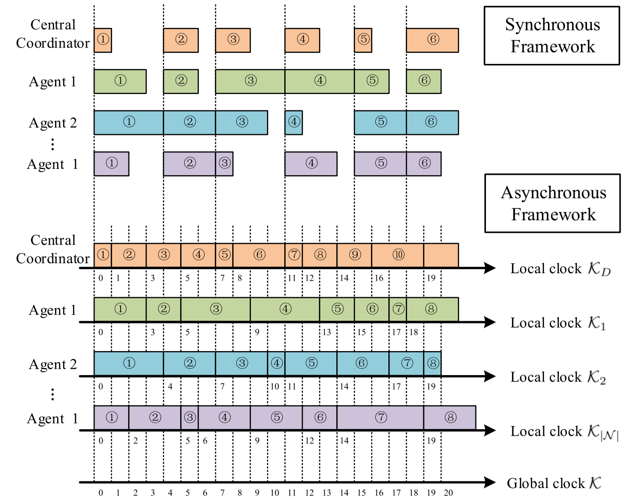


Fig. 1. Algorithm in the synchronous and asynchronous cases.

1) $D_i(\boldsymbol{\lambda})$ is differentiable and the gradient is defined as

$$\nabla D_i(\boldsymbol{\lambda}) = A_i \boldsymbol{x}_i(\boldsymbol{\lambda}). \tag{10}$$

Moreover, $\nabla D_i(\boldsymbol{\lambda})$ is L_i-Lipschitz continuous with

$$L_i = \frac{\|A_i\|^2}{c_i}.$$

2) $D(\boldsymbol{\lambda})$ is differentiable and the gradient is defined as

$$\nabla D(\boldsymbol{\lambda}) = A\boldsymbol{x}(\boldsymbol{\lambda}) - \boldsymbol{b} = \sum_{i\in\mathcal{N}} \nabla D_i(\boldsymbol{\lambda}) - \boldsymbol{b}. \tag{11}$$

Moreover, $\nabla D(\boldsymbol{\lambda})$ is L_D-Lipschitz continuous with

$$L_D = \sum_{i\in\mathcal{N}} L_i = \sum_{i\in\mathcal{N}} \frac{\|A_i\|^2}{c_i}.$$

The proof of Corollary 2 can be found in Appendix B.

III. Asynchronous and Inexact DD-DO Algorithm

In this section, we present the asynchronous and inexact DD-DO algorithm.

First of all, We describe the communication asynchrony following [6], [14], [15]. The local clock $\mathcal{K}_i$ is the set of time instants when agent i takes action. At time instant $k \in \mathcal{K}_i$, the agent i solves the subproblem with the previously stored dual variable if the latest is not received; otherwise, it keeps the solution unchanged. The local clock $\mathcal{K}_D$ is the set of time instants when the central coordinator takes action. At time instant $k \in \mathcal{K}_D$, the central coordinator updates the dual variable with the previously stored subproblem solution if the latest is not received; otherwise, it keeps the dual variable unchanged. Define the global clock as $\mathcal{K} = \mathcal{K}_D \cup \mathcal{K}_1 \cup \ldots \cup \mathcal{K}_{|\mathcal{N}|}$. Fig. 1 shows the difference of the algorithm in the synchronous and asynchronous cases.

The asynchronous and inexact DD-DO algorithm is presented in Algorithm 2. Denote by $\{\widetilde{\boldsymbol{x}}^k\}$ and $\{\boldsymbol{\lambda}^k\}$, respectively, the primal and dual sequences generated by the algorithm. Here, $\widetilde{\boldsymbol{x}}$ is an inexact version of $\boldsymbol{x}$. Note that $\widetilde{\boldsymbol{x}}^k = \mathrm{col}\{\widetilde{\boldsymbol{x}}_i^k\}_{i\in\mathcal{N}}$.

Algorithm 2: Asynchronous and Inexact DD-DO Algorithm.

Input: Accuracy tolerance $\epsilon > 0$, step size $\alpha > 0$, initial dual variable $\boldsymbol{\lambda}^0 \geq 0$, and iteration index $k = 0$.
Output: Suboptimal solution $\widetilde{\boldsymbol{x}}^*$.
S1 (Solving subproblems): If $k \in \mathcal{K}_i$, agent i solves its subproblem (12) and attains an ε_i-suboptimal solution $\widetilde{\boldsymbol{x}}_i^k$. Otherwise, set $\widetilde{\boldsymbol{x}}_i^k = \widetilde{\boldsymbol{x}}_i^{k-1}$.
S2 (Updating dual variable): If $k \in \mathcal{K}_D$, the central coordinator updates the dual variable $\boldsymbol{\lambda}^{k+1}$ by (14). Otherwise, set $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k$.
S3: If $k \in \mathcal{K}_D$ and $\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\| \leq \epsilon$, the algorithm is regarded to converge and the iteration terminates. Otherwise, set $k = k + 1$ and go to **S1**.

If $k \in \mathcal{K}_i$, the agent i will compute the subproblem and attain $\widetilde{\boldsymbol{x}}_i^k = \widetilde{\boldsymbol{x}}_i(\widehat{\boldsymbol{\lambda}}^{k,i})$, where $\widetilde{\boldsymbol{x}}_i(\widehat{\boldsymbol{\lambda}}^{k,i})$ is an inexact solution to the following subproblem:

$$\min_{\boldsymbol{x}_i \in \mathcal{X}_i} f_i(\boldsymbol{x}_i) + < A_i^T \widehat{\boldsymbol{\lambda}}^{k,i}, \boldsymbol{x}_i > \tag{12}$$

where $\widehat{\boldsymbol{\lambda}}^{k,i}$ is the previously stored dual variable by the agent i. $\widehat{\boldsymbol{\lambda}}^{k,i}$ is defined as

$$\widehat{\boldsymbol{\lambda}}^{k,i} = \boldsymbol{\lambda}^{k-\delta_{di}^k} \tag{13}$$

where $\delta_{di}^k \geq 0$ is the time delay associated with k.

If $k \notin \mathcal{K}_i$, the solution is unchanged, i.e., $\widetilde{\boldsymbol{x}}_i^k = \widetilde{\boldsymbol{x}}_i^{k-1}$.

Denoted by $\boldsymbol{x}_i^k(\widehat{\boldsymbol{\lambda}}^{k,i})$ is the optimal solution to (12). The error of the inexact solution $\widetilde{\boldsymbol{x}}_i^k(\widehat{\boldsymbol{\lambda}}^{k,i})$ from $\boldsymbol{x}_i^k(\widehat{\boldsymbol{\lambda}}^{k,i})$ will be defined later.

If $k \in \mathcal{K}_D$, the central coordinator will update the dual variable $\boldsymbol{\lambda}^{k+1}$ as

$$\boldsymbol{\lambda}^{k+1} = \left[\boldsymbol{\lambda}^k + \alpha \boldsymbol{\nu}^k\right]^+ \tag{14}$$

where $\boldsymbol{\nu}^k$ is the estimated gradient, differently from the real gradient $\nabla D(\boldsymbol{\lambda}^k)$. $\boldsymbol{\nu}^k$ is defined as

$$\boldsymbol{\nu}^k = A\widehat{\boldsymbol{x}}^k - \boldsymbol{b} \tag{15}$$

where $\widehat{\boldsymbol{x}}^k := \mathrm{col}\{\widehat{\boldsymbol{x}}_i^k\}_{i\in\mathcal{N}}$ is the previously stored primal variable. $\widehat{\boldsymbol{x}}_i^k$ is defined as

$$\widehat{\boldsymbol{x}}_i^k = \widetilde{\boldsymbol{x}}_i^{k-\delta_{pi}^k} \tag{16}$$

where $\delta_{pi}^k \geq 0$ is the time delay associated with k.

If $k \notin \mathcal{K}_D$, the dual variable is unchanged, i.e., $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k$ and $\boldsymbol{\nu}^k = \boldsymbol{0}$.

Then, we make the following assumption on the communication asynchrony.

Assumption A2: There exists an upper bound $k_0 \geq 0$ on time delays such that

$$0 \leq \delta_{di}^k \leq k_0 \ \forall i \in \mathcal{N}, k \in \mathcal{K} \tag{17a}$$

$$0 \leq \delta_{pi}^k \leq k_0 \ \forall i \in \mathcal{N}, k \in \mathcal{K}. \tag{17b}$$

From (13) and (16), we have

$$\widehat{\boldsymbol{x}}_i^k = \widetilde{\boldsymbol{x}}_i^{k-\delta_{pi}^k} = \widetilde{\boldsymbol{x}}_i\left(\widehat{\boldsymbol{\lambda}}^{k-\delta_{pi}^k,i}\right) = \widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}^{k-\delta_{pi}^k-\delta_{di}^{k-\delta_{pi}^k}}\right).$$

For simplicity, define $\delta_i^k := \delta_{pi}^k - \delta_{di}^{k-\delta_{pi}^k}$. Under Assumption A2, it follows that

$$\widehat{\boldsymbol{x}}_i^k = \widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right), \ 0 \leq \delta_i^k \leq 2k_0. \tag{18}$$

Denote by $\widetilde{D}_i(\boldsymbol{\lambda})$ the inexact value of the objective function with respect to $\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda})$, i.e.,

$$\widetilde{D}_i(\boldsymbol{\lambda}) = \mathcal{L}_i(\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda}); \boldsymbol{\lambda}) = f_i(\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda})) + < \boldsymbol{\lambda}, A_i\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda}) > . \tag{19}$$

Then, we have the following assumption on the subproblem solution inexactness.

Assumption A3: There exists an error bound $\varepsilon_i \geq 0$ such that given any $\boldsymbol{\lambda} \in \mathbb{R}_+^m$, the inexact solution $\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda})$ is ε_i-suboptimal, i.e.,

1) feasible, i.e., $\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda}) \in \mathcal{X}_i$;
2) suboptimal, i.e., $\widetilde{D}_i(\boldsymbol{\lambda}) - D_i(\boldsymbol{\lambda}) \leq \varepsilon_i$, in other words, $\mathcal{L}_i(\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda}); \boldsymbol{\lambda}) - \mathcal{L}_i(\boldsymbol{x}_i(\boldsymbol{\lambda}); \boldsymbol{\lambda}) \leq \varepsilon_i$.

Under Assumption A3, we have the following lemma about the subproblem solution inexactness.

Lemma 3: Suppose Assumptions A1 and A3 hold. The distance between the optimal and inexact solutions is bounded by

$$\|\boldsymbol{x}_i(\boldsymbol{\lambda}) - \widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda})\|^2 \leq \frac{2\varepsilon_i}{c_i}.$$

The proof of Lemma 3 can be found in Appendix C.

Define

$$\varepsilon_D := \sum_{i\in\mathcal{N}} \varepsilon_i \geq 0. \tag{20}$$

Then, k_0 and ε_D can be regarded as the indexes of asynchrony and inexactness, respectively. $k_0 = 0$ ($\varepsilon_D = 0$) indicates a synchronous (an exact) case. Otherwise, it is asynchronous (inexact).

IV. Main Result

In this section, we analyze the convergence of the asynchronous and inexact DD-DO algorithm.

Define the dual deviation

$$\boldsymbol{\sigma}^k := \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k$$

and the sum-of-square of dual deviations

$$S^k := \sum_{\kappa=0}^{k} \|\boldsymbol{\sigma}^\kappa\|^2 .$$

Besides, to avoid trivial discussions, we set $\boldsymbol{\sigma}^k = \boldsymbol{0}, \ k < 0$.

A. Bound on the Sum-of-Square of Dual Deviations

We turn to proving that the sequence $\{\sqrt{S^k}\}$ generated by Algorithm 2 increases not faster than $\mathcal{O}(\sqrt{k})$, starting with the following two lemmas.

Lemma 4: In Algorithm 2 $\forall k \in \mathcal{K}_D$

$$< \boldsymbol{\nu}^k, \boldsymbol{\sigma}^k > \geq \left\|\boldsymbol{\sigma}^k\right\|^2 / \alpha. \tag{21}$$

Proof: Recalling $\boldsymbol{\lambda}^{k+1} = [\boldsymbol{\lambda}^k + \alpha\boldsymbol{\nu}^k]^+$, we have by the projection theorem [44, Prop. 2.1.3]

$$< \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k - \alpha\boldsymbol{\nu}^k, \boldsymbol{\lambda} - \boldsymbol{\lambda}^{k+1} > \geq 0 \ \forall \boldsymbol{\lambda} \geq 0.$$

By replacing $\boldsymbol{\lambda}$ with $\boldsymbol{\lambda}^k$ and recalling $\boldsymbol{\sigma}^k = \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k$, we obtain (21) directly. □

Lemma 5: Suppose Assumptions A1–A3 hold. In Algorithm 2, we have for $k \in \mathcal{K}_D$

$$\left\|\nabla D\left(\boldsymbol{\lambda}^k\right) - \boldsymbol{\nu}^k\right\| \leq L_D \sum_{\kappa=k-2k_0}^{k-1} \|\boldsymbol{\sigma}^\kappa\| + \sqrt{2L_D\varepsilon_D}. \tag{22}$$

Proof: From the definitions of $\nabla D(\boldsymbol{\lambda}^k)$ and $\boldsymbol{\nu}^k$, we have for $k \in \mathcal{K}_D$

$$\begin{aligned}
&\left\|\nabla D\left(\boldsymbol{\lambda}^k\right) - \boldsymbol{\nu}^k\right\| \\
&= \left\|A\boldsymbol{x}\left(\boldsymbol{\lambda}^k\right) - A\widehat{\boldsymbol{x}}^k\right\| \\
&\overset{a)}{\leq} \sum_{i\in\mathcal{N}} \|A_i\| \left\|\boldsymbol{x}_i\left(\boldsymbol{\lambda}^k\right) - \widehat{\boldsymbol{x}}_i^k\right\| \\
&= \sum_{i\in\mathcal{N}} \|A_i\| \left\|\boldsymbol{x}_i\left(\boldsymbol{\lambda}^k\right) - \widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)\right\| \\
&\overset{b)}{\leq} \sum_{i\in\mathcal{N}} \|A_i\| \left\|\boldsymbol{x}_i\left(\boldsymbol{\lambda}^k\right) - \boldsymbol{x}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)\right\| \\
&\quad + \sum_{i\in\mathcal{N}} \|A_i\| \left\|\boldsymbol{x}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right) - \widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)\right\| \\
&\overset{c)}{\leq} \sum_{i\in\mathcal{N}} \|A_i\| \frac{\|A_i\|}{c_i} \left\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-\delta_i^k}\right\| + \sum_{i\in\mathcal{N}} \|A_i\| \sqrt{\frac{2\varepsilon_i}{c_i}} \\
&\overset{d)}{\leq} \sum_{i\in\mathcal{N}} L_i \sum_{\kappa=k-\delta_i^k}^{k-1} \left\|\boldsymbol{\lambda}^{\kappa+1} - \boldsymbol{\lambda}^\kappa\right\| + \sum_{i\in\mathcal{N}} \sqrt{2L_i\varepsilon_i} \\
&= \sum_{i\in\mathcal{N}} L_i \sum_{\kappa=k-\delta_i^k}^{k-1} \|\boldsymbol{\sigma}^\kappa\| + \sqrt{2}\sum_{i\in\mathcal{N}} \sqrt{L_i}\sqrt{\varepsilon_i} \\
&\overset{e)}{\leq} \sum_{i\in\mathcal{N}} L_i \sum_{\kappa=k-2k_0}^{k-1} \|\boldsymbol{\sigma}^\kappa\| + \sqrt{2}\sqrt{\left(\sum_{i\in\mathcal{N}} L_i\right)\left(\sum_{i\in\mathcal{N}} \varepsilon_i\right)} \\
&= L_D \sum_{\kappa=k-2k_0}^{k-1} \|\boldsymbol{\sigma}^\kappa\| + \sqrt{2L_D\varepsilon_D}
\end{aligned}$$

where a) follows from the triangle inequality and $\|A\boldsymbol{b}\| \leq \|A\|\|\boldsymbol{b}\| \, \forall A \in \mathbb{R}^{m\times n}, \boldsymbol{b} \in \mathbb{R}^n$, b) holds by adding and subtracting $\boldsymbol{x}_i(\boldsymbol{\lambda}^{k-\delta_i^k})$ and using the triangle inequality, c) is due to Lemmas 1 and 3, d) follows from the triangle inequality and the definition of L_i in Corollary 2, and e) holds from $\delta_i^k \leq 2k_0$ and the Cauchy $-$ Schwarz inequality. □

Then, we obtain the upper bound on $\sqrt{S^k}$, if the step size is sufficiently small.

Theorem 6: Suppose Assumptions A1–A3 hold. In Algorithm 2, provided that the step size satisfies

$$0 < \alpha < \frac{1}{(2k_0 + 1/2)\, L_D} \tag{23}$$

then it follows that

$$\sqrt{S^k} \leq \frac{\sqrt{2L_D\varepsilon_D}}{\gamma_\alpha}\sqrt{k+1} + \sqrt{\frac{D^* - D^0}{\gamma_\alpha}} \tag{24}$$

where γ_α is a positive constant defined as

$$\gamma_\alpha := \frac{1}{\alpha} - \left(2k_0 + \frac{1}{2}\right) L_D$$

and D^0 is defined as $D^0 := D(\boldsymbol{\lambda}^0)$.

Proof: Recalling that $\boldsymbol{\sigma}^k = \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k$ and $\nabla D(\cdot)$ is L_D-Lipschitz continuous, we have for $k \in \mathcal{K}_D$

$$\begin{aligned}
&D\left(\boldsymbol{\lambda}^k\right) - D\left(\boldsymbol{\lambda}^{k+1}\right) \\
&\overset{a)}{\leq} - < \nabla D\left(\boldsymbol{\lambda}^k\right), \boldsymbol{\sigma}^k > + \frac{L_D}{2}\left\|\boldsymbol{\sigma}^k\right\|^2 \\
&\overset{b)}{=} < \boldsymbol{\nu}^k - \nabla D\left(\boldsymbol{\lambda}^k\right), \boldsymbol{\sigma}^k > - < \boldsymbol{\nu}^k, \boldsymbol{\sigma}^k > + \frac{L_D}{2}\left\|\boldsymbol{\sigma}^k\right\|^2 \\
&\overset{c)}{\leq} \left\|\boldsymbol{\nu}^k - \nabla D\left(\boldsymbol{\lambda}^k\right)\right\| \left\|\boldsymbol{\sigma}^k\right\| - \frac{1}{\alpha}\left\|\boldsymbol{\sigma}^k\right\|^2 + \frac{L_D}{2}\left\|\boldsymbol{\sigma}^k\right\|^2 \\
&\overset{d)}{\leq} L_D \sum_{\kappa=k-2k_0}^{k-1} \|\boldsymbol{\sigma}^\kappa\| \left\|\boldsymbol{\sigma}^k\right\| + \sqrt{2L_D\varepsilon_D}\left\|\boldsymbol{\sigma}^k\right\| \\
&\quad + \frac{L_D}{2}\left\|\boldsymbol{\sigma}^k\right\|^2 - \frac{1}{\alpha}\left\|\boldsymbol{\sigma}^k\right\|^2 \\
&\overset{e)}{\leq} \frac{L_D}{2} \sum_{\kappa=k-2k_0}^{k-1} \left\{\left\|\boldsymbol{\sigma}^\kappa\right\|^2 + \left\|\boldsymbol{\sigma}^k\right\|^2\right\} + \sqrt{2L_D\varepsilon_D}\left\|\boldsymbol{\sigma}^k\right\| \\
&\quad + \frac{L_D}{2}\left\|\boldsymbol{\sigma}^k\right\|^2 - \frac{1}{\alpha}\left\|\boldsymbol{\sigma}^k\right\|^2 \\
&= \frac{L_D}{2} \sum_{\kappa=k-2k_0}^{k} \left\|\boldsymbol{\sigma}^\kappa\right\|^2 + \sqrt{2L_D\varepsilon_D}\left\|\boldsymbol{\sigma}^k\right\| \\
&\quad + \left(k_0 L_D - \frac{1}{\alpha}\right)\left\|\boldsymbol{\sigma}^k\right\|^2
\end{aligned} \tag{25}$$

where a) follows from the property of Lipschitz continuity [45, Lemma 1.2.3], b) holds by adding and subtracting $< \boldsymbol{\nu}^k, \boldsymbol{\sigma}^k >$, c) is due to the Cauchy-Schwarz inequality and Lemma 4, d) follows from Lemma 5, and e) holds from $ab \leq (a^2 + b^2)/2, \forall a, b \in \mathbb{R}$.

Note that (25) also holds when $k \notin \mathcal{K}_D$, since $\boldsymbol{\sigma}^k = \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k = \mathbf{0}$. Summing (25) overall k, we have

$$\begin{aligned}
&D^0 - D\left(\boldsymbol{\lambda}^{k+1}\right) \\
&\leq \frac{L_D}{2} \sum_{\kappa=0}^{k} \sum_{k'=\kappa-2k_0}^{\kappa} \left\|\boldsymbol{\sigma}^{k'}\right\|^2 + \sqrt{2L_D\varepsilon_D} \sum_{\kappa=0}^{k} \|\boldsymbol{\sigma}^\kappa\| \\
&\quad + \left(k_0 L_D - \frac{1}{\alpha}\right) \sum_{\kappa=0}^{k} \|\boldsymbol{\sigma}^\kappa\|^2
\end{aligned}$$

$$\leq \frac{(2k_0+1)\,L_D}{2}\sum_{\kappa=0}^{k}\|\boldsymbol{\sigma}^{\kappa}\|^2+\sqrt{2\,(k+1)\,L_D\varepsilon_D}\sqrt{\sum_{\kappa=0}^{k}\|\boldsymbol{\sigma}^{\kappa}\|^2}$$
$$+\left(k_0L_D-\frac{1}{\alpha}\right)\sum_{\kappa=0}^{k}\|\boldsymbol{\sigma}^{\kappa}\|^2$$
$$=\left(\left(2k_0+\frac{1}{2}\right)L_D-\frac{1}{\alpha}\right)S^k+\sqrt{2\,(k+1)\,L_D\varepsilon_D}\sqrt{S^k}$$

where the second inequality holds by noting each nonnegative term $\|\boldsymbol{\sigma}^{k'}\|^2$ appears at most $2k_0+1$ times and applying the inequality $\sum_{i=1}^{m} a_i/m \leq \sqrt{\sum_{i=1}^{m} a_i^2/m}$, $\forall a_i \in \mathbb{R}$.

Then, we have

$$\gamma_\alpha S^k-\sqrt{2\,(k+1)\,L_D\varepsilon_D}\sqrt{S^k}\leq D\left(\boldsymbol{\lambda}^{k+1}\right)-D^0 \leq D^*-D^0. \quad (26)$$

The coefficient γ_α is positive if the step size α satisfies (23). Then, we complete the proof by solving the quadratic inequality (26) as

$$\sqrt{S^k}\leq\frac{\sqrt{2\,(k+1)\,L_D\varepsilon_D}+\sqrt{2\,(k+1)\,L_D\varepsilon_D+4\gamma_\alpha\,(D^*-D^0)}}{2\gamma_\alpha}$$
$$\leq\frac{2\sqrt{2\,(k+1)\,L_D\varepsilon_D}+\sqrt{4\gamma_\alpha\,(D^*-D^0)}}{2\gamma_\alpha}$$
$$=\frac{\sqrt{2L_D\varepsilon_D}}{\gamma_\alpha}\sqrt{k+1}+\sqrt{\frac{D^*-D^0}{\gamma_\alpha}}$$

where the second inequality holds since $\sqrt{a+b}\leq\sqrt{a}+\sqrt{b}\,\forall a,b\geq 0$. □

Remark 3 (Interaction Between Asynchrony and Inexactness): $\sqrt{S^k}$ characterizes the interaction between asynchrony and inexactness in the DD-DO algorithm. If the subproblem solutions are exact, i.e., $\varepsilon_D=0$, the $\mathcal{O}(\sqrt{k})$ term in (24) vanishes. In other words, $\sqrt{S^k}$ is not greater than a positive constant, and hence, $\lim_{k\to\infty}\|\boldsymbol{\sigma}^k\|=0$, which implies the convergence of $\{\boldsymbol{\lambda}^k\}$ in the asynchronous DD-DO algorithm, as proved in [15]. However, if the subproblem solutions are inexact, i.e., $\varepsilon_D>0$, the errors will be accumulated, leading to the increasing of $\sqrt{S^k}$ in $\mathcal{O}(\sqrt{k})$, and hence, $\{\boldsymbol{\lambda}^k\}$ fails to converge to some $\boldsymbol{\lambda}\in\Lambda^*$.

B. Bound on the Dual Variable

We turn to showing that in Algorithm 2, the sequence $\{\|\boldsymbol{\lambda}^k\|\}$ increases not faster than $\mathcal{O}(\sqrt{k})$. We start from the following lemma.

Lemma 7: Suppose Assumptions A1 and A3 hold. In Algorithm 2, we have for any $\boldsymbol{\xi},\boldsymbol{\mu}\in\mathbb{R}_+^m$

$$0\leq\widetilde{D}_i(\boldsymbol{\xi})-D_i(\boldsymbol{\mu})+<\boldsymbol{\mu}-\boldsymbol{\xi},A_i\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})> \leq L_i\|\boldsymbol{\mu}-\boldsymbol{\xi}\|^2+2\varepsilon_i.$$

Proof: For the left-hand side inequality, from the definition of $\boldsymbol{x}_i(\cdot)$, we have for any $\boldsymbol{\mu}\in\mathbb{R}_+^m$

$$D_i(\boldsymbol{\mu})=\min_{\boldsymbol{x}_i\in\mathcal{X}_i} f_i(\boldsymbol{x}_i)+<\boldsymbol{\mu},A_i\boldsymbol{x}_i>$$
$$=f_i(\boldsymbol{x}_i(\boldsymbol{\mu}))+<\boldsymbol{\mu},A_i\boldsymbol{x}_i(\boldsymbol{\mu})>$$
$$\leq f_i(\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi}))+<\boldsymbol{\mu},A_i\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})>$$
$$=\widetilde{D}_i(\boldsymbol{\xi})+<\boldsymbol{\mu}-\boldsymbol{\xi},A_i\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})>.$$

For the right-hand side inequality, from the inequality $ab\leq(a^2+b^2)/2\,\forall a,b\in\mathbb{R}$, we have

$$\|A_i\|\,\|\boldsymbol{\mu}-\boldsymbol{\xi}\|\,\|\boldsymbol{x}_i(\boldsymbol{\xi})-\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})\|$$
$$=\frac{\|A_i\|}{\sqrt{c_i}}\|\boldsymbol{\mu}-\boldsymbol{\xi}\|\cdot\sqrt{c_i}\,\|\boldsymbol{x}_i(\boldsymbol{\xi})-\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})\|$$
$$\leq\frac{\|A_i\|^2}{2c_i}\|\boldsymbol{\mu}-\boldsymbol{\xi}\|^2+\frac{c_i}{2}\|\boldsymbol{x}_i(\boldsymbol{\xi})-\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})\|^2$$
$$=\frac{L_i}{2}\|\boldsymbol{\mu}-\boldsymbol{\xi}\|^2+\frac{c_i}{2}\|\boldsymbol{x}_i(\boldsymbol{\xi})-\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})\|^2. \quad (27)$$

Recalling that $\nabla D_i(\cdot)$ is L_i-Lipschitz continuous, we have

$$D_i(\boldsymbol{\mu})\overset{a)}{\geq}D_i(\boldsymbol{\xi})+<\boldsymbol{\mu}-\boldsymbol{\xi},A_i\boldsymbol{x}_i(\boldsymbol{\xi})>-\frac{L_i}{2}\|\boldsymbol{\mu}-\boldsymbol{\xi}\|^2$$
$$\overset{b)}{\geq}\widetilde{D}_i(\boldsymbol{\xi})-\varepsilon_i+<\boldsymbol{\mu}-\boldsymbol{\xi},A_i(\boldsymbol{x}_i(\boldsymbol{\xi})-\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi}))>$$
$$+<\boldsymbol{\mu}-\boldsymbol{\xi},A_i\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})>-\frac{L_i}{2}\|\boldsymbol{\mu}-\boldsymbol{\xi}\|^2$$
$$\overset{c)}{\geq}\widetilde{D}_i(\boldsymbol{\xi})-\varepsilon_i-\|A_i\|\,\|\boldsymbol{\mu}-\boldsymbol{\xi}\|\,\|\boldsymbol{x}_i(\boldsymbol{\xi})-\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})\|$$
$$+<\boldsymbol{\mu}-\boldsymbol{\xi},A_i\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})>-\frac{L_i}{2}\|\boldsymbol{\mu}-\boldsymbol{\xi}\|^2$$
$$\overset{d)}{\geq}\widetilde{D}_i(\boldsymbol{\xi})-\varepsilon_i-\frac{c_i}{2}\|\boldsymbol{x}_i(\boldsymbol{\xi})-\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})\|^2-\frac{L_i}{2}\|\boldsymbol{\mu}-\boldsymbol{\xi}\|^2$$
$$+<\boldsymbol{\mu}-\boldsymbol{\xi},A_i\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})>-\frac{L_i}{2}\|\boldsymbol{\mu}-\boldsymbol{\xi}\|^2$$
$$\overset{e)}{\geq}\widetilde{D}_i(\boldsymbol{\xi})-2\varepsilon_i+<\boldsymbol{\mu}-\boldsymbol{\xi},A_i\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})>-L_i\|\boldsymbol{\mu}-\boldsymbol{\xi}\|^2$$

where a) follows from the property of Lipschitz continuity [45, Lemma 1.2.3], b) holds by adding and subtracting $<\boldsymbol{\mu}-\boldsymbol{\xi},A_i\widetilde{\boldsymbol{x}}_i(\boldsymbol{\xi})>$ and using Assumption A3, c) follows from the Cauchy − Schwarz inequality, d) holds from (27), and e) is due to Lemma 3. □

Then, we have the following theorem on the bounded dual variables.

Theorem 8: Suppose Assumptions A1–A3 hold. In Algorithm 2, if the step size satisfies

$$0<\alpha\leq\frac{1}{4L_D} \quad (28)$$

then the dual variable $\boldsymbol{\lambda}^{k+1}$ is bounded by

$$\left\|\boldsymbol{\lambda}^{k+1}\right\|\leq 2\left\|\boldsymbol{\lambda}^*\right\|+\left\|\boldsymbol{\lambda}^0\right\|+2\sqrt{\alpha\varepsilon_D}\sqrt{k+1}+2k_0\sqrt{S^k} \quad \forall\boldsymbol{\lambda}^*\in\Lambda^*.$$

Proof: For $k\in\mathcal{K}_D$, applying the projection theorem [44, Prop. 2.1.3], we have

$$\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k-\alpha\boldsymbol{\nu}^k,\boldsymbol{\lambda}-\boldsymbol{\lambda}^{k+1}\rangle\geq 0\ \forall\boldsymbol{\lambda}\geq 0. \quad (29)$$

It follows that

$$\begin{aligned}
&\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}\right\|^2\\
&= \left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k+\boldsymbol{\lambda}^k-\boldsymbol{\lambda}\right\|^2\\
&= \left\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}\right\|^2+\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\right\|^2+2\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k,\boldsymbol{\lambda}^k-\boldsymbol{\lambda}\rangle\\
&\overset{a)}{=} \left\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}\right\|^2+\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\right\|^2\\
&\quad+2\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k,\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^{k+1}\rangle+2\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k,\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}\rangle\\
&= \left\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}\right\|^2-\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\right\|^2+2\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k,\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}\rangle\\
&\overset{b)}{\le} \left\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}\right\|^2-4\alpha L_D\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\right\|^2+2\alpha\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda},\boldsymbol{\nu}^k\rangle\\
&\overset{c)}{\le} \left\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}\right\|^2-2\alpha\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda},\boldsymbol{b}\rangle\\
&\quad+2\alpha\sum_{i\in\mathcal{N}}\left\{\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda},A_i\widehat{\boldsymbol{x}}_i^k\rangle-2L_i\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\right\|^2\right\}
\end{aligned}\tag{30}$$

where a) holds by adding and subtracting $\boldsymbol{\lambda}^{k+1}$ in the inner product term, b) is due to $4\alpha L_D \le 1$ and (29), and c) follows from the definitions of $\boldsymbol{\nu}^k$ in (15) and L_D in Corollary 2.

For $i \in \mathcal{N}$, we have

$$\begin{aligned}
&\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda},A_i\widehat{\boldsymbol{x}}_i^k\rangle-2L_i\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^k\right\|^2\\
&\overset{a)}{\le} \underbrace{\left\langle\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^{k-\delta_i^k},A_i\widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)\right\rangle-L_i\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^{k-\delta_i^k}\right\|^2}_{(\Delta_1)}\\
&\quad-\underbrace{\left\langle\boldsymbol{\lambda}-\boldsymbol{\lambda}^{k-\delta_i^k},A_i\widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)\right\rangle}_{(\Delta_2)}+2L_i\left\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}^{k-\delta_i^k}\right\|^2\\
&\overset{b)}{\le} D_i\left(\boldsymbol{\lambda}^{k+1}\right)-\widetilde{D}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)+2\varepsilon_i\\
&\quad+\widetilde{D}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)-D_i\left(\boldsymbol{\lambda}\right)+2L_i\delta_i^k\sum_{\kappa=k-\delta_i^k}^{k-1}\left\|\boldsymbol{\sigma}^\kappa\right\|^2\\
&\overset{c)}{\le} D_i\left(\boldsymbol{\lambda}^{k+1}\right)-D_i\left(\boldsymbol{\lambda}\right)+2\varepsilon_i+4k_0L_i\sum_{\kappa=k-2k_0}^{k-1}\left\|\boldsymbol{\sigma}^\kappa\right\|^2
\end{aligned}\tag{31}$$

where a) holds by adding and subtracting $\boldsymbol{\lambda}^{k-\delta_i^k}$ in both the inner product term and the 2-norm term, applying the inequality $\|\boldsymbol{a}+\boldsymbol{b}\|^2 \le 2\|\boldsymbol{a}\|^2+2\|\boldsymbol{b}\|^2, \forall \boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^n$, and using the definition of $\widehat{\boldsymbol{x}}_i^k$ in (18), in b) the term (Δ_1) is relaxed by setting $\boldsymbol{\xi}=\boldsymbol{\lambda}^{k-\delta_i^k}$ and $\boldsymbol{\mu}=\boldsymbol{\lambda}^{k+1}$ in the right-hand side inequality in Lemma 7, (Δ_2) is slacked by setting $\boldsymbol{\xi}=\boldsymbol{\lambda}^{k-\delta_i^k}$ and $\boldsymbol{\mu}=\boldsymbol{\lambda}$ in the left-hand side inequality in Lemma 7, and the quadratic term $\|\boldsymbol{\lambda}^k-\boldsymbol{\lambda}^{k-\delta_i^k}\|^2$ is relaxed by the inequality $\|\sum_{i=1}^m \boldsymbol{b}_i\|^2 \le m\sum_{i=1}^m\|\boldsymbol{b}_i\|^2, \forall \boldsymbol{b}_i\in\mathbb{R}^n$, and c) holds since $\delta_i^k \le 2k_0$.

Following (30), we have

$$\begin{aligned}
&\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}\right\|^2-\left\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}\right\|^2\\
&\le -2\alpha<\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda},\boldsymbol{b}>+2\alpha\sum_{i\in\mathcal{N}}\left\{D_i\left(\boldsymbol{\lambda}^{k+1}\right)-D_i\left(\boldsymbol{\lambda}\right)\right.\\
&\quad\left.+2\varepsilon_i+4k_0L_i\sum_{\kappa=k-2k_0}^{k-1}\left\|\boldsymbol{\sigma}^\kappa\right\|^2\right\}\\
&\le 2\alpha\left(D\left(\boldsymbol{\lambda}^{k+1}\right)-D\left(\boldsymbol{\lambda}\right)\right)+4\alpha\varepsilon_D+2k_0\sum_{\kappa=k-2k_0}^{k-1}\left\|\boldsymbol{\sigma}^\kappa\right\|^2
\end{aligned}\tag{32}$$

where the first inequality holds from (31) and the second inequality follows from $D(\boldsymbol{\lambda})=\sum_{i\in\mathcal{N}}D_i(\boldsymbol{\lambda})-<\boldsymbol{\lambda},\boldsymbol{b}>$, $\varepsilon_D=\sum_{i\in\mathcal{N}}\varepsilon_i$, and $4\alpha\sum_{i\in\mathcal{N}}L_i=4\alpha L_D\le 1$.

By replacing $\boldsymbol{\lambda}$ with an arbitrary $\boldsymbol{\lambda}^*\in\Lambda^*$ in (32) and noting that $D(\boldsymbol{\lambda}^{k+1})-D(\boldsymbol{\lambda}^*)\le 0$, we have

$$\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*\right\|^2\le\left\|\boldsymbol{\lambda}^{k}-\boldsymbol{\lambda}^*\right\|^2+4\alpha\varepsilon_D+2k_0\sum_{\kappa=k-2k_0}^{k-1}\left\|\boldsymbol{\sigma}^\kappa\right\|^2.$$

Note that the aforementioned inequality also holds when $k\notin\mathcal{K}_D$, since $\boldsymbol{\lambda}^{k+1}=\boldsymbol{\lambda}^k$. Applying the inequality $k+1$ times, we get

$$\left\|\boldsymbol{\lambda}^{k+1}-\boldsymbol{\lambda}^*\right\|^2\le\left\|\boldsymbol{\lambda}^{0}-\boldsymbol{\lambda}^*\right\|^2+4\alpha\left(k+1\right)\varepsilon_D+4k_0^2S^k$$

where we consider each non-negative term $\|\sigma^\kappa\|^2$ appears at most $2k_0$ times.

By expanding the quadratic terms, eliminating the term $\|\boldsymbol{\lambda}^*\|^2$ on each side, relaxing the nonpositive term $-<\boldsymbol{\lambda}^0,\boldsymbol{\lambda}^*>$, and applying the inequality $<\boldsymbol{\lambda}^*,\boldsymbol{\lambda}^{k+1}>\le\|\boldsymbol{\lambda}^*\|\|\boldsymbol{\lambda}^{k+1}\|$, we have

$$\left\|\boldsymbol{\lambda}^{k+1}\right\|^2-2\left\|\boldsymbol{\lambda}^*\right\|\left\|\boldsymbol{\lambda}^{k+1}\right\|\le\left\|\boldsymbol{\lambda}^0\right\|^2+4\alpha\left(k+1\right)\varepsilon_D+4k_0^2S^k.$$

Then, we complete the proof by solving the aforementioned quadratic inequality as

$$\begin{aligned}
&\left\|\boldsymbol{\lambda}^{k+1}\right\|\\
&\le\frac{2\left\|\boldsymbol{\lambda}^*\right\|+\sqrt{4\left\|\boldsymbol{\lambda}^*\right\|^2+4\left\|\boldsymbol{\lambda}^0\right\|^2+16\alpha\left(k+1\right)\varepsilon_D+16k_0^2S^k}}{2}\\
&\le 2\left\|\boldsymbol{\lambda}^*\right\|+\left\|\boldsymbol{\lambda}^0\right\|+2\sqrt{\alpha\varepsilon_D}\sqrt{k+1}+2k_0\sqrt{S^k}
\end{aligned}$$

where the second inequality holds due to $\sqrt{a+b}\le\sqrt{a}+\sqrt{b}\,\forall a,b\ge 0$. □

C. Convergence Analysis

Based on Theorems 6 and 8, we turn to proving that the asynchronous and inexact DD-DO algorithm converges in $\mathcal{O}(1/\sqrt{k})$.

Instead of the primal and dual sequences, we consider their running averages over iteration, which are defined as

$$\bar{\boldsymbol{x}}^k:=\frac{1}{\left|\mathcal{K}_D^k\right|}\sum_{\kappa\in\mathcal{K}_D^k}\widehat{\boldsymbol{x}}^\kappa,\ \bar{\boldsymbol{\lambda}}^{k+1}:=\frac{1}{\left|\mathcal{K}_D^k\right|}\sum_{\kappa\in\mathcal{K}_D^k}\boldsymbol{\lambda}^{\kappa+1}\tag{33}$$

where $\mathcal{K}_D^k$ is defined as

$$\mathcal{K}_D^k=\left\{\kappa\in\mathcal{K}_D\,|\,\kappa\le k\right\}.\tag{34}$$

Under Assumption A2, it follows that

$$\frac{k+1}{2k_0+1} \le \left|\mathcal{K}_D^k\right| \le k+1. \tag{35}$$

We have the following theorem of convergence.

Theorem 9: Suppose Assumptions A1–A3 hold. If the step size satisfies

$$0 < \alpha < \min\left\{\frac{1}{(2k_0+1/2)\,L_D}, \frac{1}{4L_D}\right\} \tag{36}$$

given any $\boldsymbol{\lambda}^* \in \Lambda^*$, Algorithm 2 has the following convergence performance.

1) The violation of the constraints is bounded by

$$\left\|\left[A\bar{\boldsymbol{x}}^k - \boldsymbol{b}\right]^+\right\| \le \frac{M_{1/2}}{\sqrt{k+1}} + \frac{M_1}{k+1}. \tag{37}$$

2) The deviation of primal value is bounded by

$$-\frac{M_{1/2}\,\|\boldsymbol{\lambda}^*\|}{\sqrt{k+1}} - \frac{M_1\,\|\boldsymbol{\lambda}^*\|}{k+1} \le F\left(\bar{\boldsymbol{x}}^k\right) - F^* \le N_0 + \frac{N_{1/2}}{\sqrt{k+1}} + \frac{N_1}{k+1}. \tag{38}$$

3) The deviation of dual value is bounded by

$$0 \le D^* - D\left(\bar{\boldsymbol{\lambda}}^{k+1}\right) \le N_0 + \frac{N_{1/2}}{\sqrt{k+1}} + \frac{N_1'}{k+1}. \tag{39}$$

4) The deviation of primal average variable is bounded by

$$\left\|\bar{\boldsymbol{x}}^k - \boldsymbol{x}^*\right\|^2 \le \frac{2N_0}{c_F} + \frac{2\left(N_{1/2} + \|\boldsymbol{\lambda}^*\|\,M_{1/2}\right)}{c_F\sqrt{k+1}} + \frac{2\left(N_1 + \|\boldsymbol{\lambda}^*\|\,M_1\right)}{c_F\,(k+1)} \tag{40}$$

where $M_{1/2}, M_1, N_0, N_{1/2}, N_1$, and N_1' are positive constants defined as

$$M_{1/2} := \frac{2k_0+1}{\alpha}\left(2\sqrt{\alpha\varepsilon_D} + \frac{2k_0\sqrt{2L_D\varepsilon_D}}{\gamma_\alpha}\right)$$

$$M_1 := \frac{2k_0+1}{\alpha}\left(2\,\|\boldsymbol{\lambda}^*\| + \left\|\boldsymbol{\lambda}^0\right\| + 2k_0\sqrt{\frac{D^*-D^0}{\gamma_\alpha}}\right)$$

$$N_0 := 2\varepsilon_D + \frac{4k_0^2\,(2k_0+1)\,L_D\varepsilon_D}{\alpha\gamma_\alpha^2}$$

$$N_{1/2} := \frac{4k_0^2(2k_0+1)\sqrt{2L_D\varepsilon_D\,(D^*-D^0)}}{\alpha\gamma_\alpha^{3/2}}$$

$$N_1 := \frac{2k_0+1}{2\alpha}\left(\left\|\boldsymbol{\lambda}^0\right\|^2 + \frac{4k_0^2\left(D^*-D^0\right)}{\gamma_\alpha}\right)$$

$$N_1' := \frac{2k_0+1}{2\alpha}\left(\left\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\right\|^2 + \frac{4k_0^2\left(D^*-D^0\right)}{\gamma_\alpha}\right).$$

Proof: 1) If $k \in \mathcal{K}_D$, we have

$$\boldsymbol{\lambda}^{k+1} = \left[\boldsymbol{\lambda}^k + \alpha\left(A\widehat{\boldsymbol{x}}^k - \boldsymbol{b}\right)\right]^+ \ge \boldsymbol{\lambda}^k + \alpha\left(A\widehat{\boldsymbol{x}}^k - \boldsymbol{b}\right)$$

i.e.,

$$\alpha\left(A\widehat{\boldsymbol{x}}^k - \boldsymbol{b}\right) \le \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k.$$

If $k \notin \mathcal{K}_D$, we have $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k$, which indicates

$$0 \le \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k.$$

Summing overall $k \in \mathcal{K}$, we obtain

$$\alpha \sum_{\kappa\in\mathcal{K}_D^k} \left(A\widehat{\boldsymbol{x}}^\kappa - \boldsymbol{b}\right) \le \boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^0 \le \boldsymbol{\lambda}^{k+1}.$$

By taking the average and applying $\left\|[\boldsymbol{y}]^+\right\| \le \left\|\boldsymbol{y}\right\|, \forall \boldsymbol{y} \in \mathbb{R}^n$, we have

$$\left\|\left[A\bar{\boldsymbol{x}}^k - \boldsymbol{b}\right]^+\right\| \le \frac{\left\|\boldsymbol{\lambda}^{k+1}\right\|}{\alpha\left|\mathcal{K}_D^k\right|} \le \frac{2k_0+1}{\alpha\,(k+1)}\left\|\boldsymbol{\lambda}^{k+1}\right\|.$$

Invoking Theorems 6 and 8 immediately yields (37).

2) For the left-hand side inequality of (38), the optimality of F^* and $\boldsymbol{\lambda}^* \ge 0$ yields

$$F^* \le F\left(\bar{\boldsymbol{x}}^k\right) + \langle \boldsymbol{\lambda}^*, A\bar{\boldsymbol{x}}^k - \boldsymbol{b}\rangle \le F\left(\bar{\boldsymbol{x}}^k\right) + \left\|\boldsymbol{\lambda}^*\right\|\left\|\left[A\bar{\boldsymbol{x}}^k - \boldsymbol{b}\right]^+\right\|.$$

Applying (37) establishes the left-hand side inequality.

For the right-hand side inequality of (38), if $k \in \mathcal{K}_D$, it follows that

$$\begin{aligned}
&\left\|\boldsymbol{\lambda}^{k+1}\right\|^2 - \left\|\boldsymbol{\lambda}^k\right\|^2 + 2\alpha\langle\boldsymbol{\lambda}^{k+1}, \boldsymbol{b}\rangle \\
&\overset{a)}{\le} 2\alpha\sum_{i\in\mathcal{N}}\left\{\langle\boldsymbol{\lambda}^{k+1}, A_i\widehat{\boldsymbol{x}}_i^k\rangle - 2L_i\left\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^k\right\|^2\right\} \\
&\overset{b)}{\le} 2\alpha\sum_{i\in\mathcal{N}}\left\{\underbrace{\left\langle\boldsymbol{\lambda}^{k-\delta_i^k}, A_i\widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)\right\rangle}_{(\Delta_3)} + 2L_i\left\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-\delta_i^k}\right\|^2\right. \\
&\qquad\left. + \underbrace{\left\langle\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^{k-\delta_i^k}, A_i\widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)\right\rangle - L_i\left\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^{k-\delta_i^k}\right\|^2}_{(\Delta_4)}\right\} \\
&\overset{c)}{\le} 2\alpha\sum_{i\in\mathcal{N}}\left\{\widetilde{D}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right) - f_i\left(\widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right)\right)\right. \\
&\qquad\left. + 2L_i\delta_i^k\sum_{\kappa=k-\delta_i^k}^{k-1}\|\boldsymbol{\sigma}^\kappa\|^2 + D_i\left(\boldsymbol{\lambda}^{k+1}\right) - \widetilde{D}_i\left(\boldsymbol{\lambda}^{k-\delta_i^k}\right) + 2\varepsilon_i\right\} \\
&\overset{d)}{\le} 2\alpha\sum_{i\in\mathcal{N}}D_i\left(\boldsymbol{\lambda}^{k+1}\right) - 2\alpha F\left(\widehat{\boldsymbol{x}}^k\right) + 4\alpha\varepsilon_D + 2k_0\sum_{\kappa=k-2k_0}^{k-1}\|\boldsymbol{\sigma}^\kappa\|^2
\end{aligned} \tag{41}$$

where a) holds by taking $\boldsymbol{\lambda} = 0$ in (30); b) holds by adding and subtracting $\boldsymbol{\lambda}^{k-\delta_i^k}$ in both the inner product term and the 2-norm term, applying the inequality $\|\boldsymbol{a} + \boldsymbol{b}\|^2 \le 2\|\boldsymbol{a}\|^2 + 2\|\boldsymbol{b}\|^2 \forall \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$, and using the definition of $\widehat{\boldsymbol{x}}_i^k$ in (18); in c), (Δ_3) follows from the definition of $\widetilde{D}_i(\boldsymbol{\lambda}^{k-\delta_i^k})$ in (19), (Δ_4) is relaxed by setting $\boldsymbol{\xi} = \boldsymbol{\lambda}^{k-\delta_i^k}$ and $\boldsymbol{\mu} = \boldsymbol{\lambda}^{k+1}$ in the right-hand side inequality in Lemma

7, and the quadratic term $\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^{k-\delta_i^k}\|^2$ is relaxed by the inequality $\|\sum_{i=1}^m \boldsymbol{b}_i\|^2 \le m\sum_{i=1}^m \|\boldsymbol{b}_i\|^2 \forall \boldsymbol{b}_i \in \mathbb{R}^n$; and d) is due to $F(\widehat{\boldsymbol{x}}^k) = \sum_{i\in\mathcal{N}} f_i(\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda}^{k-\delta_i^k}))$, $\varepsilon_D = \sum_{i\in\mathcal{N}} \varepsilon_i$, $4\alpha\sum_{i\in\mathcal{N}} L_i = 4\alpha L_D \le 1$, and $\delta_i^k \le 2k_0$.

Note that

$$\sum_{i\in\mathcal{N}} D_i\left(\boldsymbol{\lambda}^{k+1}\right) - <\boldsymbol{\lambda}^{k+1}, \boldsymbol{b}> = D\left(\boldsymbol{\lambda}^{k+1}\right) \le D^* = F^*.$$

Following (41), we have

$$\frac{\left\|\boldsymbol{\lambda}^{k+1}\right\|^2 - \left\|\boldsymbol{\lambda}^k\right\|^2}{2\alpha} \le F^* - F(\widehat{\boldsymbol{x}}^k) + 2\varepsilon_D + \frac{k_0}{\alpha}\sum_{\kappa=k-2k_0}^{k-1}\|\boldsymbol{\sigma}^\kappa\|^2.$$

If $k \notin \mathcal{K}_D$, it obviously follows that

$$\frac{\left\|\boldsymbol{\lambda}^{k+1}\right\|^2 - \left\|\boldsymbol{\lambda}^k\right\|^2}{2\alpha} \le 0.$$

Summing overall $k \in \mathcal{K}$, we obtain

$$\frac{\left\|\boldsymbol{\lambda}^{k+1}\right\|^2 - \left\|\boldsymbol{\lambda}^0\right\|^2}{2\alpha} \le \sum_{\kappa\in\mathcal{K}_D^k}\left(F^* - F(\widehat{\boldsymbol{x}}^\kappa)\right) + 2\left|\mathcal{K}_D^k\right|\varepsilon_D + \frac{2k_0^2}{\alpha}S^k$$

where we consider each nonnegative term $\|\sigma^\kappa\|^2$ appears at most $2k_0$ times.

By taking the average and using the convexity of $F(\cdot)$, we have

$$\begin{aligned} F\left(\bar{\boldsymbol{x}}^k\right) - F^* &\le 2\varepsilon_D + \frac{1}{2\alpha\left|\mathcal{K}_D^k\right|}\left(\left\|\boldsymbol{\lambda}^0\right\|^2 - \left\|\boldsymbol{\lambda}^{k+1}\right\|^2 + 4k_0^2 S^k\right) \\ &\le 2\varepsilon_D + \frac{2k_0+1}{2\alpha(k+1)}\left(\left\|\boldsymbol{\lambda}^0\right\|^2 + 4k_0^2 S^k\right). \end{aligned}$$

Invoking Theorem 6 immediately yields the right-hand side inequality of (38).

3) From the optimality of D^*, the left-hand side of (39) holds directly. For the right-hand side, by setting $\boldsymbol{\lambda} = \boldsymbol{\lambda}^*$ in (32), we have for $k \in \mathcal{K}_D$

$$\begin{aligned} &\left\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\right\|^2 - \left\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\right\|^2 \\ &\le 2\alpha\left(D\left(\boldsymbol{\lambda}^{k+1}\right) - D^*\right) + 4\alpha\varepsilon_D + 2k_0\sum_{\kappa=k-2k_0}^{k-1}\|\boldsymbol{\sigma}^\kappa\|^2. \end{aligned}$$

If $k \notin \mathcal{K}_D$, it obviously follows that

$$\left\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\right\|^2 - \left\|\boldsymbol{\lambda}^k - \boldsymbol{\lambda}^*\right\|^2 \le 0.$$

Summing overall $k \in \mathcal{K}$, we obtain

$$\begin{aligned} &\frac{\left\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\right\|^2 - \left\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\right\|^2}{2\alpha} \\ &\quad\le \sum_{\kappa\in\mathcal{K}_D^k}\left(D\left(\boldsymbol{\lambda}^{\kappa+1}\right) - D^*\right) + 2\left|\mathcal{K}_D^k\right|\varepsilon_D + \frac{2k_0^2}{\alpha}S^k. \end{aligned}$$

By taking the average and using the concavity of $D(\cdot)$, we have

$$\begin{aligned} D^* - D\left(\bar{\boldsymbol{\lambda}}^{k+1}\right) &\le 2\varepsilon_D + \frac{1}{2\alpha\left|\mathcal{K}_D^k\right|}\left(\left\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\right\|^2 - \left\|\boldsymbol{\lambda}^{k+1} - \boldsymbol{\lambda}^*\right\|^2 + 4k_0^2 S^k\right) \\ &\le 2\varepsilon_D + \frac{2k_0+1}{2\alpha(k+1)}\left(\left\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\right\|^2 + 4k_0^2 S^k\right). \end{aligned}$$

Applying Theorem 6 directly yields the right-hand side inequality of (39).

4) Recall that $\boldsymbol{x}^*$ is the optimal solution to the problem (1). From the optimality condition [46, Th. 3.25], we have

$$\langle\nabla F(\boldsymbol{x}^*) + A^T\boldsymbol{\lambda}^*, \boldsymbol{x} - \boldsymbol{x}^*\rangle \ge 0 \ \forall \boldsymbol{x} \in \mathcal{X} \tag{42a}$$

$$A\boldsymbol{x}^* \le \boldsymbol{b}, \ \boldsymbol{\lambda}^* \ge 0 \tag{42b}$$

$$\langle\boldsymbol{\lambda}^*, A\boldsymbol{x}^* - \boldsymbol{b}\rangle = 0. \tag{42c}$$

By replacing $\boldsymbol{x}$ with $\bar{\boldsymbol{x}}^k$ in (42a), we obtain

$$\begin{aligned} &-\langle\nabla F(\boldsymbol{x}^*), \bar{\boldsymbol{x}}^k - \boldsymbol{x}^*\rangle \\ &\le \langle A^T\boldsymbol{\lambda}^*, \bar{\boldsymbol{x}}^k - \boldsymbol{x}^*\rangle \\ &= \langle\boldsymbol{\lambda}^*, A\bar{\boldsymbol{x}}^k - \boldsymbol{b}\rangle - \langle\boldsymbol{\lambda}^*, A\boldsymbol{x}^* - \boldsymbol{b}\rangle \\ &= \langle\boldsymbol{\lambda}^*, A\bar{\boldsymbol{x}}^k - \boldsymbol{b}\rangle. \end{aligned} \tag{43}$$

From the strong convexity of $F(\cdot)$, we have

$$\begin{aligned} \frac{c_F}{2}\left\|\bar{\boldsymbol{x}}^k - \boldsymbol{x}^*\right\|^2 &\le F\left(\bar{\boldsymbol{x}}^k\right) - F^* - \langle\nabla F(\boldsymbol{x}^*), \bar{\boldsymbol{x}}^k - \boldsymbol{x}^*\rangle \\ &\overset{(43)}{\le} F\left(\bar{\boldsymbol{x}}^k\right) - F^* + <\boldsymbol{\lambda}^*, A\bar{\boldsymbol{x}}^k - \boldsymbol{b}> \\ &\le F\left(\bar{\boldsymbol{x}}^k\right) - F^* + \left\|\boldsymbol{\lambda}^*\right\|\left\|\left[A\bar{\boldsymbol{x}}^k - \boldsymbol{b}\right]^+\right\|. \end{aligned}$$

The proof is completed by considering (37) and the right-hand side inequality in (38). □

Remark 4 (Constant step size): A constant step size is used to attain the convergence performance in Theorem 9, instead of a diminishing one. Compared with a constant step size, a diminishing step size may lead to a slow convergence rate near the eventual limit and usually needs additional experimentations to determine how fast the step size declines [46], [47], [48]. Hence, a constant step size is more applicable and popular in practice.

Remark 5 (Generality): Theorem 9 indicates the uniform ultimate boundedness of the DD-DO algorithm considering asynchrony and inexactness. The convergence results considering only asynchrony [15] or inexactness [26] can be regarded as special cases of our result by simply setting the inexactness or asynchrony parameter as zero. On the one hand, if the algorithm is synchronous ($k_0 = 0$) and inexact ($\varepsilon_D > 0$), $N_{1/2}$ vanishes, while other parameters decrease, similarly to [26]. It indicates that asynchrony commonly slows down the convergence and magnifies errors. On the other hand, if the algorithm is asynchronous ($k_0 > 0$) and exact ($\varepsilon_D = 0$), M_1, N_1 and N_1' are unchanged, while the rest parameters are zero. Here, the algorithm converges to the optimal solution, similarly to [15].

Moreover, we show that the asynchronous algorithm converges in $\mathcal{O}(1/k)$, which, to the best of our knowledge, has not been presented in the existing literature [6], [15], [16], [17].

Corollary 10: Suppose Assumptions A1–A3 hold. If the step size satisfies (36), the *asynchronous* and *exact* version of Algorithm 2 has the following convergence performance:

$$\left\| \left[A\bar{\boldsymbol{x}}^k - \boldsymbol{b}\right]^+ \right\| \leq \frac{M_1}{k+1} \tag{44a}$$

$$-\frac{M_1 \|\boldsymbol{\lambda}^*\|}{k+1} \leq F\left(\bar{\boldsymbol{x}}^k\right) - F^* \leq \frac{N_1}{k+1} \tag{44b}$$

$$0 \leq D^* - D\left(\bar{\boldsymbol{\lambda}}^{k+1}\right) \leq \frac{N_1'}{k+1} \tag{44c}$$

$$\left\|\bar{\boldsymbol{x}}^k - \boldsymbol{x}^*\right\|^2 \leq \frac{2\left(N_1 + \|\boldsymbol{\lambda}^*\| M_1\right)}{c_F (k+1)} \tag{44d}$$

where M_1, N_1, and N_1' are defined as Theorem 9.

In addition, by setting $k_0 = 0$ and $\varepsilon_D = 0$, the results in Theorem 9 reduce to the synchronous and exact case as follows. Note that the convergence rate is $\mathcal{O}(1/k)$, which is the same as the results of [39] and [40].

Corollary 11: Suppose Assumptions A1–A3 hold. If the step size satisfies (36), the *synchronous* and *exact* version of Algorithm 2 has the following convergence performance:

$$\left\| \left[A\bar{\boldsymbol{x}}^k - \boldsymbol{b}\right]^+ \right\| \leq \frac{2\|\boldsymbol{\lambda}^*\| + \|\boldsymbol{\lambda}^0\|}{\alpha(k+1)} \tag{45a}$$

$$-\frac{2\|\boldsymbol{\lambda}^*\|^2 + \|\boldsymbol{\lambda}^0\|\|\boldsymbol{\lambda}^*\|}{\alpha(k+1)} \leq F\left(\bar{\boldsymbol{x}}^k\right) - F^* \leq \frac{\|\boldsymbol{\lambda}^0\|^2}{2\alpha(k+1)} \tag{45b}$$

$$0 \leq D^* - D\left(\bar{\boldsymbol{\lambda}}^{k+1}\right) \leq \frac{\|\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*\|^2}{2\alpha(k+1)} \tag{45c}$$

$$\left\|\bar{\boldsymbol{x}}^k - \boldsymbol{x}^*\right\|^2 \leq \frac{\|\boldsymbol{\lambda}^0\|^2 + 2\|\boldsymbol{\lambda}^0\|\|\boldsymbol{\lambda}^*\| + 4\|\boldsymbol{\lambda}^*\|^2}{\alpha c_F (k+1)}. \tag{45d}$$

Remark 6 (Averaging to combat inexactness): In our main results, the running average sequences $\{\bar{\boldsymbol{x}}^k\}$ and $\{\bar{\boldsymbol{\lambda}}^{k+1}\}$ defined in (33) are leveraged to develop our main results. The insight is that averaging can alleviate errors in iterative algorithms [25], [26], [49], [50]. From [42], the violation of the constraints and the errors of the objectives can be reduced by averaging due to the convexity of constraints and the primal objective and the concavity of the dual objective.

For instance, consider a sequence $\{\boldsymbol{\lambda}^k\}$ satisfying $D^* - D(\boldsymbol{\lambda}^k) \leq \varepsilon^k$. From the Jensen's inequality [51] and the concavity of $D(\cdot)$, we have

$$D\left(\frac{1}{k}\sum_{\kappa=1}^{k}\boldsymbol{\lambda}^\kappa\right) \geq \frac{1}{k}\sum_{\kappa=1}^{k} D\left(\boldsymbol{\lambda}^\kappa\right) \geq D^* - \frac{1}{k}\sum_{\kappa=1}^{k}\varepsilon^\kappa.$$

As long as the sum of the errors $\sum_{\kappa=1}^{k}\varepsilon^\kappa$ increases slower than $\mathcal{O}(k)$, the error of the running average sequence will decrease.

V. Illustrative Example

In this section, the convergence results of the asynchronous and inexact DD-DO algorithm are demonstrated by numerical simulations carried out on a six-agent system and a 100-agent system.

A. Overview of Implementation

1) Platform: The simulation is carried on a desktop with Intel i7-10700 CPU and 16-GB memory. The simulation platform is MATLAB 2016B, and commercial solver CPLEX [52] is utilized to solve subproblems with the intermediary toolbox YALMIP [53].

2) Problem: The dual decomposition algorithm is applied to the network utility maximization (NUM) problem [9], [10], [11]. There are n sources (agents) connected by m links, where the agent i wants to maximize its utility $U_i(x_i)$ with respect to the resource transmission rate x_i through the given static path. The system congestion is the maximal transmission capacity b_j of every link j. The NUM problem is formulated as

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) = -\sum_i U_i(x_i) = -\sum_i \left\{C_b - C_a\left(x_i - \overline{x}_i\right)^2\right\}$$
$$\text{s.t. } x_i \in \mathcal{X}_i := \{z \mid \underline{x}_i \leq z \leq \overline{x}_i\}\ \forall i$$
$$A\boldsymbol{x} \leq \boldsymbol{b}$$

where the matrix A indicates the topology of the network as

$$A_{j,i} = \begin{cases} 1, & \text{source } i \text{ goes through link } j \\ 0, & \text{otherwise} \end{cases}$$

and the jth entry of the vector $\boldsymbol{b}$ is the capacity of link j.

3) Asynchrony: To guarantee the asynchrony satisfying Assumption A2, we design the following method to generate the local clocks $\mathcal{K}_i, \mathcal{K}_D$ and the time delays $\delta_{di}^k, \delta_{pi}^k$. We assume that in each local clock $\mathcal{K}_i$ ($\mathcal{K}_D$), the first element is 0 and the difference between any two adjacent elements follows the discrete uniform distribution within $[1, k_0+1]$. The randomized differences are generated by the function *randi* in MATLAB. After generating local clocks, we define the time delays $\delta_{di}^k, k \in \mathcal{K}_i$ as

$$\delta_{di}^k = \max_{\kappa}\ k - \kappa \tag{46a}$$

$$\text{s.t. } \kappa - 1 \in \mathcal{K}_D,\ \kappa \leq k. \tag{46b}$$

From the definition (46), the range of δ_{di}^k is $[0, k_0]$, which satisfies Assumption A2. The minimal value, $\delta_{di}^k = 0$, happens when $k-1 \in \mathcal{K}_D$, and then, $\kappa^* = k$, while the maximal value, $\delta_{di}^k = k_0$, happens when $k \in \mathcal{K}_D$, the previous element in $\mathcal{K}_D$ before k is $k - k_0 - 1$, and then, $\kappa^* = k - k_0$.

The definition of $\delta_{pi}^k, k \in \mathcal{K}_D$ is similar to (46) as

$$\delta_{pi}^k = \max_{\kappa}\ k - \kappa \tag{47a}$$

$$\text{s.t. } \kappa \in \mathcal{K}_i,\ \kappa \leq k. \tag{47b}$$

4) Inexactness: First, we obtain the exact optimal value $D_i(\boldsymbol{\lambda})$ by CPLEX. Then, we attain an inexact solution $\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda})$ by solving the following problem:

$$\widetilde{\boldsymbol{x}}_i(\boldsymbol{\lambda}) \in \arg\max_{\boldsymbol{x}_i \in \mathcal{X}_i}\ \mathcal{L}_i(\boldsymbol{x}_i; \boldsymbol{\lambda}) \tag{48a}$$

$$\text{s.t. } \mathcal{L}_i(\boldsymbol{x}_i; \boldsymbol{\lambda}) - D_i(\boldsymbol{\lambda}) \leq \varepsilon_i \tag{48b}$$

which is the worst-case solution satisfying Assumption A3. If the solution to the problem (48) is not unique, we randomly pick one. Noting that the inexact solution is worst case, the representation of our simulation results is guaranteed.

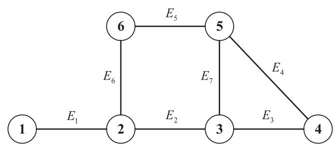


Fig. 2. Topology of the six-agent network.

TABLE I
AGENT DATA

Agent	$\underline{x}_s$	$\overline{x}_s$	C_a	C_b	Path
1	0	5.9	1.8	62.658	$E_1 \to E_2 \to E_7$
2	0	6.6	2.2	95.832	$E_6 \to E_5 \to E_4$
3	0	7.5	2.7	151.875	$E_7 \to E_5 \to E_6 \to E_1$
4	0	4.8	3.5	80.640	$E_3 \to E_2 \to E_6 \to E_5$
5	0	5.4	1.2	34.992	$E_4 \to E_3 \to E_2 \to E_1$
6	0	8.1	0.5	32.805	$E_6 \to E_2 \to E_3 \to E_4$

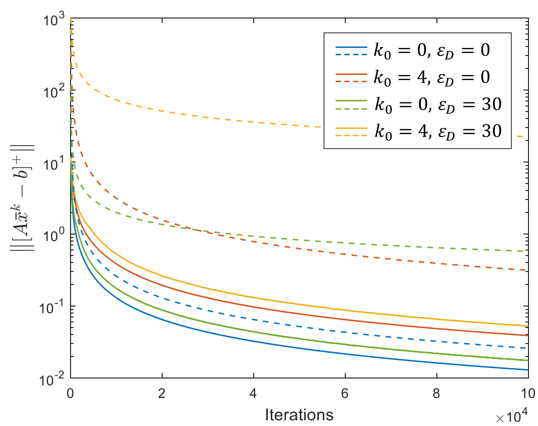


Fig. 3. Violation of the constraints during iterations.

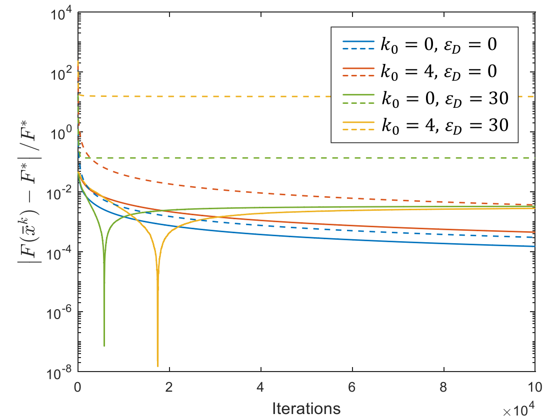


Fig. 4. Relative error of the primal objective during iterations.

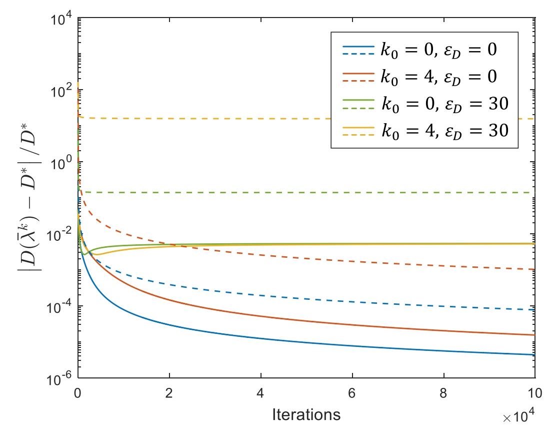


Fig. 5. Relative error of the dual objective during iterations.

B. Simulation on a Six-Agent System

Consider an MAS with six sources (agents) and seven links, whose topology is presented in Fig. 2. The parameters of the agents are provided in Table I. The capacities of the links are as follows:

$$\boldsymbol{b} = [15, 17, 20, 15, 20, 20, 15]^T .$$

The asynchronous and inexact DD-DO algorithm is implemented in the NUM problem of the six-agent system. From (36) in Theorem 9, the sufficient condition to guarantee the convergence is that the step size α satisfies $0 < \alpha < 0.0069$. A small step size may slow down the convergence, whereas a big one may lead to nonconvergence. We choose a moderate $\alpha = 0.004$. Set the parameters $k_0 = 0$ or 4 and $\varepsilon_D = 0$ or 30.

Figs. 3–6 show the curves of the violation of the constraints, the relative errors of the primal and dual objectives, and the deviation of the primal variable during iterations, respectively. The x-axis uses a linear scale, while the y-axis uses a logarithmic scale. In each figure, there are eight curves in blue (synchronous and exact), red (asynchronous and exact), green (synchronous and inexact), and yellow (asynchronous and inexact). The solid lines are the real trajectories generated by Algorithm 2, while the dashed lines are the upper bounds computed by (37)–(40) in Theorem 9.

The violation of the constraints during iterations is shown in Fig. 3. The curves all converge, even if the algorithm is inexact or/and asynchronous. It verifies the convergence result (37) in Theorem 9, i.e., the feasibility is always satisfied. From this figure, we also find that both asynchrony and inexactness increase the violation of the constraints.

Figs. 4 and 5 are the relative errors of the primal and dual objectives during iterations. The comparison of real and dashed curves verifies the convergence results (38) and (39) in Theorem 9, i.e., the value of the objective converges to a neighborhood of the optimal value. It is clear that the inexactness has a remarkable influence on the upper bounds on the ranges of neighborhoods, while the convergence speed slows down from synchronous to asynchronous algorithms. Under the inexact conditions (green and yellow curves in Fig. 4), the difference of the primal objective values $F(\bar{\boldsymbol{x}}^k) - F^*$ turns from negative to positive during iterations. This transition leads to the rapid drop in the exponential coordinate.

The deviation of the primal variable from the optimal solution is shown in Fig. 6. The simulation results verify our

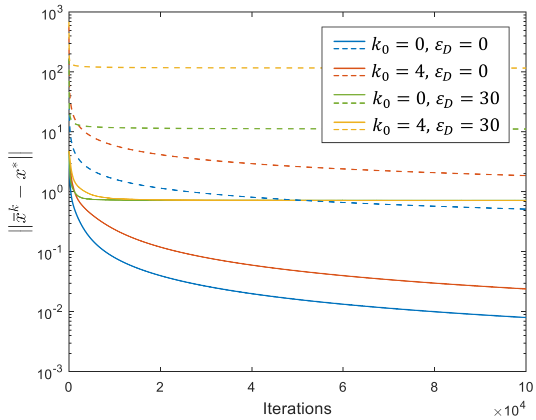


Fig. 6. Deviation of the primal variable during iterations.

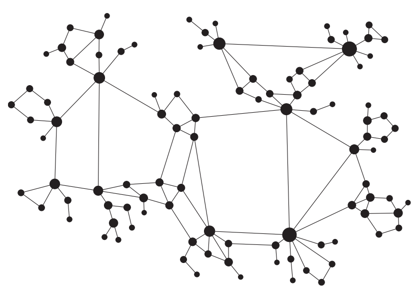


Fig. 7. Topology of the 100-agent network.

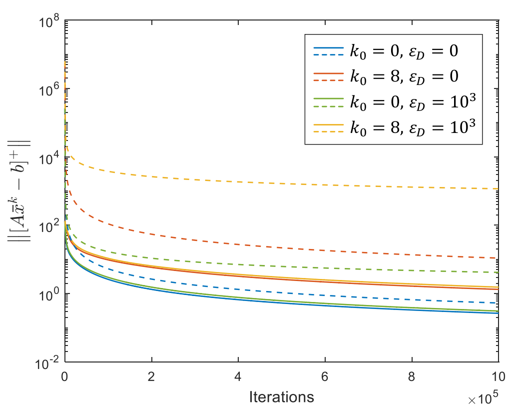


Fig. 8. Violation of the constraints during iterations.

theoretic analysis on the solution accuracy (40) in Theorem 9. If the subproblem solutions are inexact, the sequence $\{\bar{\boldsymbol{x}}^k\}$ converges to a neighborhood of the optimal solution. Otherwise, the deviation decreases to zero and the optimal solution can be obtained. On the other hand, asynchrony slows down the convergence of the primal variable, similarly to the influence on the objective values, as indicated in Theorem 9.

C. Simulation on a 100-Agent System

Consider an MAS with 100 sources (agents) and 133 links, whose topology is presented in Fig. 7. For each agent i, the path

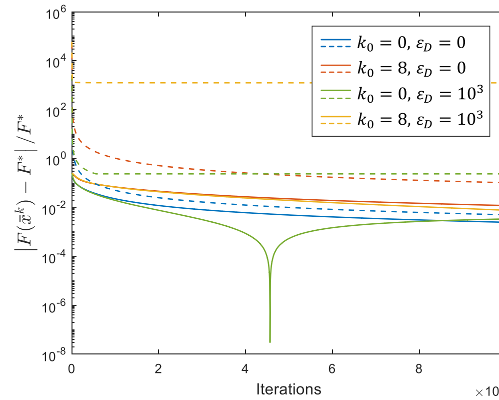


Fig. 9. Relative error of the primal objective during iterations.

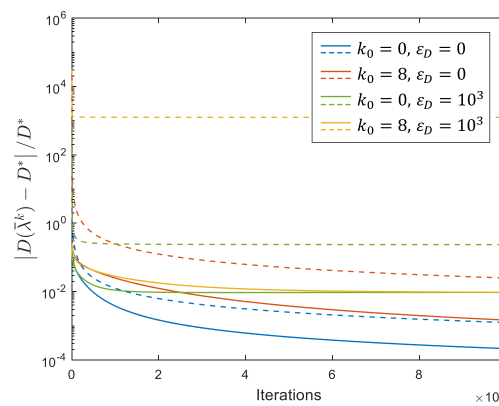


Fig. 10. Relative error of the dual objective during iterations.

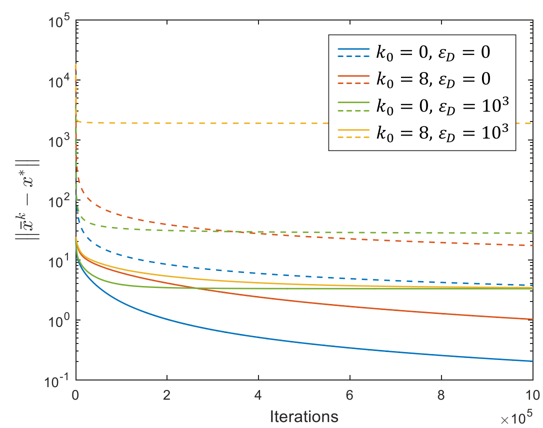


Fig. 11. Deviation of the primal variable during iterations.

of the transmitting source is generated by randomly picking another agent j and finding a shortest path by the Dijkstra algorithm [54]. The capacities of the links are randomly chosen in $[0, 200]$. For each agent i, the lower bound of x_i is $\underline{x}_i = 0$, while the upper bound $\overline{x}_i$ is randomly generated in the range of $[2, 8]$. The parameter C_a of the utility $U_i(x_i)$ is randomly selected in the range of $[3, 5]$, while C_b is assigned to $C_a \overline{x}_i^2$ such that $U_i(0) = 0$.

The asynchronous and inexact DD-DO algorithm is implemented in the NUM problem of the 100-agent system. From (36) in Theorem 9, the sufficient condition to guarantee the convergence is that the step size α satisfies $0 < \alpha < 0.00035$. We choose a moderate $\alpha = 0.0003$. Set the parameters $k_0 = 0$ or 8, $\varepsilon_D = 0$ or 1000.

Figs. 8–11 show the curves of the violation of the constraints, the relative errors of the primal and dual objectives, and the deviation of the primal variable during iterations, respectively. The solid lines are the real trajectories, while the dashed lines are the upper bounds calculated by (37)–(40) in Theorem 9. Fig. 8 indicates the convergence of the violation of the constraints, even if the subproblem solutions are inexact. In Figs. 9 and 10, the convergence of the primal and dual objective values is verified to satisfy (38) and (39) in Theorem 9. Fig. 11 shows that the solution generated by the asynchronous and inexact DD-DO algorithm converges to a certain neighborhood of the optimal solution, which verifies the theoretic result (40). The simulation results also validate that the asynchronous and inexact DD-DO algorithm is applicable for large-scale systems.

VI. Conclusion

In this article, we have studied the DD-DO algorithm in MASs. It is the first time that both the communication asynchrony and the subproblem solution inexactness are considered in the dual decomposition algorithm. Due to the asynchronous communication or nonidentical computation clocks, agents have to solve their subproblems with the previously stored information. Limited by computational accuracy, the subproblem solutions are inexact. We have proved that values of primal and dual objectives converge to some neighborhoods of the optimal values, the solution converges to some neighborhood of the optimal solution, and the violation of the constraints vanishes, all in an $\mathcal{O}(1/\sqrt{k})$ rate of convergence. Our convergence results generalize and unify existing works of dual decomposition algorithms considering only asynchrony or inexactness. Numerical simulation verifies the convergence performance of the asynchronous and inexact DD-DO algorithm.

It is expected that this work could provide useful insights and facilitate the implementations of dual decomposition algorithms in complicated realistic systems, which would inspire more applications in a wide broad of fields.

Appendix A
Proof of Lemma 1

Proof: Given any $\boldsymbol{\lambda} \in \mathbb{R}_+^m$, $\boldsymbol{x}_i(\boldsymbol{\lambda})$ is the optimal solution defined as (7). From the optimality condition [46, Th. 3.24], we have

$$\left\langle \nabla f_i\left(\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right)\right) + A_i^T\boldsymbol{\lambda}, \boldsymbol{y} - \boldsymbol{x}_i\left(\boldsymbol{\lambda}\right)\right\rangle \geq 0 \ \forall \boldsymbol{y} \in \mathcal{X}_i. \tag{A.1}$$

Replacing $\boldsymbol{y}$ in (A.1) with $\boldsymbol{x}_i(\boldsymbol{\mu})\, \forall \boldsymbol{\mu} \in \mathbb{R}_+^m$, we have

$$\left\langle \nabla f_i\left(\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right)\right) + A_i^T\boldsymbol{\lambda}, \boldsymbol{x}_i\left(\boldsymbol{\mu}\right) - \boldsymbol{x}_i\left(\boldsymbol{\lambda}\right)\right\rangle \geq 0. \tag{A.2}$$

Similarly, we have the optimality condition at $\boldsymbol{x}_i(\boldsymbol{\mu})$

$$\left\langle \nabla f_i\left(\boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right) + A_i^T\boldsymbol{\mu}, \boldsymbol{x}_i\left(\boldsymbol{\lambda}\right) - \boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right\rangle \geq 0. \tag{A.3}$$

Flipping the signs of the two terms of $< \cdot, \cdot >$ in (A.2) and adding (A.3), it follows that

$$\underbrace{\left\langle \nabla f_i\left(\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right)\right) - \nabla f_i\left(\boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right), \boldsymbol{x}_i\left(\boldsymbol{\lambda}\right) - \boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right\rangle}_{(\Delta_5)} \leq \underbrace{\left\langle A_i^T\boldsymbol{\mu} - A_i^T\boldsymbol{\lambda}, \boldsymbol{x}_i\left(\boldsymbol{\lambda}\right) - \boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right\rangle}_{(\Delta_6)}.$$

From the strong convexity of $f_i(\cdot)$, we have

$$c_i \left\|\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right) - \boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right\|^2 \leq \Delta_5.$$

From the Cauchy–Schwarz inequality, we have

$$\Delta_6 \leq \left\|A_i\right\| \left\|\boldsymbol{\lambda} - \boldsymbol{\mu}\right\| \left\|\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right) - \boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right\|.$$

No matter if $\|\boldsymbol{x}_i(\boldsymbol{\lambda}) - \boldsymbol{x}_i(\boldsymbol{\mu})\| = 0$ or not, it immediately follows that

$$\left\|\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right) - \boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right\| \leq \frac{\left\|A_i\right\|}{c_i} \left\|\boldsymbol{\lambda} - \boldsymbol{\mu}\right\|$$

which completes the proof. □

Appendix B
Proof of Corollary 2

Proof: 1) Recall that $\mathcal{X}_i$ is nonempty and compact, $f_i(\cdot)$ is continuous over $\mathcal{X}$, and $\boldsymbol{x}_i(\boldsymbol{\lambda})$ is the unique optimal solution to (7). Invoking [44, Prop. 6.1.1], $D_i(\boldsymbol{\lambda})$ is differentiable and its gradient is defined as (10).

To prove the Lipschitz continuity, we have for any $\boldsymbol{\lambda}, \boldsymbol{\mu} \in \mathbb{R}^m$

$$\begin{aligned}\left\|\nabla D_i\left(\boldsymbol{\lambda}\right) - \nabla D_i\left(\boldsymbol{\mu}\right)\right\| &= \left\|A_i\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right) - A_i\boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right\| \\ &\leq \left\|A_i\right\| \left\|\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right) - \boldsymbol{x}_i\left(\boldsymbol{\mu}\right)\right\| \\ &\leq \frac{\left\|A_i\right\|^2}{c_i} \left\|\boldsymbol{\lambda} - \boldsymbol{\mu}\right\|.\end{aligned}$$

2) Similarly, from [44, Prop. 6.1.1], $D(\boldsymbol{\lambda})$ is differentiable and its gradient is defined as (11). To prove the Lipschitz continuity, we have for any $\boldsymbol{\lambda}, \boldsymbol{\mu} \in \mathbb{R}^m$

$$\begin{aligned}&\left\|\nabla D\left(\boldsymbol{\lambda}\right) - \nabla D\left(\boldsymbol{\mu}\right)\right\| \\ &\leq \sum_{i\in\mathcal{N}} \left\|\nabla D_i\left(\boldsymbol{\lambda}\right) - \nabla D_i\left(\boldsymbol{\mu}\right)\right\| \\ &\leq \left(\sum_{i\in\mathcal{N}} \frac{\left\|A_i\right\|^2}{c_i}\right) \left\|\boldsymbol{\lambda} - \boldsymbol{\mu}\right\|\end{aligned}$$

which completes the proof. □

Appendix C
Proof of Lemma 3

Proof: Given $\boldsymbol{\lambda} \in \mathbb{R}_+^m$, from the strong convexity of $\mathcal{L}_i(\cdot;\boldsymbol{\lambda})$ in $\boldsymbol{x}_i$, we have

$$\begin{aligned}\mathcal{L}_i\left(\widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}\right);\boldsymbol{\lambda}\right) \geq\ & \mathcal{L}_i\left(\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right);\boldsymbol{\lambda}\right) + \frac{c_i}{2}\left\|\widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}\right) - \boldsymbol{x}_i\left(\boldsymbol{\lambda}\right)\right\|^2 \\ &+ \underbrace{\left\langle \nabla f_i\left(\boldsymbol{x}_i\left(\boldsymbol{\lambda}\right)\right) + A_i^T\boldsymbol{\lambda}, \widetilde{\boldsymbol{x}}_i\left(\boldsymbol{\lambda}\right) - \boldsymbol{x}_i\left(\boldsymbol{\lambda}\right)\right\rangle}_{(\Delta_7)}.\end{aligned}$$

From (A.1) in Appendix A, (Δ_7) is nonnegative, which completes the proof. □

References

[1] A. Papachristodoulou and A. Jadbabaie, "Delay robustness of nonlinear internet congestion control schemes," *IEEE Trans. Autom. Control*, vol. 55, no. 6, pp. 1421–1427, Jun. 2010.

[2] S. He, J. Chen, D. K. Yau, and Y. Sun, "Cross-layer optimization of correlated data gathering in wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 11, no. 11, pp. 1678–1691, Nov. 2012.

[3] S. Magnússon, C. Enyioha, N. Li, C. Fischione, and V. Tarokh, "Convergence of limited communication gradient methods," *IEEE Trans. Autom. Control*, vol. 63, no. 5, pp. 1356–1371, May 2018.

[4] N. Komodakis, N. Paragios, and G. Tziritas, "MRF optimization via dual decomposition: Message-passing revisited," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.

[5] P. Strandmark and F. Kahl, "Parallel and distributed graph cuts by dual decomposition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 2085–2092.

[6] D. Alkano, J. M. Scherpen, and Y. Chorfi, "Asynchronous distributed control of biogas supply and multienergy demand," *IEEE Trans. Automat. Sci. Eng.*, vol. 14, no. 2, pp. 558–572, Apr. 2017.

[7] S. Huang, Y. Sun, and Q. Wu, "Stochastic economic dispatch with wind using versatile probability distribution and L-BFGS-B based dual decomposition," *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6254–6263, Nov. 2018.

[8] A. Falsone, K. Margellos, and M. Prandini, "A decentralized approach to multi-agent MILPs: Finite-time feasibility and performance guarantees," *Automatica*, vol. 103, pp. 141–150, 2019.

[9] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: A mathematical theory of network architectures," *Proc. IEEE*, vol. 95, no. 1, pp. 255–312, Jan. 2007.

[10] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[11] D. P. Palomar and M. Chiang, "Alternative distributed algorithms for network utility maximization: Framework and applications," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2254–2269, Dec. 2007.

[12] D. Bertsekas and A. Nedic, *Convex Analysis and Optimization*. Belmont, MA, USA: Athena Sci., 2003.

[13] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Old Tappan, NJ, USA: Prentice-Hall, 1989.

[14] Z. Wang et al., "Asynchronous distributed power control of multi-microgrid systems," *IEEE Trans. Control Netw. Syst.*, vol. 7, no. 4, pp. 1960–1973, Dec. 2020.

[15] 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.

[16] S. Magnússon, G. Qu, and N. Li, "Distributed optimal voltage control with asynchronous and delayed communication," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3469–3482, Jul. 2020.

[17] 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.

[18] R. Zhang and J. Kwok, "Asynchronous distributed ADMM for consensus optimization," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1701–1709.

[19] 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, Jun. 2016.

[20] T.-H. Chang, W.-C. Liao, M. Hong, and X. Wang, "Asynchronous distributed ADMM for large-scale optimization—Part II: Linear convergence analysis and numerical performance," *IEEE Trans. Signal Process.*, vol. 64, no. 12, pp. 3131–3144, Jun. 2016.

[21] Y. Zhang, Y. Su, and F. Liu, "Protocol for constrained multi-agent optimization with arbitrary local solvers," in *Proc. IEEE 11th Int. Conf. Inf. Sci. Technol.*, 2021, pp. 148–157.

[22] 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.

[23] Z. Wang et al., "Exponential stability of partial primal-dual gradient dynamics with nonsmooth objective functions," *Automatica*, vol. 129, 2021, Art. no. 109585.

[24] F. Mansoori and E. Wei, "Superlinearly convergent asynchronous distributed network newton method," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 2874–2879.

[25] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Math. Program.*, vol. 146, no. 1-2, pp. 37–75, 2014.

[26] I. Necoara and V. Nedelcu, "Rate analysis of inexact dual first-order methods application to dual decomposition," *IEEE Trans. Autom. Control*, vol. 59, no. 5, pp. 1232–1243, May 2014.

[27] M. Fazlyab, S. Paternain, A. Ribeiro, and V. M. Preciado, "Distributed smooth and strongly convex optimization with inexact dual methods," in *Proc. IEEE Annu. Amer. Control Conf.*, 2018, pp. 3768–3773.

[28] Y. Zhang and M. M. Zavlanos, "Augmented Lagrangian optimization under fixed-point arithmetic," *Automatica*, vol. 122, 2020, Art. no. 109218.

[29] M. Mehyar, D. Spanos, and S. H. Low, "Optimization flow control with estimation error," in *Proc. IEEE Conf. Comput. Commun.*, 2004, vol. 2, pp. 984–992.

[30] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.

[31] B. F. Svaiter, "A partially inexact ADMM with $o(1/n)$ asymptotic convergence rate, $\mathcal{O}(1/n)$ complexity, and immediate relative error tolerance," *Optimization*, vol. 70, pp. 2061–2080, 2020.

[32] A. Patrascu and I. Necoara, "On the convergence of inexact projection primal first-order methods for convex minimization," *IEEE Trans. Autom. Control*, vol. 63, no. 10, pp. 3317–3329, Oct. 2018.

[33] F. Zhang, H. Wang, J. Wang, and K. Yang, "Inexact primal–dual gradient projection methods for nonlinear optimization on convex set," *Optimization*, vol. 69, no. 10, pp. 2339–2365, 2020.

[34] M. Schmidt, N. L. Roux, and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," in *Adv. Neural Inf. Process. Syst.*, vol. 24, 2011.

[35] K. Jiang, D. Sun, and K.-C. Toh, "An inexact accelerated proximal gradient method for large scale linearly constrained convex SDP," *SIAM J. Optim.*, vol. 22, no. 3, pp. 1042–1064, 2012.

[36] N. Bastianello and E. Dall'Anese, "Distributed and inexact proximal gradient method for online convex optimization," in *Proc. Eur. Control Conf.*, 2021, pp. 2432–2437.

[37] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization–I: Algorithm," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2162–2175, Sep. 2013.

[38] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization—Part II: Convergence," *IEEE Trans. Autom. Control*, vol. 58, no. 9, pp. 2176–2188, Sep. 2013.

[39] 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.

[40] S. Magnússon, C. Enyioha, N. Li, C. Fischione, and V. Tarokh, "Communication complexity of dual decomposition methods for distributed resource allocation optimization," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 4, pp. 717–732, Aug. 2018.

[41] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[42] 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.

[43] D. P. Bertsekas, *Nonlinear Programming*, 2nd ed. Nashua, NH, USA: Athena Scientific, 1999.

[44] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. Berlin, Germany: Springer, 2003.

[45] A. Ruszczynski, *Nonlinear Optimization*. Princeton, PA, USA: Princeton Univ. Press, 2006.

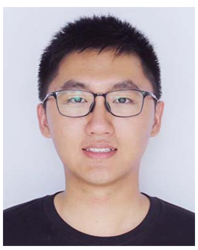
[46] D. Blatt, A. O. Hero, and H. Gauchman, "A convergent incremental gradient method with a constant step size," *SIAM J. Optim.*, vol. 18, no. 1, pp. 29–51, 2007.

[47] H. Yu, "Weak convergence properties of constrained emphatic temporal-difference learning with constant and slowly diminishing stepsize," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 7745–7802, 2016.

[48] Q. Liu, S. Yang, and Y. Hong, "Constrained consensus algorithms with fixed step size for distributed convex optimization over multiagent networks," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4259–4265, Aug. 2017.

[49] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[50] D. Mateos-Núnez and J. Cortés, "Distributed saddle-point subgradient algorithms with Laplacian averaging," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2720–2735, Jun. 2017.
[51] *Encyclopedia of Mathematics*, 2022. [Online]. Available: http://encyclopediaofmath.org/index.php?title=Jensen_inequality&oldid=47465
[52] Cplex, 2022. [Online]. Available: https://www.ibm.com/analytics/cplex-optimizer
[53] Yalmip, 2022. [Online]. Available: https://yalmip.github.io/
[54] E. W. Dijkstra et al., "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.



Yifan Su received the B.Sc. degree in electrical engineering in 2019 from Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree in in electrical engineering.

His research interests include distributed optimization and control, robust dispatch, and demand response in power systems.

Zhaojian Wang (Member, IEEE) received the B.Sc. degree in electrical engineering from Tianjin University, Tianjin, China, in 2013, and the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 2018.

From 2016 to 2017, he was a visiting Ph.D. student with the California Institute of Technology, Pasadena, CA, USA. From 2018 to 2020, he was a Postdoctoral Scholar with Tsinghua University. He is currently an Assistant Professor with Shanghai Jiao Tong University. His research interests include stability analysis, optimal control, microgrid planning, and game-theory-based decision making in energy and power systems.

Dr. Wang is an Associate Editor for IEEE SYSTEMS JOURNALand *IET Renewable Power Generation*.

Ming Cao (Fellow, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Yale University, New Haven, CT, USA, in 2007.

He is currently a Professor of systems and control with the Engineering and Technology Institute, University of Groningen, Groningen, The Netherlands, where he started as an Assistant Professor in 2008. From 2007 to 2008, he was a Postdoctoral Research Associate with the Department of Mechanical and Aerospace Engineering, Princeton University, Princeton, NJ, USA. He worked as a Research Intern with the Mathematical Sciences Department, IBM T. J. Watson Research Center, Armonk, NY, USA, in 2006. His main research interests include autonomous agents and multiagent systems, decision-making dynamics, and complex networks.

Prof. Cao was the 2017 and inaugural recipient of the Manfred Thoma Medal from the International Federation of Automatic Control and the 2016 recipient of the European Control Award sponsored by the European Control Association. He is a Senior Editor for Systems and Control Letters and an Associate Editor for IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS, and IEEE ROBOTICS AND AUTOMATION MAGAZINE.

Mengshuo Jia (Member, IEEE) received the B.E. degree in electrical engineering from North China Electric Power University, Beijing, China, in 2016, and the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, in 2021.

He is currently a Postdoctoral Researcher with the Department of Information Technology and Electrical Engineering, Swiss Federal Institute of Technology, Zürich, Switzerland. His research interests include the probabilistic analysis and stochastic optimization of power systems, in both centralized and distributed manners.

Feng Liu (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in electrical engineering from Tsinghua University, Beijing, China, in 1999 and 2004, respectively.

He is currently an Associate Professor with Tsinghua University. From 2015 to 2016, he was a Visiting Associate with the California Institute of Technology, Pasadena, CA, USA. He is the author/co-author of more than 200 peer-reviewed technical papers and two books, and holds more than 20 issued/pending patents. His research interests include stability analysis, optimal control, robust dispatch and game-theory-based decision making in energy and power systems.

Dr. Liu is an Associated Editor for several international journals, including the IEEE TRANSACTIONS ON SMART GRID, IEEE POWER ENGINEERING LETTERS, IEEE ACCESS, etc. He also served as a Guest Editor for the IEEE TRANSACTIONS ON ENERGY CONVERSION.