A Distributed Augmented Lagrangian Method over Stochastic Networks for Economic Dispatch of Large-Scale Energy Systems

Wicak Ananduta, Carlos Ocampo-Martinez, and Angelia Nedić

***Abstract*—In this paper, we propose a distributed model predictive control (MPC) scheme for economic dispatch of energy systems with a large number of active components. The scheme uses a distributed optimization algorithm that works over random communication networks and asynchronous updates, implying the resiliency of the proposed scheme with respect to communication problems, such as link failures, data packet drops, and delays. The distributed optimization algorithm is based on the augmented Lagrangian approach, where the dual of the considered convex economic dispatch problem is solved. Furthermore, in order to improve the convergence speed of the algorithm, we adapt Nesterov's accelerated gradient method and apply the warm start method to initialize the variables. We show through numerical simulations of a well-known case study the performance of the proposed scheme.**

***Index terms* – economic dispatch, multi-agent optimization, model predictive control, stochastic time-varying network**

I. Introduction

The aim of achieving clean, efficient, and sustainable energy production and consumption [1] drives a rapid installation of small-scale production units, particularly those that use renewable energy sources, and storage units [2]. Based on these developments, distributed optimization and control approaches are perceived to be suitable for future energy networks [2], [3]. The advantages of distributed approaches over the centralized counterpart include better cybersecurity, communication efficiency, scalability, and privacy [3].

This paper focuses on the energy management problem of large-scale systems with a large number of active components, which are previously mentioned. Specifically, we consider a distributed model predictive control (MPC) approach, which allows us to include forecasts of uncontrollable generation and loads, the slow dynamics of the storage units, and hard operational constraints [4], [5]. Moreover, the decisions (set points of the active components) are updated at each time step based on the latest measurement, making the decisions more robust against uncertainty [4].

Various distributed methods to solve economic dispatch problems can be found, especially in the recent literature, e.g., [6]–[10], or see [3], [11] for a survey. One of the key differences of those methods is in the way of decomposing the problem. In this regard, many distributed methods are based on the (augmented) Lagrangian duality theory [12]. Nevertheless, typically distributed methods, including those that are based on the Lagrangian duality, are iterative and require the exchange of some necessary information at each iteration. Due to its reliance on communication, it is important to ensure the resiliency of the distributed method in case some communication problems arise, such as link failures, delays, or data packet drops.

We can model the above communication problems as having a time-varying communication network and asynchronous updates. However, there is only a few existing work that discusses this problem in distributed economic dispatch problem, e.g., [13]–[15]. In [13], the time-varying network is assumed to be strongly connected with a positive probability. Moreover, in [14], [15], the time-varying communication network is assumed to be jointly (strongly) connected. Additionally, [13]–[15] do not model asynchronous updates. Differently, [16], [17] consider stochastic communication networks for consensus and constrained optimization problems, respectively.

On the other hand, an MPC-based scheme requires solving an optimization problem online. Therefore, having a fast distributed algorithm as well as implementing heuristics, such as warm start method, become important. One fast gradient-based optimization method is proposed by Nesterov in [18]. Some adaptations of Nesterov's accelerated gradient method have also been proposed to solve the dual of some problems, e.g., MPC [19], [20] and network resource allocation problems [21], resulting in distributed algorithms. However, these papers consider problems with a strongly convex primal cost function, which does not hold in the problem that we consider. In the class of augmented Lagrangian methods, the papers [22], [23] present fast alternating direction method of multipliers (ADMM) algorithms, which implement Nesterov's gradient steps. However, the problem considered is limited to that with only an equality constraint whereas an economic dispatch problem typically also has inequality constraints. Moreover, these ADMM-based distributed methods require the agents to perform their updates sequentially and do not consider time-varying communication networks.

In this work, as the main contribution, we propose two novel distributed optimization algorithms for MPC-based economic dispatch of large-scale energy systems. The algorithms have parallel implementation and are based on an augmented La-

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 675318 (INCITE).

W. Ananduta is with Delft Center for Systems and Control (DCSC), TU Delft, Netherlands. C. Ocampo-Martinez is with Institut de Robòtica i Informàtica Industrial (CSIC-UPC), Barcelona, Spain. A. Nedić is with School of Electrical, Computer and Energy Engineering, Arizona State University. (emails: `w.ananduta@tudelft.nl`, `carlos.ocampo@upc.edu`, `angelia.nedich@asu.edu`).

grangian method, firstly introduced in [24], [25]. Furthermore, the proposed algorithms work over stochastic communication networks, thus they are robust against some communication problems. The first algorithm is an adaptation of the method presented in [26] for large-scale economic dispatch problems whereas the second method is an accelerated version of the first based on Nesterov's approach. We show through a numerical study the efficacy of these algorithms using a well-known benchmark case. As MPC-based economic dispatch methods, both algorithms produce a comparable performance under an unreliable communication network. However, indeed the accelerated algorithm converges faster than the unaccelerated one, highlighting its advantage.

The paper proceeds as follows. Section II presents the model of a large-scale energy system considered in this paper, the network partitioning, and the MPC-based economic dispatch problem of the system. Then, Section III provides the proposed distributed method based on the augmented Lagrangian approach, whereas Section IV discusses the techniques considered to improve the convergence speed of the proposed method and presents the accelerated version of the proposed method. Furthermore, Section V is devoted to showing the performance of the proposed method and its accelerated version through numerical simulations using a well-known benchmark case study. Finally, Section VI concludes the paper with some remarks and a discussion of future work.

Notation

The sets of real and integer numbers are denoted by $\mathbb{R}$ and $\mathbb{Z}$, respectively. For any $a \in \mathbb{R}$, $\mathbb{R}_{\geq a} = \{b \in \mathbb{R} : b \geq a\}$. Similar definition is also used for $\mathbb{Z}_{\geq a}$, and the strict inequality cases. The set of real vectors with dimension n is denoted by $\mathbb{R}^n$. The inner product of vectors $x, y \in \mathbb{R}^n$ is denoted by $\langle x, y\rangle$. The Euclidean norm of vector x is denoted by $\|x\|_2$. The all-ones vector with the size of n is denoted by $\mathbb{1}_n$, whereas the identity matrix with the size $n \times n$ is denoted by I_n. Finally, $\mathrm{col}(\cdot)$ constructs a column vector from its arguments, whereas the block-diagonal operator, which constructs a block diagonal matrix of the arguments, is denoted by $\mathrm{blkdiag}(\cdot)$.

II. MPC-BASED ECONOMIC DISPATCH PROBLEM

In this section, a graph-based model for large-scale electrical networks and its distributed MPC-based economic dispatch problem are formulated.

A. Model of the network

Let a large-scale electrical network be represented by an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N} = \{1, 2, \dots, n\}$ denotes the set of busses (nodes) and $\mathcal{E} \subseteq \mathcal{N} \times \mathcal{N}$ denotes the set of links that physically connect the nodes. In this system, each node might have load, storage unit, and generation unit. Moreover, each pair of nodes that are connected, e.g., nodes i and j such that $\{i, j\} \in \mathcal{E}$, might also exchange power between each other. Some nodes might also be connected with the main grid. We denote by $\mathcal{N}^{\mathrm{dg}} \subseteq \mathcal{N}$, $\mathcal{N}^{\mathrm{st}} \subseteq \mathcal{N}$, and $\mathcal{N}^{\mathrm{mg}} \subseteq \mathcal{N}$ the set of nodes that have dispatchable generation units, storage units, and are connected with the main grid, respectively.

We denote by $d_i \in \mathbb{R}$, for each $i \in \mathcal{N}$, the difference between the load and power generated by non-dispatchable generation units at node i. Note that if $d_i > 0$, the load is larger than the generated power. Furthermore, we define the *local decision* of node i at time step t by $u_{i,t} = \mathrm{col}(u_{i,t}^{\mathrm{st}}, u_{i,t}^{\mathrm{dg}}, u_{i,t}^{\mathrm{mg}}) \in \mathbb{R}^{n_i^{\mathrm{l}}}$, where $u_{i,t}^{\mathrm{st}} \in \mathbb{R}$, $u_{i,t}^{\mathrm{dg}} \in \mathbb{R}^{n_i^{\mathrm{dg}}}$, and $u_{i,t}^{\mathrm{mg}} \in \mathbb{R}_{\geq 0}$ are the set points of the power delivered from/to the storage unit, the power produced by the dispatchable generation units, and the power imported from the third party, respectively. Note that $n_i^{\mathrm{l}} = 2 + n_i^{\mathrm{dg}}$. On the other hand, let $\mathcal{N}_i := \{j : \{i, j\} \in \mathcal{E}\}$ be the set of neighbors of node i. Then, we denote by $v_{i,t}^j \in \mathbb{R}$ the power exchanged between nodes i and $j \in \mathcal{N}_i$. Note that $v_{i,t}^j$ is a decision of node i, whereas the decision of node j is denoted by $v_{j,t}^i$ and it must hold that

$$v_{i,t}^j + v_{j,t}^i = 0, \quad \forall j \in \mathcal{N}_i,\ \forall i \in \mathcal{N}. \tag{1}$$

Additionally, locally, each node must also meet a power balance, which is reflected by

$$\mathbb{1}_{n_i^{\mathrm{l}}}^\top u_{i,t} + \sum_{j \in \mathcal{N}_i} v_{i,t}^j = d_{i,t}. \tag{2}$$

Now, we describe the operational constraints of each component in the network as follows.

1) *Dispatchable generation units.* The production capacity constraints of these units are stated as follows:

$$u_i^{\mathrm{dg,min}} \leq u_{i,t}^{\mathrm{dg}} \leq u_i^{\mathrm{dg,max}}, \quad \forall i \in \mathcal{N}, \tag{3}$$

where $u_i^{\mathrm{dg,min}}$ and $u_i^{\mathrm{dg,max}}$ denote the minimum and maximum power that can be generated by the dispatchable generation units in node i, respectively. Note that, for $i \notin \mathcal{N}^{\mathrm{dg}}$, $u_i^{\mathrm{dg,min}} = u_i^{\mathrm{dg,max}} = 0$.

2) *Storage units.* The capability of storing energy is modeled by discrete-time integrator dynamics, i.e.,

$$x_{i,t+1} = a_i x_{i,t} + b_i u_{i,t}^{\mathrm{st}}, \quad \forall i \in \mathcal{N}^{\mathrm{st}}, \tag{4}$$

where x_i denotes the state of charge (SoC) of the storage unit. The scalar $a_i \in (0, 1]$ denotes the efficiency of the storage whereas $b_i = -\frac{T_s}{e_i^{\mathrm{cap}}}$, where T_s denotes the sampling time of the system and e_i^{cap} denotes the energy capacity of the storage unit. Moreover, the SoC and the power delivered to/from the storage unit are also constrained as follows:

$$x_i^{\min} \leq x_{i,t+1} \leq x_i^{\max}, \quad \forall i \in \mathcal{N}^{\mathrm{st}}, \tag{5}$$

$$-u_i^{\mathrm{ch}} \leq u_{i,t}^{\mathrm{st}} \leq u_i^{\mathrm{dh}}, \quad \forall i \in \mathcal{N}^{\mathrm{st}}, \tag{6}$$

$$u_{i,t}^{\mathrm{st}} = 0, \quad \forall i \notin \mathcal{N}^{\mathrm{st}}, \tag{7}$$

where $x_i^{\min}, x_i^{\max} \in [0, 1]$ denote the upper and lower limits of SoC whereas $u_i^{\mathrm{ch}}, u_i^{\mathrm{dh}} \in \mathbb{R}_{\geq 0}$ denote the maximum charging and discharging power of the storage unit.

3) *Imported power.* The amount of power that can be imported from the main grid is limited as follows:

$$0 \leq u_{i,t}^{\mathrm{mg}} \leq u_i^{\mathrm{mg,max}}, \quad \forall i \in \mathcal{N}^{\mathrm{mg}}, \tag{8}$$

$$u_{i,t}^{\mathrm{mg}} = 0, \quad \forall i \notin \mathcal{N}^{\mathrm{mg}}, \tag{9}$$

where $u_i^{\mathrm{mg,max}} \in \mathbb{R}_{>0}$ denotes the maximum power that can be imported.

4) *Exchanged power.* The amount of power that can be exchanged between two neighboring nodes is also limited as follows:

$$-v_i^{j,\max} \leq v_{i,t}^j \leq v_i^{j,\max}, \quad \forall j \in \mathcal{N}_i,\ \forall i \in \mathcal{N}, \tag{10}$$

where $v_i^{j,\max}$ is the maximum power that can be delivered to/from node i through link $\{i,j\}$.

B. Distributed MPC-based economic dispatch problem

In a distributed control scheme, it is assumed that there exists m computational units (controllers), where $m \ll n$. These controllers cooperatively manage the operation of the network $\mathcal{G}$. In this regard, suppose that the network is decomposed into m non-overlapping partitions, defined by the set $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_p : p = 1, 2, \dots, m\}$ where the definition of partitions is given in Definition 1. Note that the partitioning of the network can be done based on the network design, market design, or the energy contracts [7], [27]–[31].

Definition 1 (Non-overlapping partitions): Let $\mathcal{M}_p$, for $p = 1, 2, \dots, m$, be a non-empty subset of $\mathcal{N}$. Then, the set $\boldsymbol{\mathcal{M}} = \{\mathcal{M}_p : p = 1, 2, \dots, m\}$ defines m non-overlapping partitions of the network $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, if $\bigcup_{p=1}^m \mathcal{M}_p = \mathcal{N}$ and $\mathcal{M}_p \cap \mathcal{M}_q = \emptyset$, for any $\mathcal{M}_p, \mathcal{M}_q \in \boldsymbol{\mathcal{M}}$ and $p \neq q$. □

We consider that each partition is a subsystem controlled by a local controller. In this regard, we describe the network of subsystems by the undirected graph $\mathcal{G}^{\mathrm{s}} = (\mathcal{P}, \mathcal{E}^{\mathrm{s}})$. The set $\mathcal{P} = \{1, 2, \dots, m\}$ denotes the index set of the subsystems and $\mathcal{E}^{\mathrm{s}} \subseteq \mathcal{P} \times \mathcal{P}$ denotes the set of links among the subsystems. Furthermore, if there exists at least one node in subsystem p connected to a node in subsystem q, then $\{p, q\} \in \mathcal{E}^{\mathrm{s}}$. Therefore, let $\mathcal{N}_p^{\mathrm{s}}$ denote the set of neighbors of subsystem $p \in \mathcal{P}$, i.e., $\mathcal{N}_p^{\mathrm{s}} := \{q \in \mathcal{P} : \exists \{i, j\} \in \mathcal{E}, i \in \mathcal{M}_p, j \in \mathcal{M}_q, p \neq q\}$. Figure 1 illustrates how the network of subsystems of a small-scale energy system is formed. Additionally, since the partitions are non-overlapping, each node is associated with one subsystem. In this regard, let $\phi : \mathcal{N} \to \mathcal{P}$ be the function that gives the subsystem index with which a node is associated, i.e., $\phi(i) := p$, where $p \in \mathcal{P}$ is such that $i \in \mathcal{M}_p$.

We are now in the position to state the optimization problem that underlies the MPC-based economic dispatch scheme. To that end, first, we define the concatenated decision variables over a certain time horizon h, i.e., for all $\tau \in \mathcal{T} = \{t, t+1, \dots, t+h-1\}$, with bold symbols, i.e., $\boldsymbol{u}_{i,t} := \mathrm{col}(\{u_{i,\tau}\}_{\tau \in \mathcal{T}}) \in \mathbb{R}^{n_i^{\mathrm{l}} h}$, $\boldsymbol{v}_{i,t}^j := \mathrm{col}(\{v_{i,\tau}^j\}_{\tau \in \mathcal{T}}) \in \mathbb{R}^h$, and $\boldsymbol{v}_{i,t} = \mathrm{col}(\{\boldsymbol{v}_{i,t}^j\}_{j \in \mathcal{N}_i}) \in \mathbb{R}^{|\mathcal{N}_i| h}$. Therefore, at time step t, the controllers must cooperatively solve

$$\underset{\{(\boldsymbol{u}_{i,t}, \boldsymbol{v}_{i,t})\}_{i \in \mathcal{N}}}{\text{minimize}} \ \sum_{i \in \mathcal{N}} \left(f_i^{\mathrm{l}}(\boldsymbol{u}_{i,t}) + f_i^{\mathrm{c}}(\boldsymbol{v}_{i,t}) \right) \tag{11a}$$

$$\text{s.t. } (\boldsymbol{u}_{i,t}, \boldsymbol{v}_{i,t}) \in \mathcal{L}_{i,t}, \ \forall i \in \mathcal{N}, \tag{11b}$$

$$\boldsymbol{v}_{i,t}^j + \boldsymbol{v}_{j,t}^i = 0, \quad \forall j \in \mathcal{N}_i,\ \forall i \in \mathcal{N}, \tag{11c}$$

where $f_i^{\mathrm{l}}(\boldsymbol{u}_{i,t})$ and $f_i^{\mathrm{c}}(\boldsymbol{v}_{i,t})$ in (11a) are the cost functions associated with node i and the set $\mathcal{L}_{i,t}$, for each $i \in \mathcal{N}$, is the local polyhedral constraint set defined such that (2)-(10), for all times $t, t+1, \dots, t+h-1$, hold.

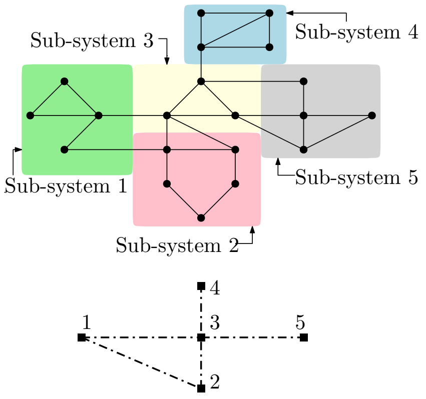

Fig. 1. An example of a small-scale energy network. Top figure shows graph $\mathcal{G}$ where the nodes in $\mathcal{N}$ are depicted as dots and the links in $\mathcal{E}$ are depicted as solid lines. The graph $\mathcal{G}$ is partitioned into five subsystems. Bottom figure shows graph $\mathcal{G}^{\mathrm{s}}$ where the subsystems in $\mathcal{P}$ are depicted as filled squares and the links in $\mathcal{E}^{\mathrm{s}}$ are depicted as dash-dotted lines.

Remark 1: Although we specify that $\mathcal{L}_{i,t}$ is defined by (2)-(10), one might consider additional operational constraints for some components, e.g., those that are inter-temporal. For instance, the rate of power produced by either the generation units or the storage units, which still results in polyhedral $\mathcal{L}_{i,t}$. □

Assumption 1: For each $i \in \mathcal{N}$, the functions $f_i^{\mathrm{l}} : \mathbb{R}^{n_i^{\mathrm{l}} h} \to \mathbb{R}$ and $f_i^{\mathrm{c}} : \mathbb{R}^{|\mathcal{N}_i| h} \to \mathbb{R}$ are continuously differentiable and convex. Moreover, $f_i^{\mathrm{l}}(\boldsymbol{u}_{i,t})$ is strongly convex. □

Assumption 2: The feasible set of Problem (11), for any $t \in \mathbb{Z}_{\geq 0}$, is non-empty. □

Furthermore, we let Assumptions 1 and 2 hold. Assumption 1 is commonly considered [3], [5], [6] and typically f_i^{l} is a strongly convex quadratic function, i.e., $\boldsymbol{u}_{i,t}^\top Q_i \boldsymbol{u}_{i,t} + q_i^\top \boldsymbol{u}_{i,t}$, with positive definite Q_i, which is typically used to approximate the operational cost of the distributed generation units, penalize the usage of storage units as well as the cost of importing power from the main grid [8], [10] whereas $f_i^{\mathrm{c}} = 0$. Moreover, Assumption 2 is a technical assumption to ensure the existence of an optimal solution to Problem (11).

In an MPC scheme, Problem (11) is solved at each time step t, and only the decisions associated to the current time step t are applied, as summarized in Algorithm 1.

Algorithm 1 MPC-based economic dispatch

```
Iteration: For each $t \in \mathbb{Z}_{\geq 0}$,
  1) Compute $\boldsymbol{u}_{i,t}$ and $\boldsymbol{v}_{i,t}$, for all $i \in \mathcal{N}$, by solving Problem (11).
  2) Implement the decisions of the current time step $t$, i.e., $u_{i,t}$ and $v_{i,t}$, for all $i \in \mathcal{N}$.
  3) Measure the states $x_{i,t+1}$, for all $i \in \mathcal{N}^{\mathrm{st}}$.
```

Remark 2: We consider a nominal problem in (11) and do not specifically deal with uncertainties from power demands and non-dispatchable generation. Nevertheless, in order to complement this work with a robustification method, we must ensure when the problem is robustified, it remains convex and the assumptions we consider are satisfied. □

III. Distributed augmented Lagrangian method

This section describes a distributed method to solve Problem (11) that works under imperfect communication. We drop the time index t for convenience as the proposed algorithm solves Problem (11) at a fixed t (step 1 of Algorithm 1). Furthermore, we also introduce the iteration index k used by the algorithm. Prior to describing the algorithm, we model imperfect communication as random processes.

A. Stochastic communication process

We consider that not all subsystems might perform the updates at each iteration (asynchronicity) and the communication links between neighboring subsystems are time-varying. Therefore, we represent the communication network of the local controllers by the undirected graph $\mathcal{G}^{\mathrm{c}}(k) = (\mathcal{A}(k), \mathcal{E}^{\mathrm{c}}(k))$, where $\mathcal{A}(k) \subset \mathcal{P}$ and $\mathcal{E}^{\mathrm{c}}(k) \subseteq \mathcal{E}^{\mathrm{s}}$ denote the set of subsystems and the set of communication links that are active at iteration $k-1$, respectively, i.e., $p \in \mathcal{A}(k)$ means that subsystem p is active, whereas $\{p, q\} \in \mathcal{E}^{\mathrm{c}}(k)$ means that the communication link between subsystems p and q is active at $k-1$. Furthermore, we denote the set of active neighbors of subsystem $p \in \mathcal{P}$ by $\mathcal{A}_p(k)$, i.e., $\mathcal{A}_p(k) = \{q \in \mathcal{A}(k) : \{p, q\} \in \mathcal{E}^{\mathrm{c}}(k)\}$. Finally, the communication network is modeled as a random graph as follows [16].

Assumption 3 (Random network): The set $\mathcal{E}^{\mathrm{c}}(k) \subseteq \mathcal{E}^{\mathrm{s}}$ is a random variable that is independent and identically distributed across iterations. Furthermore, any communication link between two coupled subsystems p and q, where $\{p, q\} \in \mathcal{E}^{\mathrm{s}}$, is active with a positive probability denoted by β_{pq}, i.e., $\mathbb{P}(\{p, q\} \in \mathcal{E}^{\mathrm{c}}(k)) = \beta_{pq} > 0$. □

Assumption 4 (Asynchronous update): The set $\mathcal{A}(k) \subseteq \mathcal{P}$ is a random variable that is independent and identically distributed across iterations. Moreover, a subsystem $p \in \mathcal{P}$ is active at iteration k with a positive probability denoted by γ_p, i.e., $\mathbb{P}(p \in \mathcal{A}(k)) = \gamma_p > 0$. □

B. Stochastic DAL algorithm

The proposed distributed method, as presented in Algorithm 2, is based on solving the dual problem associated with the augmented problem of (11), where the augmentation terms and the Lagrange multipliers, denoted by $\boldsymbol{\lambda}_i^j \in \mathbb{R}^h$, for all $j \in \mathcal{N}_i$ and $i \in \mathcal{N}$, are associated with the coupling constraints. A detailed explanation of the derivation is provided in [26]. Note that η_i^j, for all $j \in \mathcal{N}_i$ and $i \in \mathcal{N}$, are step sizes used to perform the gradient step ascent of the Lagrange multipliers, whereas the auxiliary variable $\boldsymbol{z}_i^j \in \mathbb{R}^h$, for each $j \in \mathcal{N}_i$ and $i \in \mathcal{N}$, is used by subsystem $\phi(i)$ to track the information of the neighbor nodes of node i, i.e., $\boldsymbol{v}_j^i$, for all $j \in \mathcal{N}_i$.

Now, we explain how each iteration in Algorithm 2 works.

- *Primal variable updates:* Each subsystem $p \in \mathcal{P}$ updates $\boldsymbol{u}_i(k+1)$ and $\boldsymbol{v}_i(k+1)$, for all $i \in \mathcal{M}_p$, by first solving the local optimization problem (12), where $\hat{\boldsymbol{v}}_i(k)$ denotes an auxiliary variable used to update $\boldsymbol{v}_i(k+1)$. Then, in step 2, $\boldsymbol{v}_i^j(k+1)$ is updated by applying a convex combination of $\hat{\boldsymbol{v}}_i^j(k)$ and $\boldsymbol{v}_i^j(k)$. In this step, the coupled variable $\boldsymbol{v}_i^j$ is only updated with the convex combination step if $\phi(i)$ and $\phi(j)$ can exchange information, i.e., both $\phi(i)$ and $\phi(j)$, when $\phi(i) \neq \phi(j)$, as well as the link $\{\phi(i), \phi(j)\}$ are active. Note that if nodes $i, j \in \mathcal{M}_p$ and $p \in \mathcal{A}(k+1)$, then $\boldsymbol{v}_i^j$ and $\boldsymbol{v}_j^i$ are updated.
- *Information exchange:* In step 3, each active subsystem $p \in \mathcal{A}(k+1)$ exchanges information with its active neighbors $q \in \mathcal{A}_p(k+1)$.
- *Dual variable updates:* In step 4, the dual variables are updated with a gradient ascent step. Similarly to the update of $\boldsymbol{v}_i^j$, $\boldsymbol{\lambda}_i^j$ is also updated depending whether both subsystems $\phi(i)$ and $\phi(j)$ as well as the communication link $\{\phi(i), \phi(j)\}$ are active. Note that if $\phi(i) = \phi(j)$ and $\phi(i) \in \mathcal{A}(k+1)$, then $\boldsymbol{\lambda}_i^j$ is updated with the gradient step.

Algorithm 2 Stochastic DAL method

Initialization: For each node $i \in \mathcal{N}$, $\boldsymbol{v}_i(0) \in \mathbb{R}^{|\mathcal{N}_i|h}$. Moreover, $\boldsymbol{z}_i^j(0) = \boldsymbol{v}_j^i(0)$, $\boldsymbol{\lambda}_i^j(0) = \boldsymbol{\lambda}_j^i(0) \in \mathbb{R}^h$, and $\eta_i^j = \eta_j^i \in (0, \frac{1}{4})$, for all $j \in \mathcal{N}_i$ and $i \in \mathcal{N}$.

Iteration: For each subsystem $p \in \mathcal{A}(k+1)$,

1) Update $(\boldsymbol{u}_i(k+1), \hat{\boldsymbol{v}}_i(k))$, for all $i \in \mathcal{M}_p$, according to

$$\begin{aligned}&\{(\boldsymbol{u}_i(k+1), \hat{\boldsymbol{v}}_i(k))\}_{i \in \mathcal{M}_p}\\ &= \arg \min_{\{(\boldsymbol{u}_i, \boldsymbol{v}_i) \in \mathcal{L}_i\}_{i \in \mathcal{M}_p}} \sum_{i \in \mathcal{M}_p} \Bigg(f_i^{\mathrm{l}}(\boldsymbol{u}_i) + f_i^{\mathrm{c}}(\boldsymbol{v}_i) + \\ &+ \sum_{j \in \mathcal{N}_i} \left(2\langle \boldsymbol{\lambda}_i^j(k), \boldsymbol{v}_i^j \rangle + \|\boldsymbol{v}_i^j + \boldsymbol{z}_i^j(k)\|_2^2 \right) \Bigg). \end{aligned} \tag{12}$$

2) Update $\boldsymbol{v}_i^j(k+1)$, for all $j \in \mathcal{N}_i$ and $i \in \mathcal{M}_p$, as follows:

$$\boldsymbol{v}_i^j(k+1) = \begin{cases} \eta_i^j \hat{\boldsymbol{v}}_i^j(k) + \left(1 - \eta_i^j\right) \boldsymbol{v}_i^j(k), \\ \qquad\qquad \text{if } \phi(j) \in \mathcal{A}_p(k+1) \cup \{p\}, \\ \boldsymbol{v}_i^j(k), \quad \text{otherwise.} \end{cases} \tag{13}$$

3) For each $j \in \mathcal{N}_i \backslash \mathcal{M}_p$ such that $\phi(j) \in \mathcal{A}_p(k+1)$ and $i \in \mathcal{M}_p$, send $\boldsymbol{v}_i^j(k+1)$ to and receive $\boldsymbol{v}_j^i(k+1)$ from subsystem $\phi(j)$.
4) Update the auxiliary and dual variables $\boldsymbol{z}_i(k+1)$ and $\boldsymbol{\lambda}_i(k+1)$, for all $i \in \mathcal{M}_p$, according to

$$\boldsymbol{z}_i^j(k+1) = \begin{cases} \boldsymbol{v}_j^i(k+1), & \text{if } \phi(j) \in \mathcal{A}_p(k+1) \cup \{p\}, \\ \boldsymbol{z}_i^j(k), & \text{otherwise,} \end{cases} \tag{14}$$

$$\boldsymbol{\lambda}_i^j(k+1) = \begin{cases} \boldsymbol{\lambda}_i^j(k) + \eta_i^j \left(\boldsymbol{v}_i^j(k+1) + \boldsymbol{z}_i^j(k+1) \right), \\ \qquad\qquad \text{if } \phi(j) \in \mathcal{A}_p(k+1) \cup \{p\}, \\ \boldsymbol{\lambda}_i^j(k), \quad \text{otherwise.} \end{cases} \tag{15}$$

For subsystem $p \notin \mathcal{A}(k+1)$, we have $\boldsymbol{u}_i(k+1) = \boldsymbol{u}_i(k)$, $\boldsymbol{v}_i(k+1) = \boldsymbol{v}_i(k)$, $\boldsymbol{z}_i(k+1) = \boldsymbol{z}_i(k)$, and $\boldsymbol{\lambda}_i(k+1) = \boldsymbol{\lambda}_i(k)$ for all $i \in \mathcal{M}_p$.

In addition, it is important to mention that, based on Assumptions 3 and 4, the variables $\boldsymbol{v}_i^j$, $\boldsymbol{z}_i^j$ and $\boldsymbol{\lambda}_i^j$, for each $j \in \mathcal{N}_i$

and $i \in \mathcal{N}$, are updated with a positive probability, denoted by α_{ij}, which is defined as follows:

$$\alpha_{ij} = \begin{cases} \beta_{pq}\gamma_p\gamma_q, & \text{if } p \neq q, \\ \gamma_p, & \text{otherwise}, \end{cases} \tag{16}$$

where $p = \phi(i)$ and $q = \phi(j)$.

Theorem 1 ([26, Theorem 1]): Let Assumptions 1-4 hold. Furthermore, let the sequences $\{(\boldsymbol{u}_i(k), \boldsymbol{v}_i(k))\}$, for all $i \in \mathcal{N}$, be generated by Algorithm 2. Then, with probability 1,

a. (Feasibility) $\lim_{k\to\infty} \|\boldsymbol{v}_i^j(k) + \boldsymbol{v}_j^i(k)\|_2^2 = 0$, for all $j \in \mathcal{N}_i$ and $i \in \mathcal{N}$,
b. (Primal variable convergence) $\lim_{k\to\infty} \boldsymbol{u}_i(k) = \boldsymbol{u}_i^\star$ and $\lim_{k\to\infty} \boldsymbol{v}_i(k) = \boldsymbol{v}_i^\star$, for all $i \in \mathcal{N}$, where $\{\boldsymbol{u}_i^\star, \boldsymbol{v}_i^\star\}_{i\in\mathcal{N}}$ is an optimal solution to Problem (11). □

Proof: A complete proof is provided in [26], with a minor modification on the definition of α_{ij} as defined in (16). ■

IV. Improving convergence speed

This section provides two techniques that can be used to reduce the number of iterations of the stochastic DAL method.

A. Nesterov's accelerated method

In [18], Nesterov proposes an optimal gradient method that has convergence rate of $\mathcal{O}(\frac{1}{k^2})$ for a convex smooth optimization. The main idea of this accelerated gradient method is that the gradient step is taken from a smartly chosen interpolated point of the last two iterations. This method has been further extended for non-smooth cases in [32]. Furthermore, a generalization of the accelerated method and its variants and a unifying framework to analyze them are provided in [33].

To show the accelerated gradient method, we consider the following problem:

$$\underset{u}{\text{minimize}}\ f(u),$$

where $u \in \mathbb{R}^{n_u}$ and $f(u) : \mathbb{R}^{n_u} \to \mathbb{R}$ is a differentiable convex function wih Lipschitz continuous gradient. Moreover, we denote by ω the Lipschitz constant of the gradient of $f(u)$, denoted by $\nabla f(u)$. The accelerated gradient method for the preceding problem, as presented in [23], is shown in Algorithm 3. As can be seen, $\hat{u}(k)$ is the interpolated point used to perform the gradient step.

Algorithm 3 Nesterov's accelerated gradient descent

```
Initialization: $\theta(0) = 1$, $u(0) = u(-1) \in \mathbb{R}^{n_u}$, $\eta \leq 1/\omega$.
Iteration:
  1) Update $\theta(k+1)$ by $\theta(k+1) = \frac{1}{2}(1 + \sqrt{4\theta(k)^2 + 1})$
  2) Update $\hat{u}(k+1)$ by $\hat{u}(k+1) = u(k) + \frac{\theta(k)-1}{\theta(k+1)}(u(k) - u(k-1))$
  3) Update $u(k+1)$ by $u(k+1) = \hat{u}(k+1) - \eta\nabla f(\hat{u}(k+1))$
```

We adapt this acceleration technique to the stochastic DAL method, as shown in Algorithm 4. Since we apply the gradient step to update the dual variable, an interpolated point of a dual variable must be computed at each iteration based on the rule shown in steps 1 and 2 of Algorithm 3. Due to this requirement, all subsystems must always be active at each iteration, i.e., the following assumption holds.

Assumption 5: The probability of subsystem $p \in \mathcal{P}$ being active is 1, i.e., $\mathbb{P}(p \in \mathcal{A}(k)) = 1$, for all $p \in \mathcal{P}$ and $k \in \mathbb{Z}_{\geq 0}$. □

Algorithm 4 Stochastic accelerated DAL method

```
Initialization: For each node $i \in \mathcal{N}$, $\boldsymbol{v}_i(0) \in \mathbb{R}^{|\mathcal{N}_i|h}$. Moreover, $\boldsymbol{z}_i^j(0) = \boldsymbol{v}_j^i(0)$, $\boldsymbol{\lambda}_i^j(0) = \boldsymbol{\lambda}_j^i(0) \in \mathbb{R}^h$, and $\eta_i^j = \eta_j^i \in (0, \frac{1}{4})$, for all $j \in \mathcal{N}_i$ and $i \in \mathcal{N}$. Additionally, $\theta(0) = 1$.
Iteration: For each subsystem $p \in \mathcal{P}$,
  1) Update $\theta(k+1)$ as follows:
       $$\theta(k+1) = \frac{1}{2}(1 + \sqrt{4\theta(k)^2 + 1}).$$
  2) Update $\hat{\boldsymbol{\lambda}}_i(k+1)$, for all $i \in \mathcal{M}_p$, as follows:
       $$\hat{\boldsymbol{\lambda}}_i(k+1) = \boldsymbol{\lambda}_i(k) + \frac{\theta(k)-1}{\theta(k+1)}(\boldsymbol{\lambda}_i(k) - \boldsymbol{\lambda}_i(k-1)).$$
  3) Update $(\boldsymbol{u}_i(k+1), \hat{\boldsymbol{v}}_i(k))$, for all $i \in \mathcal{M}_p$, according to
       $$\begin{aligned} &\{(\boldsymbol{u}_i(k+1), \hat{\boldsymbol{v}}_i(k))\}_{i\in\mathcal{M}_p} \\ &= \arg \min_{\{(\boldsymbol{u}_i,\boldsymbol{v}_i)\in\mathcal{L}_i\}_{i\in\mathcal{M}_p}} \sum_{i\in\mathcal{M}_p} \Bigg( f_i^{\mathrm{l}}(\boldsymbol{u}_i) + f_i^{\mathrm{c}}(\boldsymbol{v}_i) + \\ &\quad + \sum_{j\in\mathcal{N}_i} \Big( 2\langle \hat{\boldsymbol{\lambda}}_i^j(k+1), \boldsymbol{v}_i^j \rangle + \|\boldsymbol{v}_i^j + \boldsymbol{z}_i^j(k)\|_2^2 \Big) \Bigg). \end{aligned}$$
  4) Update $\boldsymbol{v}_i^j(k+1)$, for all $j \in \mathcal{N}_i$ and $i \in \mathcal{M}_p$, as follows:
       $$\boldsymbol{v}_i^j(k+1) = \begin{cases} \eta_i^j \hat{\boldsymbol{v}}_i^j(k) + \left(1 - \eta_i^j\right) \boldsymbol{v}_i^j(k), \\ \qquad\qquad \text{if } \phi(j) \in \mathcal{A}_p(k+1) \cup \{p\}, \\ \boldsymbol{v}_i^j(k), \quad \text{otherwise}. \end{cases}$$
  5) For each $j \in \mathcal{N}_i \backslash \mathcal{M}_p$ such that $\phi(j) \in \mathcal{A}_p(k+1)$ and $i \in \mathcal{M}_p$, send $\boldsymbol{v}_i^j(k+1)$ to and receive $\boldsymbol{v}_j^i(k+1)$ from subsystem $\phi(j)$.
  6) Update the auxiliary and dual variables $\boldsymbol{z}_i(k+1)$ and $\boldsymbol{\lambda}_i(k+1)$, for all $i \in \mathcal{M}_p$, according to
       $$\boldsymbol{z}_i^j(k+1) = \begin{cases} \boldsymbol{v}_j^i(k+1), & \text{if } \phi(j) \in \mathcal{A}_p(k+1) \cup \{p\}, \\ \boldsymbol{z}_i^j(k), & \text{otherwise}, \end{cases}$$
       $$\boldsymbol{\lambda}_i^j(k+1) = \begin{cases} \hat{\boldsymbol{\lambda}}_i^j(k+1) + \eta_i^j \left( \boldsymbol{v}_i^j(k+1) + \boldsymbol{z}_i^j(k+1) \right), \\ \qquad\qquad \text{if } \phi(j) \in \mathcal{A}_p(k+1) \cup \{p\}, \\ \hat{\boldsymbol{\lambda}}_i^j(k+1), \quad \text{otherwise}. \end{cases}$$
```

Assumption 5 implies that Algorithm 4 is performed synchronously. Nevertheless, the communication network might still be randomly time-varying. Notice that in step 3 of Algorithm 4, each subsystem uses the interpolated points $\hat{\boldsymbol{\lambda}}_i(k+1)$, for all $i \in \mathcal{N}$, to update the primal variables. Note that we treat the acceleration technique as a heuristic method and the effectiveness will be shown through numerical simulations. Although [22], [23] show the convergence rate analysis of an accelerated ADMM method, the problems considered in those papers are limited and do not include Problem (11). Specifically, [23], which provides a more general result than [22], shows the convergence rate of $\mathcal{O}(\frac{1}{k^2})$ when the cost function is composed of two component, one is strongly convex while the other is quadratic. Moreover, the problems considered only have an equality constraint but do not include local constraint sets.

B. Warm start method

We also consider the warm start method [34, §4.3.3] to reduce the number of iterations performed by the distributed algorithms. As suggested by its name, the idea of this method is carefully choosing the initial condition of the variables. Particularly for an MPC scheme, where a similar problem is repeatedly solved at each time step, the variables can be initialized using the computed value from the previous time step. Note that the difference between the problem solved at one step and another is the value $d_{i,\tau}$, for all $i \in \mathcal{N}$ and $t \in \mathcal{T}$, considered at each step. At time step t, although only $u_{i,t}$ and $v_{i,t}$ are implemented, the decisions for h time steps ahead, i.e., $u_{i,\tau}$ and $v_{i,\tau}$, for all $\tau \in \mathcal{T}$, are computed. Furthermore, we can also record the dual variables from the last iteration at t and use them as the initial value at the next time step, $t+1$.

Now, consider Algorithms 2 and 4 and let $\boldsymbol{v}^j_{i,t|t} = \mathrm{col}(\{v^j_{i,\tau|t}\}_{\tau\in\mathcal{T}})$ and $\boldsymbol{\lambda}^j_{i,t|t} = \mathrm{col}(\{\lambda^j_{i,\tau|t}\}_{\tau\in\mathcal{T}})$, for all $j \in \mathcal{N}_i$ and $i \in \mathcal{N}$, denote the coupled and dual variables computed at time $t \in \mathbb{Z}_{\geq 0}$. Therefore, in the next time step, $t+1$, they can be initialized as follows:

$$\begin{aligned}\boldsymbol{v}^j_{i,t+1}(0) &= \mathrm{col}(\{\boldsymbol{v}^j_{i,\tau|t}\}^{t+h-1}_{\tau=t+1}, \boldsymbol{v}^j_{i,t+h-1|t}),\\ \boldsymbol{\lambda}^j_{i,t+1}(0) &= \mathrm{col}(\{\boldsymbol{\lambda}^j_{i,\tau|t}\}^{t+h-1}_{\tau=t+1}, \boldsymbol{\lambda}^j_{i,t+h-1|t}),\end{aligned}$$

for all $j \in \mathcal{N}_i$ and $i \in \mathcal{N}$.

Notice that we initialize the variables associated with the last step in the horizon, i.e., $t+h$, in the same way as we initialize the variables at $t+h-1$. The reason of this choice is twofold. Firstly, as previously explained, the difference between the optimization problem solved at t and $t+1$ is the uncertain variables d_i, for each $i \in \mathcal{N}$. Secondly, the difference between $d_{i,t+h}$ and $d_{i,t+h-1}$, for each $i \in \mathcal{N}$, might not be large, especially when the sampling period is small.

V. Numerical simulations

We consider the PG&E 69-bus distribution network, with the addition of 11 solar-powered non-dispatchable generation units, 11 storage units, and 13 dispatchable generation units. Figure 2 not only shows the locations of these units in the network but also the decomposition of the network into eight subsystems, which is obtained from [29]. The parameters of the active components in the network are given in Table II. Moreover, we use the available load data set as the maximum value of the load at each node. The nodes that have a maximum load greater than 100 kW are considered to have a commercial load profile. Otherwise, they have a residential load profile. Furthermore, Table I shows the maximum power generation of the non-dispatchable generation units. Additionally, it is assumed that the subsystems have the knowledge of the loads and non-dispatchable power generation, implying perfect forecasts, so that the analysis can be focused only on the outcomes of the algorithms. An example of residential and commercial load profiles is shown in the top plot of Figure 3, whereas a solar-based generation profile is shown in the bottom plot of Figure 3.

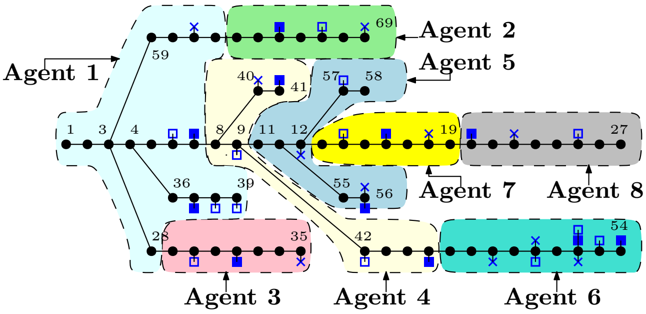

Fig. 2. The locations of non-dispatchable units (indicated by filled squares, ■), dispatchable generators (indicated by empty squares, □), and storage units (indicated by crosses, ×) in the PG&E 69-bus distribution network, which is partitioned into eight subsystems [29].

TABLE I
Maximum generated power of solar-based generation units

Nodes	Maximum power [kW]
$7, 32, 37, 45, 56, 65$	50
$20, 41$	75
$16, 54$	100
52	150

The simulation time is one day with the sampling time of 15 minutes, implying 96 steps. Moreover, the prediction horizon in the DMPC scheme is $h = 4$. Furthermore, we assume that the probability of each communication link being active is equal, i.e., $\beta_{pq} = \beta = 0.9$, for all $\{p,q\} \in \mathcal{E}^{\mathrm{s}}$, whereas all subsystems are always active at each iteration to accommodate Algorithm 4 (c.f. Assumption 5). We simulate both Algorithms 2 and 4 by using the same case and considering the step sizes to be $\eta^j_i = 0.2$, for all $\{i,j\} \in \mathcal{E}$. Moreover, their stopping criterion is $\|\boldsymbol{v}_t(k)+\boldsymbol{z}_t(k)\|_2 \leq 1$, which is the primal residual [34]. Note that since the stopping criterion corresponds to the coupling constraints and $\max \|\boldsymbol{v}^j_i + \boldsymbol{v}^i_j\|_2 = 80$, the choice of the stopping criterion is acceptably small. Additionally, we also apply the warm start method described in Section IV-B to both algorithms. Note that all simulations are carried out in MATLAB with Yalmip [35] on a PC with 16 GB of RAM and 2.6 GHz Intel core i7.

Figures 4-6 show the simulation results. Figure 4 shows the number of iterations required by the stochastic DAL and its accelerated version (Ac-DAL) over all time steps. The average number of iterations are 1294.9 and 730.6, respectively. From these results, we can see that Nesterov's accelerated method reduces the number of iterations needed to satisfy the stopping criterion close to 50%. However, it is also important to note that the acceleration technique does not allow the subsystems to be inactive randomly since they must always perform the interpolation steps at each iteration. Furthermore, we also

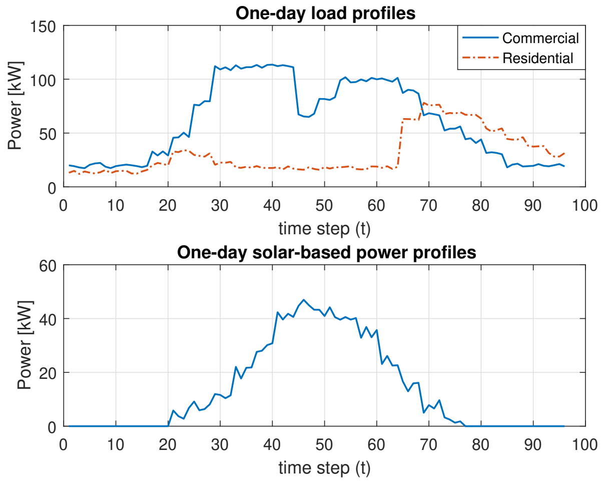

Fig. 3. One-day residential and commercial load profiles (top plot) and one-day solar-based power generation profile (bottom plot).

TABLE II
PARAMETERS OF THE NETWORK COMPONENTS

Parameters	Value	Unit	Bus
$u_i^{\mathrm{dg,min}}$, $u_i^{\mathrm{dg,max}}$	0, 400	kW	$i \in \mathcal{N}^{\mathrm{dg}}$
$x_i^{\min}$, $x_i^{\max}$, $x_{i,0}$	30, 100, 50	%	$i \in \mathcal{N}^{\mathrm{st}}$
u_i^{ch}, u_i^{dh}	100, 100	kW	$i \in \mathcal{N}^{\mathrm{st}}$
$e_{\mathrm{cap},i}$	1000	kWh	$i \in \mathcal{N}^{\mathrm{st}}$
a_i	1	-	$i \in \mathcal{N}^{\mathrm{st}}$
$v_i^{j,\max}$	200	kW	$\{i,j\} \in \mathcal{E}$

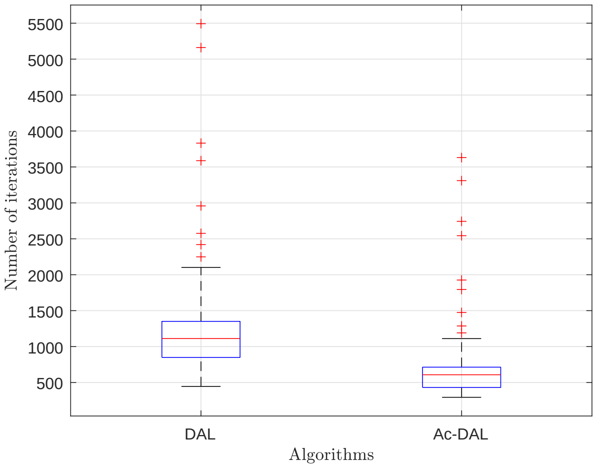

Fig. 4. The number of iterations performed by each algorithms. The blue boxes show the 25th percentiles until the 75th percentiles, the red lines show the median, and the + symbols indicate outliers.

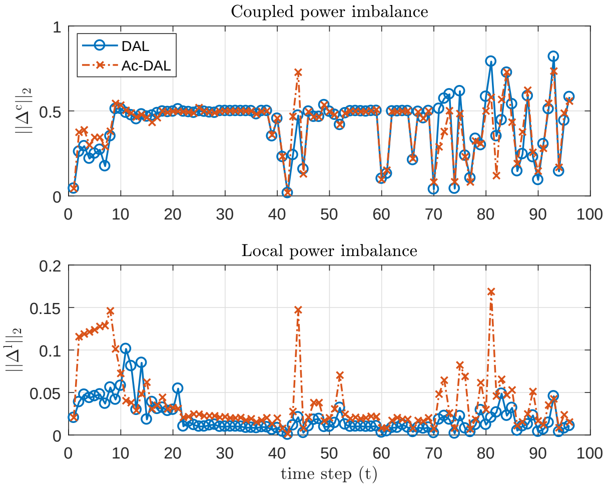

Fig. 5. Top plot shows the coupled power imbalance, $||\Delta^{\mathrm{c}}||_2$ (see (17)). Bottom plot shows the local power imbalance, $||\Delta^{\mathrm{l}}||_2$ (see (18)).

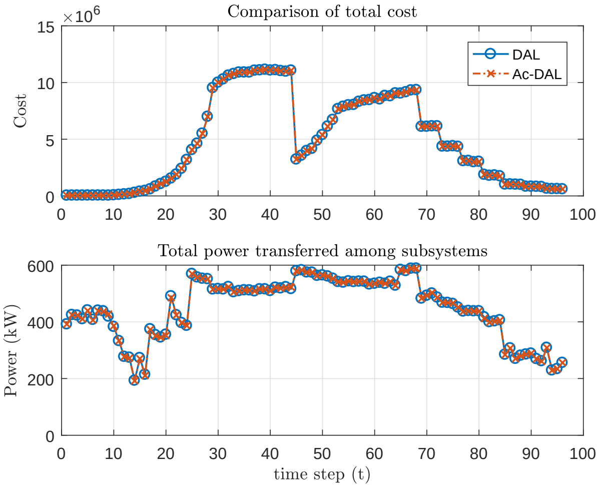

Fig. 6. The stage cost at each iteration (top plot) and the total power transferred among subsystems at each iteration (bottom plot).

evaluate the power imbalances. Based on (1) and (2), the coupled and local power imbalance indices, denoted by Δ_t^{l} and Δ_t^{c}, respectively, are defined as follows:

$$\Delta_t^{\mathrm{c}} = \mathrm{col}(\{v_{i,t}^j + v_{j,t}^i\}_{j \in \mathcal{N}_i, i \in \mathcal{N}}), \tag{17}$$

$$\Delta_t^{\mathrm{l}} = \mathrm{col}(\{\mathbb{1}_{n_i^{\mathrm{l}}}^\top u_{i,t} + \mathbb{1}_{|\mathcal{N}_i|}^\top v_{i,t} - d_{i,t}\}_{i \in \mathcal{N}}). \tag{18}$$

Figure 5 shows $\|\Delta_t^{\mathrm{c}}\|_2$ (top plot) and $\|\Delta_t^{\mathrm{l}}\|_2$ (bottom plot). It can be observed that $\|\Delta_t^{\mathrm{c}}\|_2$ is relatively similar. However, in terms of local power imbalance, $\|\Delta_t^{\mathrm{l}}\|_2$ obtained from the accelerated algorithm is most of the time larger than the standard algorithm, although the difference is quite small. Finally, we also show the cost achieved by both algorithms at each time step in the top plot of Figure 6. In addition, the subsystems must actively exchange power among each other, as shown in the bottom plot of Figure 6. Thus, Figures 5-6 indicate that the stochastic DAL and its accelerated version achieve a comparable performance.

VI. Conclusion and future work

We present a distributed model predictive control scheme for an economic dispatch of large-scale energy systems. The scheme is based on the distributed augmented Lagrangian approach and works over random communication networks and asynchronous updates. Furthermore, we also propose to implement two techniques, i.e., Nesterov's accelerated gradient method and the warm start method, to improve the convergence speed. The performance of the proposed scheme is showcased through a numerical study using a benchmark case. As ongoing work, we are performing a theoretical analysis of the accelerated version of the method. Furthermore, we also consider extending the proposed method such that more general problems can be handled, e.g., non-convex power flow equations or its convex approximations and coupled cost functions.

References

[1] D. E. Olivares, A. Mehrizi-Sani, A. H. Etemadi, C. A. Cañizares, R. Iravani, M. Kazerani, A. H. Hajimiragha, O. Gomis-Bellmunt, M. Saeedifard, R. Palma-Behnke, G. A. Jiménez-Estévez, and N. D. Hatziargyriou, "Trends in microgrid control," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1905–1919, 2014.

[2] T. Morstyn, B. Hredzak, and V. G. Agelidis, "Control strategies for microgrids with distributed energy storage systems: An overview," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3652–3666, 2018.

[3] D. K. Molzahn, F. Dörfler, H. Sandberg, S. H. Low, S. Chakrabarti, R. Baldick, and J. Lavaei, "A survey of distributed optimization and control algorithms for electric power systems," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 2941–2962, 2017.

[4] A. Parisio, C. Wiezorek, T. Kyntäjä, J. Elo, K. Strunz, and K. H. Johansson, "Cooperative MPC-based energy management for networked microgrids," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 3066–3074, 2017.

[5] C. A. Hans, P. Braun, J. Raisch, L. Grüne, and C. Reincke-Collon, "Hierarchical distributed model predictive control of interconnected microgrids," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 1, pp. 407–416, 2019.

[6] K. Baker, J. Guo, G. Hug, and X. Li, "Distributed MPC for efficient coordination of storage and renewable energy sources across control areas," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 992–1001, 2016.

[7] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, "Dynamic network energy management via proximal message passing," *Foundations and Trends in Optimization*, vol. 1, no. 2, pp. 73–126, 2014.

[8] T. Wang, D. O'Neill, and H. Kamath, "Dynamic control and optimization of distributed energy resources in a microgrid," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 2884–2894, 2015.

[9] G. K. H. Larsen, N. D. van Foreest, and J. M. A. Scherpen, "Distributed MPC applied to a network of households with micro-CHP and heat storage," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 2106–2114, 2014.

[10] A. K. Sampathirao, C. A. Hans, and J. Raisch, "Two-stage consensus-based distributed mpc for interconnected microgrids," *arXiv preprint arXiv:1810.03361*, 2018.

[11] A. Kargarian, J. Mohammadi, J. Guo, S. Chakrabarti, M. Barati, G. Hug, S. Kar, and R. Baldick, "Toward distributed/decentralized DC optimal power flow implementation in future electric power systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2574–2594, 2018.

[12] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2010, vol. 25, no. 3.

[13] J. Wu, T. Yang, D. Wu, K. Kalsi, and K. H. Johansson, "Distributed optimal dispatch of distributed energy resources over lossy communication networks," *IEEE Transactions on Smart Grid*, vol. 8, no. 6, pp. 3125–3137, 2017.

[14] T. Yang, D. Wu, H. Fang, W. Ren, H. Wang, Y. Hong, and K. H. Johansson, "Distributed energy resource coordination over time-varying directed communication networks," *IEEE Transactions on Control of Network Systems*, vol. 6, no. 3, pp. 1124–1134, 2019.

[15] M. Zholbaryssov, C. N. Hadjicostis, and A. D. Dominguez-Garcia, "Fast distributed coordination of distributed energy resources over time-varying communication networks," pp. 1–13, 2019, arXiv:1907.07600.

[16] E. Wei and A. Ozdaglar, "On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers," pp. 1–30, 2013, arXiv:1307.8254.

[17] T. Chang, "A proximal dual consensus ADMM method for multi-agent constrained optimization," *IEEE Transactions on Signal Processing*, vol. 64, no. 14, pp. 3719–3734, 2016.

[18] Y. Nesterov, "A method for solving the convex programming problem with convergence rate $O(1/k^2)$," *Dokl. Akad. Nauk SSSR*, vol. 27, p. 543–547, 1983, translated as Sov. Math. Dokl.

[19] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control," *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 18–33, 2013.

[20] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer, "Accelerated gradient methods and dual decomposition in distributed model predictive control," *Automatica*, vol. 49, no. 3, pp. 829–833, 2013.

[21] A. Beck, A. Nedić, A. Ozdaglar, and M. Teboulle, "An $o(1/k)$ gradient method for network resource allocation problems," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 64–73, 2014.

[22] D. Goldfarb, S. Ma, and K. Scheinberg, "Fast alternating linearization methods for minimizing the sum of two convex functions," *Mathematical Programming*, vol. 141, no. 1-2, pp. 349–382, 2013.

[23] T. Goldstein, B. O'Donoghue, S. Setzer, and R. Baraniuk, "Fast alternating direction optimization methods," *SIAM Journal on Imaging Sciences*, vol. 7, no. 3, pp. 1588–1623, 2014.

[24] N. Chatzipanagiotis, D. Dentcheva, and M. M. Zavlanos, "An augmented Lagrangian method for distributed optimization," *Mathematical Programming*, vol. 152, no. 1, pp. 405–434, 2015.

[25] S. Lee, N. Chatzipanagiotis, and M. M. Zavlanos, "Complexity certification of a distributed augmented Lagrangian method," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 827–834, 2018.

[26] W. Ananduta, A. Nedić, and C. Ocampo-Martinez, "Distributed augmented Lagrangian method for link-based resource sharing problems of multi-agent systems," *IEEE Transactions on Automatic Control*, accepted, preprint available at https://arxiv.org/abs/2103.00404.

[27] C. Schwaegerl and L. Tao, *The Microgrids Concept*. John Wiley & Sons, Ltd, 2013, ch. 1, pp. 1–24.

[28] T. Sousa, T. Soares, P. Pinson, F. Moret, T. Baroche, and E. Sorin, "Peer-to-peer and community-based markets: A comprehensive review," *Renewable and Sustainable Energy Reviews*, vol. 104, pp. 367–378, 2019.

[29] S. A. Arefifar, Y. A.-R. I. Mohamed, and T. H. M. El-Fouly, "Supply-adequacy-based optimal construction of microgrids in smart distribution systems," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1491–1502, 2012.

[30] S. A. Arefifar, Y. A.-R. I. Mohamed, and T. H. M. EL-Fouly, "Optimum microgrid design for enhancing reliability and supply-security," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1567–1575, 2013.

[31] M. Barani, J. Aghaei, M. A. Akbari, T. Niknam, H. Farahmand, and M. Korpås, "Optimal partitioning of smart distribution systems into supply-sufficient microgrids," *IEEE Transactions on Smart Grid*, 2018, in press.

[32] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[33] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *SIAM Journal on Optimization*, submitted.

[34] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[35] J. Löfberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Proceedings of the CACSD Conference*, vol. 3. Taipei, Taiwan, 2004.