ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

A Duality-Based Approach for Distributed Min-Max Optimization

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
A Duality-Based Approach for Distributed Min-Max Optimization / Notarnicola, Ivano; Franceschelli, Mauro; Notarstefano, Giuseppe. - In: IEEE TRANSACTIONS ON AUTOMATIC CONTROL. - ISSN 1558-2523. - ELETTRONICO. - 64:6(2019), pp. 2559-2566. [10.1109/TAC.2018.2872200]

Availability:
This version is available at: https://hdl.handle.net/11585/671879 since: 2020-02-23

Published:

DOI: http://doi.org/10.1109/TAC.2018.2872200

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

19 April 2024

This is the final peer-reviewed accepted manuscript of:

I. Notarnicola, M. Franceschelli and G. Notarstefano, "A Duality-Based Approach for Distributed Min–Max Optimization," in IEEE Transactions on Automatic Control, vol. 64, no. 6, pp. 2559-2566, June 2019

The final published version is available online at: **http://dx.doi.org/10.1109/TAC.2018.2872200**

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

A Duality-Based Approach for Distributed Min-Max Optimization

Ivano Notarnicola[1], Mauro Franceschelli[2], Giuseppe Notarstefano[3]

***Abstract*—In this paper we consider a distributed optimization scenario, motivated by peak-demand minimization, in which a set of processors aims at cooperatively solving a class of min-max optimization problems. The min-max structure and a twofold coupling make the problem challenging in a distributed set-up. We propose a distributed algorithm based on the derivation of a series of dual problems and the application of properties from min-max optimization. The resulting distributed algorithm, despite its complex derivation, has a simple structure consisting of a primal optimization and a suitable dual update. We prove the convergence of the proposed algorithm in objective value, and, moreover, that every limit point of the primal sequence is an optimal (thus feasible) solution. This primal recovery property is of key importance in applications, since it allows each agent to compute its portion of the global optimal strategy without resorting to any recovery mechanism. Finally, we provide numerical computations for peak-demand optimization in a network of thermostatically controlled loads and show that our algorithm outperforms a plain distributed subgradient performed on the dual.**

I. Introduction

Smart grids provide a rich set of motivating scenarios for distributed optimization. An interesting example is the design of smart generators, accumulators and loads that cooperatively execute Demand Side Management (DSM) programs [2] to reduce hourly and daily variations and peaks of electric demand. A widely adopted objective in DSM programs is Peak-to-Average Ratio (PAR), defined as the ratio between peak-daily and average-daily power demands [3]. This problem has been already investigated in the literature in a noncooperative framework [3], [4]. It is worth noting that in this literature the term "distributed" indicates that data are deployed on a set of devices performing local computation simultaneously. Differently from the references above, we consider a cooperative, distributed computation model in which the agents in the network solve the optimization problem: (i) without any knowledge of aggregate quantities, (ii) by exchanging information over a communication graph, and (iii) by performing local computations (with no central coordinator).

The distributed algorithm proposed in the paper heavily relies on duality theory, which is widely used in parallel and (classical) distributed optimization algorithms as shown, e.g., in [5]–[7]. Primal recovery is a key issue in dual methods, since the primal sequence is not guaranteed, in general, to satisfy the dualized primal constraint. Thus, in the (centralized) optimization literature running average strategies have been proposed to cope with this issue, see, e.g., [8]–[11]. In [12]–[14] a distributed optimization set-up with coupling constraints, as the one we have in this paper, is considered. These papers propose distributed algorithms based on a "single duality step" with a recovering mechanism. In [15] the authors propose a subgradient method to generate approximate saddle-points, while in [16] a distributed algorithm based on a suitable penalty approach has been proposed for min-max problems without coupling constraints. In [17] a distributed projected subgradient method is proposed to solve constrained saddle-point problems with agreement constraints. Although our problem set-up fits in the ones considered in [17], our algorithmic approach and analysis are different. In [18], [19] saddle point dynamics are used to design distributed algorithms for standard separable optimization problems. Another class of algorithms exploits the exchange of active constraints among the network nodes to solve constrained optimization problems which include min-max problems, [20], [21].

In this paper we propose a distributed algorithm to solve min-max optimization problems strongly motivated by peak power-demand minimization in DSM. The problem is challenging when approached in a distributed way since it is *doubly coupled*. Each term of the max function (i.e., the cost at a given time-slot) is the sum of local functions known only by each agent. Moreover, the local constraints (due to the device dynamics) impose a coupling between different slots in the time-horizon. The proposed distributed algorithm has a very simple and clean structure in which a primal optimization and a dual update are iteratively performed. Despite this simple structure, apparently reminiscent of a standard distributed dual method, the algorithm derivation relies on several proper duality steps, performed to decompose the originally coupled problem into locally-coupled subproblems, and thus design a distributed algorithm. To the best of our knowledge this approach has never been proposed so far. A key property of the algorithm is that every limit point of the primal minimizing sequence at each node is an optimal (thus feasible) solution of the original optimization problem although this is only

A preliminary short version of this paper has appeared as [1]. The research leading to these results has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 638992 - OPT4SMART) and from the Ministero dell'Istruzione, dell'Università e della Ricerca (MIUR) under the italian programme "Scientific Independence of Young Researchers" (SIR) 2014, grant no. F22I15000080008, project "CoNetDomeSys".

[1]I. Notarnicola is with the Department of Engineering, Università del Salento, Lecce, Italy, ivano.notarnicola@unisalento.it.

[2]M. Franceschelli is with the Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy, mauro.franceschelli@diee.unica.it.

[3]G. Notarstefano is with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy, giuseppe.notarstefano@unibo.it.

convex but not strictly convex. This property is obtained without resorting to averaging schemes, as usually done in centralized [8], [9], [11] and distributed approaches [13], [14], [17]. We show through simulations that our distributed algorithm significantly outperforms a plain distributed subgradient applied to the dual of the given problem. Finally, since each node only computes the decision variable of interest, our algorithm is scalable with the number of agents (large-scale problems) and with the length of the horizon (big-data problems).

The paper is structured as follows. In Section II we formalize the distributed min-max optimization set-up and present our duality-based distributed optimization method, while in Section III we characterize its convergence properties. In Section IV we corroborate the theoretical results with a numerical analysis on peak minimization in smart-grids.

II. Problem Set-up and Distributed Optimization Algorithm

In this section we set up the distributed min-max optimization framework and present our novel distributed algorithm.

A. Distributed min-max optimization set-up

We consider a network of N processors which communicate according to a *connected*, undirected graph $\mathcal{G} = (\{1,\dots,N\},\mathcal{E})$, where $\mathcal{E} \subseteq \{1,\dots,N\}\times\{1,\dots,N\}$ is the set of edges. That is, the edge (i,j) models the fact that node i and j exchange information. We denote by $\mathcal{N}_i$ the set of *neighbors* of node i in the fixed graph $\mathcal{G}$, i.e., $\mathcal{N}_i := \{j \in \{1,\dots,N\} \mid (i,j)\in\mathcal{E}\}$. Also, we denote by a_{ij} the element i,j of the adjacency matrix. We recall that $a_{ij}=1$ if $(i,j)\in\mathcal{E}$ and $i\neq j$, and $a_{ij}=0$ otherwise.

Next, we introduce the min-max optimization problem to be solved by the network processors in a distributed way. Specifically, we associate to each processor i a decision vector $\mathbf{x}^i = [\mathbf{x}^i_1,\dots,\mathbf{x}^i_S]^\top \in \mathbb{R}^S$, a constraint set $X^i \subseteq \mathbb{R}^S$ and local functions g^i_s, $s\in\{1,\dots,S\}$. Note that we use the superscript $i\in\{1,\dots,N\}$ to indicate that a vector $\mathbf{x}^i\in\mathbb{R}^S$ belongs to node i, while we use the subscript $s\in\{1,\dots,S\}$ to identify a vector component, i.e., $\mathbf{x}^i_s$ is the s-th component of $\mathbf{x}^i$. In the peak-minimization problem, S represents the horizon on which we want to reduce the peak.

Thus, we set up the following optimization problem

$$\begin{aligned} \min_{\mathbf{x}^1,\dots,\mathbf{x}^N} \ & \max_{s\in\{1,\dots,S\}} \sum_{i=1}^N g^i_s(\mathbf{x}^i_s) \\ \text{subj. to } \ & \mathbf{x}^i \in X^i, \qquad\qquad i\in\{1,\dots,N\}, \end{aligned} \tag{1}$$

where for each $i\in\{1,\dots,N\}$ the set $X^i\subseteq\mathbb{R}^S$ is nonempty, convex and compact, and the functions $g^i_s:\mathbb{R}\to\mathbb{R}$, $s\in\{1,\dots,S\}$, are convex. Using a standard approach for min-max problems, we introduce an auxiliary variable P to write the so-called epigraph representation of problem (1), given by

$$\begin{aligned} \min_{\mathbf{x}^1,\dots,\mathbf{x}^N,P} \ & P \\ \text{subj. to } \ & \mathbf{x}^i \in X^i, \qquad\qquad i\in\{1,\dots,N\} \\ & \sum_{i=1}^N g^i_s(\mathbf{x}^i_s) \le P, \qquad s\in\{1,\dots,S\}. \end{aligned} \tag{2}$$

It is worth noticing that this problem has a particular structure, which gives rise to interesting challenges in a distributed set-up. First of all, two types of couplings are present, which involve simultaneously the N agents and the S components of each decision variable $\mathbf{x}^i$. Specifically, for a given slot s, the constraint $\sum_{i=1}^N g^i_s(\mathbf{x}^i_s)\le P$ couples all the vectors $\mathbf{x}^i$, $i\in\{1,\dots,N\}$. At the same time, for a given agent $i\in\{1,\dots,N\}$, the constraint X^i couples all the components $\mathbf{x}^i_1,\dots,\mathbf{x}^i_S$ of $\mathbf{x}^i$. Figure 1 provides a nice graphical representation of this interlaced coupling. Moreover, the problem is both large-scale and big-data. That is, both the number of decision variables and the number of constraints depend on N (and thus scale badly with the number of agents in the network). Also, the dimension of the coupling constraint, S, can be large. Therefore, common approaches as reaching a consensus among the nodes on an optimal solution and/or exchanging constraints are not computationally affordable.

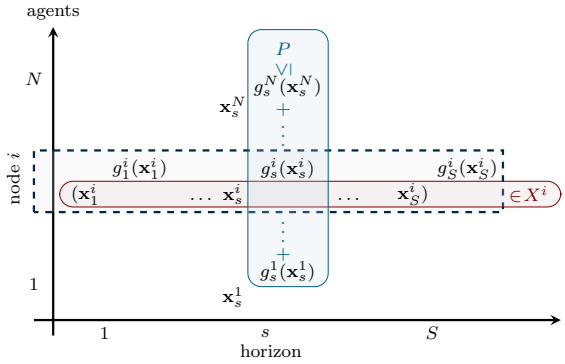

Fig. 1. Graphical representation of interlaced constraints.

To conclude this section, we notice that problem (2) is convex, but not strictly convex. This means that it is not guaranteed to have a unique optimal solution. As discussed in the introduction, this impacts on dual approaches when trying to recover a primal optimal solution, see e.g., [8], [9], [11]. This aspect is even more delicate in a distributed set-up in which nodes only know part of the constraints and of the objective function.

B. Distributed Duality-Based Peak Minimization (DDPM)

Next, we introduce our distributed optimization algorithm. Informally, the algorithm consists of a two-step iterative procedure. First, each node $i\in\{1,\dots,N\}$ stores a set of variables $((\mathbf{x}^i,\, P^i),\, \boldsymbol{\mu}^i)$ obtained as a primal-dual optimal solution pair of a local optimization problem with an epigraph structure as the centralized problem. The coupling with the other nodes in the original formulation is replaced by a term depending on neighboring variables $\boldsymbol{\lambda}^{ij}$, $j\in\mathcal{N}_i$. These variables are updated in the second step according to a suitable linear law weighting the difference of neighboring $\boldsymbol{\mu}^i$. Nodes use a diminishing step-size denoted by $\gamma(t)$ and can initialize the variables $\boldsymbol{\lambda}^{ij}$, $j\in\mathcal{N}_i$, to arbitrary values. In the Distributed Algorithm table below we formally state our Distributed Duality-Based Peak Minimization (DDPM) algorithm from i-th node's perspective.

Distributed Algorithm DDPM

Processor states: $(\mathbf{x}^i, P^i)$, $\boldsymbol{\mu}^i$ and $\boldsymbol{\lambda}^{ij}$ for $j \in \mathcal{N}_i$
Evolution:
Gather $\boldsymbol{\lambda}^{ji}(t)$ from $j \in \mathcal{N}_i$
Compute $\big((\mathbf{x}^i(t+1), P^i(t+1)), \boldsymbol{\mu}^i(t+1)\big)$ as a primal-dual optimal solution pair of

$$\begin{aligned} &\min_{\mathbf{x}^i, P^i} \; P^i \\ &\text{subj. to } \mathbf{x}^i \in X^i \\ &\qquad g_s^i(\mathbf{x}_s^i) + \sum_{j \in \mathcal{N}_i} \big(\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t)\big)_s \le P^i, \\ &\qquad\qquad\qquad\qquad s \in \{1, \dots, S\} \end{aligned} \tag{3}$$

Gather $\boldsymbol{\mu}^j(t+1)$ from $j \in \mathcal{N}_i$
Update for all $j \in \mathcal{N}_i$

$$\boldsymbol{\lambda}^{ij}(t+1) = \boldsymbol{\lambda}^{ij}(t) - \gamma(t)(\boldsymbol{\mu}^i(t+1) - \boldsymbol{\mu}^j(t+1)) \tag{4}$$

The structure of the algorithm and the meaning of the updates will be clear in the constructive analysis carried out in the next section. We point out that although problem (3) has the same epigraph structure of problem (2), P^i is not a copy of the centralized cost P, but rather a local contribution to that cost. That is, as we will see, the total cost P will be the sum of the P^is.

III. Algorithm Analysis

The analysis of the proposed DDPM distributed algorithm is constructive and heavily relies on duality theory tools.

First, we derive several equivalent formulations of problem (2). Second, the last formulation is solved by means of a proper dual subgradient algorithm that we prove to be well posed and implementable in a distributed manner. The DDPM distributed algorithm turns out to be an operative way to implement such a subgradient method. Finally, we prove that both optimality and feasibility are asymptotically satisfied by the sequences generated by the DDPM distributed algorithm.

We start by writing the dual problem of (2). Let $\boldsymbol{\mu} := [\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_S]^\top \in \mathbb{R}^S$, then its partial Lagrangian is given by

$$\begin{aligned} \mathcal{L}_1(\mathbf{x}^1, \dots, \mathbf{x}^N, P, \boldsymbol{\mu}) &= P + \sum_{s=1}^S \boldsymbol{\mu}_s \Big(\sum_{i=1}^N g_s^i(\mathbf{x}_s^i) - P \Big) \\ &= P\Big(1 - \sum_{s=1}^S \boldsymbol{\mu}_s\Big) + \sum_{i=1}^N \sum_{s=1}^S \boldsymbol{\mu}_s g_s^i(\mathbf{x}_s^i). \end{aligned}$$

The dual function is defined as

$$q(\boldsymbol{\mu}) := \min_{\mathbf{x}^1 \in X^1, \dots, \mathbf{x}^N \in X^N, P} \mathcal{L}_1(\mathbf{x}^1, \dots, \mathbf{x}^N, P, \boldsymbol{\mu}),$$

where the presence of constraints $\mathbf{x}^i \in X^i$ for all $i \in \{1, \dots, N\}$ is due to the fact that we have not dualized them.

The minimization of $\mathcal{L}_1$ with respect to P gives rise to the simplex constraint $\sum_{s=1}^S \boldsymbol{\mu}_s = 1$. The minimization with respect to $\mathbf{x}^i$, $i \in \{1, \dots, N\}$, splits over i, so that the dual problem can be written as

$$\begin{aligned} &\max_{\boldsymbol{\mu} \in \mathbb{R}^S} \; \sum_{i=1}^N q^i(\boldsymbol{\mu}) \\ &\text{subj. to } \mathbf{1}^\top \boldsymbol{\mu} = 1, \; \boldsymbol{\mu} \succeq \mathbf{0}, \end{aligned} \tag{5}$$

with $\mathbf{1} := [1, \dots, 1]^\top \in \mathbb{R}^S$ and

$$q^i(\boldsymbol{\mu}) := \min_{\mathbf{x}^i \in X^i} \sum_{s=1}^S \boldsymbol{\mu}_s g_s^i(\mathbf{x}_s^i), \quad \forall\, i \in \{1, \dots, N\}. \tag{6}$$

Moreover, the primal problem is convex and satisfies the constraint qualification. Indeed, the cost and the constraints are convex, and a strictly feasible point can be easily found by choosing $\bar{\mathbf{x}}^i \in X^i$ and a sufficiently large (finite) $\bar{P}$ such that the coupling inequalities are strictly satisfied. Thus, by [22, Proposition 5.3.1] strong duality holds, i.e., the optimal cost $P^\star$ of problem (2) is equal to the optimal cost $q^\star$ of (5). Now, a discussion on how to solve problem (5) is in order. The distributed subgradient method introduced in [23] can be applied to solve problem (5). However, we point out a critical aspect, namely that the aforementioned approach is not able to recover a solution of the primal problem (2) unless an averaging mechanism is employed. In applications as DSM, the primal recovery property is fundamental, since it guarantees that each agent is able to compute its portion of the global optimal strategy in a distributed way.

The DDPM algorithm we propose in this paper to solve (2), explicitly explores the problem structure, thus allowing us to compute a primal feasible solution without any averaging mechanisms. Moreover, in Section IV we compare our approach with a plain distributed subgradient [23] applied to the dual problem (5), and show that it exhibits better performances.

We start by rewriting problem (5) into an equivalent form which will make it amenable for a distributed solution. To this end, we introduce copies of the common optimization variable $\boldsymbol{\mu}$ and coherence constraints having the sparsity of the connected graph $\mathcal{G}$, thus obtaining

$$\begin{aligned} &\max_{\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N} \; \sum_{i=1}^N q^i(\boldsymbol{\mu}^i) \\ &\text{subj. to } \mathbf{1}^\top \boldsymbol{\mu}^i = 1, \; \boldsymbol{\mu}^i \succeq \mathbf{0}, \qquad i \in \{1, \dots, N\} \\ &\qquad\qquad \boldsymbol{\mu}^i = \boldsymbol{\mu}^j, \qquad\quad (i,j) \in \mathcal{E}. \end{aligned} \tag{7}$$

Notice that we have also duplicated the simplex constraint, so that it becomes *local* at each node.

With formulation (7) at hand, we want to apply the dual subgradient algorithm to solve it. In the following, we formally derive the algorithm and prove that our DDPM is a proper implementation of it.

The partial Lagrangian of (7) is given by

$$\begin{aligned} &\mathcal{L}_2(\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N, \{\boldsymbol{\lambda}^{ij}\}_{(i,j) \in \mathcal{E}}) \\ &\qquad = \sum_{i=1}^N \Big(q^i(\boldsymbol{\mu}^i) + \sum_{j \in \mathcal{N}_i} \boldsymbol{\lambda}^{ij\top} (\boldsymbol{\mu}^i - \boldsymbol{\mu}^j) \Big), \end{aligned} \tag{8}$$

where each $\boldsymbol{\lambda}^{ij} \in \mathbb{R}^S$ is a multiplier associated to the constraints $\boldsymbol{\mu}^i - \boldsymbol{\mu}^j = \mathbf{0}$, for all $(i,j) \in \mathcal{E}$.

Since the communication graph $\mathcal{G}$ is undirected and connected, we can exploit the symmetry of the constraints. In fact, for each $(i,j) \in \mathcal{E}$ we also have $(j,i) \in \mathcal{E}$, and, expanding all the terms in (8), for given i and j, we always have both the terms ${\boldsymbol{\lambda}^{ij}}^\top(\boldsymbol{\mu}^i - \boldsymbol{\mu}^j)$ and ${\boldsymbol{\lambda}^{ji}}^\top(\boldsymbol{\mu}^j - \boldsymbol{\mu}^i)$. Thus, after some simple algebraic manipulations, we get

$$\mathcal{L}_2(\boldsymbol{\mu}^1, \dots, \boldsymbol{\mu}^N, \{\boldsymbol{\lambda}^{ij}\}_{(i,j)\in\mathcal{E}}) = \sum_{i=1}^N \Big(q^i(\boldsymbol{\mu}^i) + {\boldsymbol{\mu}^i}^\top \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij} - \boldsymbol{\lambda}^{ji}) \Big),$$

which is separable with respect to $\boldsymbol{\mu}^i$.

The dual of problem (7) is thus

$$\min_{\{\boldsymbol{\lambda}^{ij}\}_{(i,j)\in\mathcal{E}}} \eta(\{\boldsymbol{\lambda}^{ij}\}_{(i,j)\in\mathcal{E}}) = \sum_{i=1}^N \eta^i(\{\boldsymbol{\lambda}^{ij}, \boldsymbol{\lambda}^{ji}\}_{j\in\mathcal{N}_i}), \quad (9)$$

with, for all $i \in \{1, \dots, N\}$,

$$\eta^i(\{\boldsymbol{\lambda}^{ij}, \boldsymbol{\lambda}^{ji}\}_{j\in\mathcal{N}_i}) := \max_{\mathbf{1}^\top\boldsymbol{\mu}^i = 1, \boldsymbol{\mu}^i \succeq \mathbf{0}} q^i(\boldsymbol{\mu}^i) + {\boldsymbol{\mu}^i}^\top \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij} - \boldsymbol{\lambda}^{ji}). \quad (10)$$

Next, we characterize the properties of problem (9). We notice that problem (5) is a dual problem, thus its cost function $\sum_{i=1}^N q^i(\boldsymbol{\mu})$ is concave on its (convex) domain, namely the simplex constraint. Moreover, its optimal cost $q^\star = P^\star$ is finite. Problem (7) is an equivalent formulation of (5) and, thus, has the same (finite) optimal cost $q^\star$. Since it has a nonempty polyhedral domain, by [22, Proposition 5.2.1] it enjoys strong duality, i.e., its dual problem (9) has finite optimal cost $\eta^\star$ with

$$\eta^\star = q^\star = P^\star.$$

Problem (9) has a particularly appealing structure for distributed computation. In fact, the cost function η is separable and each term η^i depends only on neighboring variables $\boldsymbol{\lambda}^{ij}$ and $\boldsymbol{\lambda}^{ji}$, $j \in \mathcal{N}_i$. Thus, a subgradient method applied to this problem turns out to be a distributed algorithm. Since problem (7) has compact domain, we recall, [22, Section 6.1], how to compute a subgradient of its dual function η with respect to each component, that is,

$$\frac{\tilde{\partial}\eta(\{\boldsymbol{\lambda}^{ij}\}_{(i,j)\in\mathcal{E}})}{\partial\boldsymbol{\lambda}^{ij}} = \boldsymbol{\mu}^{i\star} - \boldsymbol{\mu}^{j\star}, \quad (11)$$

where $\frac{\tilde{\partial}\eta(\cdot)}{\partial\boldsymbol{\lambda}^{ij}}$ denotes the component associated to the variable $\boldsymbol{\lambda}^{ij}$ of a subgradient of η, and

$$\boldsymbol{\mu}^{k\star} \in \operatorname*{argmax}_{\mathbf{1}^\top\boldsymbol{\mu}^k = 1, \boldsymbol{\mu}^k \succeq \mathbf{0}} \Big(q_k(\boldsymbol{\mu}^k) + {\boldsymbol{\mu}^k}^\top \sum_{h\in\mathcal{N}_k} (\boldsymbol{\lambda}^{kh} - \boldsymbol{\lambda}^{hk}) \Big), k = i, j.$$

The distributed dual subgradient algorithm for problem (7) can be summarized as follows. For each node $i \in \{1, \dots, N\}$:

(S1) receive $\boldsymbol{\lambda}^{ji}(t)$, for each $j \in \mathcal{N}_i$, and compute a subgradient $\boldsymbol{\mu}^i(t+1)$ by solving

$$\max_{\boldsymbol{\mu}^i} \; q^i(\boldsymbol{\mu}^i) + {\boldsymbol{\mu}^i}^\top \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t)) \quad \text{subj. to } \mathbf{1}^\top\boldsymbol{\mu}^i = 1, \boldsymbol{\mu}^i \succeq \mathbf{0} \quad (12)$$

(S2) exchange with neighbors the updated $\boldsymbol{\mu}^j(t+1)$, $j \in \mathcal{N}_i$, and update $\boldsymbol{\lambda}^{ij}$, $j \in \mathcal{N}_i$, via

$$\boldsymbol{\lambda}^{ij}(t+1) = \boldsymbol{\lambda}^{ij}(t) - \gamma(t)(\boldsymbol{\mu}^i(t+1) - \boldsymbol{\mu}^j(t+1)).$$

where $\gamma(t)$ denotes a diminishing step-size satisfying

$$\lim_{t\to\infty} \gamma(t) = 0, \; \sum_{t=1}^\infty \gamma(t) = \infty, \; \sum_{t=1}^\infty \gamma(t)^2 < \infty. \quad (13)$$

It is worth noting that in (12) the value of $\boldsymbol{\lambda}^{ij}(t)$ and $\boldsymbol{\lambda}^{ji}(t)$, for $j \in \mathcal{N}_i$, is fixed as highlighted by the index t. Moreover, we want to stress, once again, that the algorithm is *not* implementable as it is written, since functions q^i are not available in closed form. Finally, we notice that dual subgradient algorithm (S1)-(S2) could be implemented in a distributed way by letting each node i solve problem (14) and exchange $\boldsymbol{\lambda}^{ij}(t)$ and $\boldsymbol{\lambda}^{ji}(t)$ with neighbors $j \in \mathcal{N}_i$.

We point out that here we slightly abuse notation since in (S1)-(S2), we use $\boldsymbol{\mu}^i(t)$ as in the DDPM algorithm, but without proving their equivalence yet. Since we will prove it in the next, we preferred not to overweight the notation.

We can now discuss the convergence in objective value of the sequences $\{\boldsymbol{\lambda}^{ij}(t)\}_{t\geq 0}$, $(i,j) \in \mathcal{E}$ generated by dual subgradient (S1)-(S2). As already recalled in equation (11), we can build subgradients of η by solving a problem in the form (12). Since in (12), the maximization of the concave function q^i is performed over the nonempty, compact (and convex) probability simplex $\mathbf{1}^\top\boldsymbol{\mu}^i = 1$, $\boldsymbol{\mu}^i \succeq \mathbf{0}$, then the maximum is always attained at a finite value. As a consequence, at each iteration the subgradients of η are bounded quantities. Let the step-size $\gamma(t)$ satisfy (13), thus we can invoke [24, Proposition 3.2.6] to conclude that $\{\boldsymbol{\lambda}^{ij}(t)\}_{t\geq 0}$, $(i,j) \in \mathcal{E}$, converges in objective value to the optimal cost $\eta^\star$ of (9).

In the following we will further explore the structure of steps (S1)-(S2) to prove that DDPM implements them and that, as a consequence, solves the original problem (2). We start by explicitly rephrasing update (12) by plugging in the definition of q^i, given in (6), thus obtaining the following max-min optimization problem

$$\max_{\mathbf{1}^\top\boldsymbol{\mu}^i = 1, \boldsymbol{\mu}^i \succeq \mathbf{0}} \Big(\min_{\mathbf{x}^i \in X^i} \sum_{s=1}^S \boldsymbol{\mu}_s^i \Big(g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s \Big) \Big). \quad (14)$$

Notice that (14) is a local problem at each node i once $\boldsymbol{\lambda}^{ij}(t)$ and $\boldsymbol{\lambda}^{ji}(t)$ for all $j \in \mathcal{N}_i$ are given.

The next lemma is a first instrumental result.

Lemma 3.1: Consider the optimization problem

$$\max_{\boldsymbol{\mu}^i} \; \sum_{s=1}^S \boldsymbol{\mu}_s^i \Big(g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s \Big) \quad \text{subj. to } \mathbf{1}^\top\boldsymbol{\mu}^i = 1, \boldsymbol{\mu}^i \succeq \mathbf{0}, \quad (15)$$

with given $\mathbf{x}^i$, $\boldsymbol{\lambda}^{ij}(t)$ and $\boldsymbol{\lambda}^{ji}(t)$, $j \in \mathcal{N}_i$. Then, the problem

$$\min_{P^i} \; P^i \quad \text{subj. to } g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s \leq P^i, s \in \{1, \dots, S\} \quad (16)$$

is dual of (15) and strong duality holds.

Proof: First, since $\mathbf{x}^i$ (as well as $\boldsymbol{\lambda}^{ij}(t)$ and $\boldsymbol{\lambda}^{ji}(t)$) is given, problem (15) is a feasible *linear* program (the simplex constraint is nonempty) and, thus, strong duality holds. Introducing a scalar multiplier P^i associated to the constraint $\mathbf{1}^\top \boldsymbol{\mu}^i = 1$, we write the partial Lagrangian of (15)

$$\mathcal{L}_3(\boldsymbol{\mu}^i, P^i) = \sum_{s=1}^{S} \boldsymbol{\mu}_s^i \Big(g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s \Big) + P^i(1 - \mathbf{1}^\top \boldsymbol{\mu}^i)$$

and rearrange it as

$$\mathcal{L}_3(\boldsymbol{\mu}^i, P^i) = \sum_{s=1}^{S} \boldsymbol{\mu}_s^i \Big(g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s - P^i \Big) + P^i.$$

The dual function $\max_{\boldsymbol{\mu}^i \succeq \mathbf{0}} \mathcal{L}_3(\boldsymbol{\mu}^i, P^i)$ is equal to P^i with domain given by the inequalities $P^i \geq g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s$, $s \in \{1, \ldots, S\}$. Thus, the dual problem is obtained by maximizing the dual function over its domain giving (16), so that the proof follows. □

The next lemma is a second instrumental result.

Lemma 3.2: Max-min optimization problem (14) is the saddle point problem associated to problem (3)

$$\min_{\mathbf{x}^i, P^i} \; P^i$$
$$\text{subj. to } \mathbf{x}^i \in X^i$$
$$g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} \big(\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t)\big)_s \leq P^i, s \in \{1, \ldots, S\}.$$

Moreover, a primal-dual optimal solution pair of (3), call it $((\mathbf{x}^i(t+1), P^i(t+1)), \boldsymbol{\mu}^i(t+1))$, exists and $(\mathbf{x}^i(t+1), \boldsymbol{\mu}^i(t+1))$ is a solution of (14).

Proof: We give a constructive proof which clarifies how the problem (3) is derived from (14). Define

$$\phi(\mathbf{x}^i, \boldsymbol{\mu}^i) := \sum_{s=1}^{S} \boldsymbol{\mu}_s^i \Big(g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s \Big)$$

and note that (i) $\phi(\cdot, \boldsymbol{\mu}^i)$ is closed and convex for all $\boldsymbol{\mu}^i \succeq \mathbf{0}$ (affine transformation of a convex function with compact domain X^i) and (ii) $\phi(\mathbf{x}^i, \cdot)$ is closed and concave since it is a linear function with compact domain $\{\boldsymbol{\mu}^i \succeq \mathbf{0} \mid \mathbf{1}^\top \boldsymbol{\mu}^i = 1\}$, for all $\mathbf{x}^i \in \mathbb{R}^S$. Thus, we can invoke [25, Propositions 4.3] which allows us to switch max and min operators, and write

$$\max_{\mathbf{1}^\top \boldsymbol{\mu}^i = 1, \boldsymbol{\mu}^i \succeq \mathbf{0}} \Big(\min_{\mathbf{x}^i \in X^i} \sum_{s=1}^{S} \boldsymbol{\mu}_s^i \Big(g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s \Big) \Big)$$
$$= \min_{\mathbf{x}^i \in X^i} \Big(\max_{\mathbf{1}^\top \boldsymbol{\mu}^i = 1, \boldsymbol{\mu}^i \succeq \mathbf{0}} \sum_{s=1}^{S} \boldsymbol{\mu}_s^i \Big(g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s \Big) \Big). \tag{17}$$

Since the inner maximization problem depends nonlinearly on $\mathbf{x}^i$ (which is itself an optimization variable), it cannot be performed without also considering the minimization over $\mathbf{x}^i$. We overcome this issue by substituting the inner maximization problem with its equivalent dual minimization. In fact, by Lemma 3.1 we can rephrase the right hand side of (17) as

$$\min_{\mathbf{x}^i \in X^i} \Big(\min_{\substack{P^i \,:\, g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t))_s \leq P^i \\ s \in \{1, \ldots, S\}}} P^i \Big). \tag{18}$$

At this point, a *joint* (constrained) minimization with respect to $\mathbf{x}^i$ and P^i can be performed leading to problem (3).

To prove the second part, namely that a primal-dual optimal solution pair exists and solves problem (14), we first notice that problem (3) is convex. Indeed, the cost function is linear and the constraints are convex (X^i is convex as well as the functions $g_s^i(\mathbf{x}_s^i) + \sum_{j\in\mathcal{N}_i} \big(\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t)\big)_s$ and $-P^i$). Then, exploiting the structure of problem (3), it can be shown to satisfy Slater's constraint qualification, so that strong duality holds. Thus, a primal-dual optimal solution pair $((\mathbf{x}^i(t+1), P^i(t+1)), \boldsymbol{\mu}^i(t+1))$ exists and from the previous arguments $(\mathbf{x}^i(t+1), \boldsymbol{\mu}^i(t+1))$ solves (14), thus concluding the proof. □

We point out that from the previous lemma the minimization in (3) is shown to be equivalent to performing step (S1).

An important consequence of Lemma 3.1 is that each iteration of the algorithm can be in fact performed (since a primal-dual optimal solution pair of (3) exists). Notice that the solvability of problem (3) at each iteration t is equivalent to the boundedness of subgradients of η, which is ensured by the compactness of the simplex constraint in (12).

The next corollary is a byproduct of the proof of Lemma 3.2.

Corollary 3.3: Let $P^i(t+1)$, for each $i \in \{1, \ldots, N\}$, be the optimal cost of problem (3) with fixed values $\{\boldsymbol{\lambda}^{ij}(t), \boldsymbol{\lambda}^{ji}(t)\}_{j\in\mathcal{N}_i}$. Then, it holds that

$$P^i(t+1) = \eta^i(\{\boldsymbol{\lambda}^{ij}(t), \boldsymbol{\lambda}^{ji}(t)\}_{j\in\mathcal{N}_i}), \; \forall\, t \geq 0,$$

where η^i is defined in (10).

Proof: To prove the corollary, we first rewrite explicitly the definition of $\eta^i(\{\boldsymbol{\lambda}^{ij}(t), \boldsymbol{\lambda}^{ji}(t)\}_{j\in\mathcal{N}_i})$ given in (10), i.e.,

$$\eta^i(\{\boldsymbol{\lambda}^{ij}(t), \boldsymbol{\lambda}^{ji}(t)\}_{j\in\mathcal{N}_i}) = \max_{\mathbf{1}^\top \boldsymbol{\mu}^i = 1, \boldsymbol{\mu}^i \succeq \mathbf{0}} \Big(\Big(\min_{\mathbf{x}^i \in X^i} \sum_{s=1}^{S} \boldsymbol{\mu}_s^i g_s^i(\mathbf{x}_s^i) \Big) + {\boldsymbol{\mu}^i}^\top \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t)) \Big). \tag{19}$$

Then, being $P^i(t+1)$ the optimal cost of problem (3), it is also the optimal cost of problem (18), which is equivalent to the right hand side of equation (17). The proof follows by noting that the expression of η^i in (19) is exactly the left hand side of (17) after rearranging some terms. □

We are now ready to state the main result of the paper, namely that DDPM generates sequences that asymptotically converge to primal feasible optimal solutions of problem (2).

Theorem 3.4: Let $\{(\mathbf{x}^i(t), P^i(t))\}_{t\geq 0}$, $i \in \{1, \ldots, N\}$, be a sequence generated by the DDPM distributed algorithm, with $\gamma(t)$ satisfying (13). Then, the following holds:

(i) the sequence $\big\{ \sum_{i=1}^{N} P^i(t) \big\}_{t\geq 0}$ converges to the optimal cost $P^\star$ of problem (1), and

(ii) every limit point of the primal sequence $\{\mathbf{x}^i(t)\}_{t\geq 0}$, with $i \in \{1, \ldots, N\}$, is an optimal (thus feasible) solution of (1).

Proof: We prove the theorem in two main steps. We start with the convergence of $\{\sum_{i=1}^{N} P^i(t)\}_{t\geq 0}$ to the optimal cost by properly combining the duality results given in the previous lemmas. Second, by exploiting the network connectivity and

the epigraph structure of the problem, we also show that any limit point of $\{\mathbf{x}^i(t)\}_{t\geq 0}$ is primal feasible, i.e., satisfies the coupling constraints.

For each $i \in \{1,\dots,N\}$, let $\{\boldsymbol{\mu}^i(t)\}_{t\geq 0}$, and $\{\boldsymbol{\lambda}^{ij}(t)\}_{t\geq 0}$, $j \in \mathcal{N}_i$, be the auxiliary sequences defined by DDPM associated to $\{(\mathbf{x}^i(t), P^i(t))\}_{t\geq 0}$. From Lemma 3.2 a primal-dual optimal solution pair $((\mathbf{x}^i(t+1), P^i(t+1)), \boldsymbol{\mu}^i(t+1))$ of (3) in fact exists (so that the distributed algorithm is well-posed) and $(\mathbf{x}^i(t+1), \boldsymbol{\mu}^i(t+1))$ solves (14). Recalling that solving (14) is equivalent to solving (12), it follows that $\boldsymbol{\mu}^i(t+1)$ in the DDPM implements step (S1) of the dual subgradient (S1)-(S2). Moreover, noting that update (4) of $\boldsymbol{\lambda}^{ij}$ is exactly step (S2), it follows that DDPM is an operative way to implement the dual subgradient algorithm (S1)-(S2). We have shown that algorithm (S1)-(S2) converges in objective value, that is

$$\lim_{t\to\infty} \sum_{i=1}^{N} \eta^i(\{\boldsymbol{\lambda}^{ij}(t), \boldsymbol{\lambda}^{ji}(t)\}_{j\in\mathcal{N}_i}) = \eta^\star = P^\star,$$

where the second equality follows from the properties of problem (9). Then, by Corollary 3.3 we have

$$\sum_{i=1}^{N} P^i(t+1) = \sum_{i=1}^{N} \eta^i(\{\boldsymbol{\lambda}^{ij}(t), \boldsymbol{\lambda}^{ji}(t)\}_{j\in\mathcal{N}_i}), \qquad \forall\, t \geq 0,$$

so that $\lim_{t\to\infty} \sum_{i=1}^{N} P^i(t) = P^\star$, thus concluding the proof of the first statement.

To prove the second statement, we show that every limit point of the (primal) sequence $\{\mathbf{x}^i(t)\}_{t\geq 0}$, $i \in \{1,\dots,N\}$, is feasible and optimal for problem (1). Let us start by summing both sides of inequality constraints in (3) over $i \in \{1,\dots,N\}$ for all $t \geq 0$, obtaining

$$\sum_{i=1}^{N} \Big(g_s^i(\mathbf{x}_s^i(t)) - \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t-1) - \boldsymbol{\lambda}^{ji}(t-1))_s \Big) \leq \sum_{i=1}^{N} P^i(t). \tag{20}$$

Let us denote by a_{ij} the (i,j)-th entry of the adjacency matrix associated to the undirected graph $\mathcal{G}$. Then, we can write

$$\begin{aligned}\sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t)) &= \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}(\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t)) \\ &= \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\boldsymbol{\lambda}^{ij}(t) - \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}\boldsymbol{\lambda}^{ji}(t).\end{aligned}$$

Since $\mathcal{G}$ is undirected $a_{ij} = a_{ji}$ for all $(i,j) \in \mathcal{E}$ it follows that the two summations in the latter equation are identical, implying that $\sum_{i=1}^{N} \sum_{j\in\mathcal{N}_i} (\boldsymbol{\lambda}^{ij}(t) - \boldsymbol{\lambda}^{ji}(t)) = 0$ for all $t \geq 0$. Hence, (20) reduces to

$$\sum_{i=1}^{N} g_s^i(\mathbf{x}_s^i(t)) \leq \sum_{i=1}^{N} P^i(t), \tag{21}$$

for all $s \in \{1,\dots,S\}$ and $t \geq 0$.

By construction $\{[\mathbf{x}^1(t),\dots,\mathbf{x}^N(t)]\}_{t\geq 0}$ is a bounded sequence since $\mathbf{x}^i(t) \in X^i$ (compact set) for all $i \in \{1,\dots,N\}$. Then there exists a convergent sub-sequence $\{[\mathbf{x}^1(t_n),\dots,\mathbf{x}^N(t_n)]\}_{n\geq 0}$, with $[\bar{\mathbf{x}}^1,\dots,\bar{\mathbf{x}}^N] = \lim_{n\to\infty}[\mathbf{x}^1(t_n),\dots,\mathbf{x}^N(t_n)]$ being its limit point. Since each g_s^i is a (finite) convex function over $\mathbb{R}$, it is also continuous over any compact subset of $\mathbb{R}$. By taking the limit of (21), it holds

$$\begin{aligned}\lim_{n\to\infty} \sum_{i=1}^{N} g_s^i(\mathbf{x}_s^i(t_n)) &= \sum_{i=1}^{N} g_s^i\Big(\lim_{n\to\infty} \mathbf{x}_s^i(t_n)\Big) \\ &= \sum_{i=1}^{N} g_s^i(\bar{\mathbf{x}}_s^i) \leq \lim_{n\to\infty} \sum_{i=1}^{N} P^i(t_n) = P^\star,\end{aligned} \tag{22}$$

for all $s \in \{1,\dots,S\}$, where the last inequality follows from (21) and the last equality follows from the first statement of the theorem. Since the sub-sequence $\{\mathbf{x}^i(t_n)\}_{n\geq 0}$ is arbitrary, equation (22) shows that every limit point $\bar{\mathbf{x}}^i$, $i \in \{1,\dots,N\}$, is feasible, i.e., it satisfies both the local constraint (by construction) and the coupling constraints.

To prove optimality, let us introduce the sequence $\{P(t)\}_{t\geq 0}$ defined as

$$P(t) := \max_{s\in\{1,\dots,S\}} \sum_{i=1}^{N} g_s^i(\mathbf{x}_s^i(t)), \ \forall\, t \geq 0, \tag{23}$$

and compute a lower and an upper bound for it.

In order to lower bound $P(t)$, we notice that it is also the cost of problem (2) associated to $[\mathbf{x}^1(t),\dots,\mathbf{x}^N(t), P(t)]$. Since for all $i \in \{1,\dots,N\}$ each $\mathbf{x}^i(t) \in X^i$, then their cost $P(t)$ must be higher than the optimal one, that is

$$P^\star \leq P(t), \quad \forall\, t \geq 0.$$

As for the upper bound, since (21) holds for all $s \in \{1,\dots,S\}$, then it holds also for the maximum over s, giving

$$P^\star \leq P(t) = \max_{s\in\{1,\dots,S\}} \sum_{i=1}^{N} g_s^i(\mathbf{x}_s^i(t)) \leq \sum_{i=1}^{N} P^i(t). \tag{24}$$

Since the sequences $\{\mathbf{x}^i(t)\}_{t\geq 0}$, $i \in \{1,\dots,N\}$, are bounded, consider convergent sub-sequences $\{\mathbf{x}^i(t_n)\}_{n\geq 0}$ (with limit points $\bar{\mathbf{x}}^i$). Taking the limit as $n \to \infty$ in (24), we get

$$P^\star \leq \lim_{n\to\infty} \Big(\max_{s\in\{1,\dots,S\}} \sum_{i=1}^{N} g_s^i(\mathbf{x}_s^i(t_n)) \Big) \leq \lim_{n\to\infty} \sum_{i=1}^{N} P^i(t_n) = P^\star.$$

By noting that the maximization is over a finite set and recalling that g_s^i is continuous over any compact subset of $\mathbb{R}$, it follows $P^\star \leq \max_{s\in\{1,\dots,S\}} \sum_{i=1}^{N} g_s^i(\bar{\mathbf{x}}_s^i) \leq P^\star$ proving that any limit point $\bar{\mathbf{x}}^i$, $i \in \{1,\dots,N\}$, is also optimal, thus concluding the proof. □

IV. Numerical Simulations

In this section we propose a numerical study to show the performance of our algorithm. We also compare our approach with a plain distributed subgradient algorithm [23] applied to the dual problem (5). We consider a network of Thermostatically Controlled Loads (TCLs) (such as air conditioners, heat pumps, electric water heaters), [26]. The discrete-time dynamical model of the i-th device is given by

$$T_{s+1}^i = T_s^i e^{-\alpha\Delta\tau} + \Big(1 - e^{-\alpha\Delta\tau}\Big)\Big(\frac{Q}{\alpha}\mathbf{x}_s^i + \frac{\delta_s^i}{\alpha} + T_{out,s}^i\Big), \tag{25}$$

where T_s^i is the temperature, $\alpha > 0$ is a parameter depending on geometric and thermal characteristics, $\Delta\tau > 0$ models the sampling interval, $T_{out,s}^i$ is the air temperature outside the device, δ_s^i represents a known time-varying forcing term

onto the internal temperature of the device, $\mathbf{x}_s^i \in [0,1]$ is the control input, and $Q > 0$ is a scaling factor. Moreover, we constrain the temperature to stay within a given interval $[T_{min}, T_{max}]$ with $T_{max} > T_{min} \geq 0$. The constraints due to the dynamics and the bounds on the temperature can be written in a compact form as inequality constraints on the input in the form $A_i \mathbf{x}^i \leq b_i$, for each agent $i \in \{1,\dots,N\}$, for some A_i and b_i depending on (25). Finally, we assume that the power consumption $g_s^i(\mathbf{x}_s^i)$ of the i-th device is directly proportional to $\mathbf{x}_s^i$, i.e., $g_s^i(\mathbf{x}_s^i) = c^i \mathbf{x}_s^i$.

Thus, optimization problem (2) for this scenario is

$$\begin{aligned} \min_{\mathbf{x}^1,\dots,\mathbf{x}^N,P} \quad & P \\ \text{subj. to} \quad & A_i \mathbf{x}^i \preceq b_i,\ \mathbf{x}^i \in [0,1]^S, \quad i \in \{1,\dots,N\} \\ & \sum_{i=1}^N c^i \mathbf{x}_s^i \leq P, \qquad\qquad s \in \{1,\dots,S\}. \end{aligned} \tag{26}$$

We consider a horizon of $S = 60$ and we choose each δ_s^i in (25) to be constant for an interval of 5 slots and zero otherwise. The nonzero values are set in the central part of the horizon $\{1,\dots,S\}$ by randomly shifting the center. Then, we randomly choose the heterogeneous power consumption coefficient $c^i \in \mathbb{R}$ of each device from a set of five values, drawn from a uniform distribution in $[1,3]$. We consider $N = 20$ agents communicating according to an undirected connected Erdős-Rényi random graph $\mathcal{G}$ with parameter 0.2. We use a diminishing step-size in the form $\gamma(t) = t^{-0.65}$.

In Figure 2 we show the evolution at each algorithm iteration t of the local objective functions $P^i(t)$, $i \in \{1,\dots,N\}$, (solid lines) which converge to stationary values. Moreover, we also plot their sum $\sum_{i=1}^N P^i(t)$ (dashed line) and the value $P(t)$ (dotted line), introduced in (23). As proven in Corollary 3.3, both of them asymptotically converge to the centralized optimal cost $P^\star$ of problem (26). It is worth noting that, at each iteration t, the curve $P(t)$ stays above the optimal value $P^\star$ and below the curve $\sum_{i=1}^N P^i(t)$, i.e., condition (24) is satisfied.

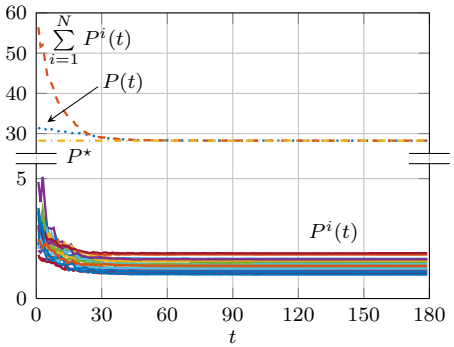

Fig. 2. Evolution of $P^i(t)$, $i \in \{1,\dots,N\}$, (solid lines), their sum $\sum_{i=1}^N P^i(t)$ (dashed line), $P(t)$ (dotted line) and (centralized) optimal cost $P^\star$ (dash-dotted line).

In Figure 3 (left) the local solutions at the last algorithm iteration are depicted. We denote them $\mathbf{x}^{i\star}$, $i \in \{1,\dots,N\}$, to highlight that they satisfy the cost optimality up to the required tolerance 10^{-3}. We also plot the resulting aggregate optimal consumption, i.e., $\sum_{i=1}^N c^i \mathbf{x}^{i\star}$, which, as expected, in fact shaves off the power demand peak. Moreover, the optimal local solutions satisfy the box constraint $[0,1]$ for each slot $s \in \{1,\dots,S\}$. In fact, as we have proven, the algorithm converges in an interior point fashion, i.e., the local constraint at each node $i \in \{1,\dots,N\}$, is satisfied at all the algorithm iterations and in Figure 3 (right) we depict, as an example, the behavior of the components of $\mathbf{x}^1(t)$.

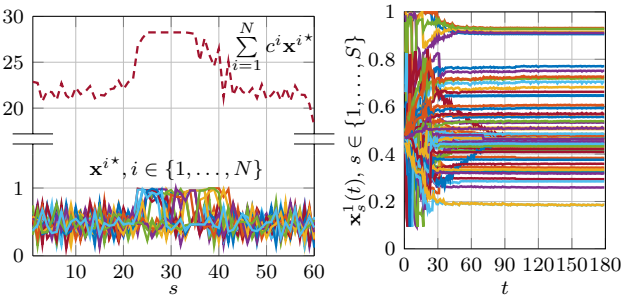

Fig. 3. Left: profile of optimal solutions $\mathbf{x}^{i\star}$ (solid lines), and $\sum_{i=1}^N c^i \mathbf{x}_s^{i\star}$ (dashed line) over the horizon $\{1,\dots,S\}$. Right: algorithmic evolution of $\mathbf{x}^1$.

In Figure 4 (left) we show, the violation of the coupling constraints, for all $s \in \{1,\dots,S\}$ at each iteration t. As expected, the violations asymptotically go to nonnegative values, consistently with the claimed primal recovery feature enjoyed by our distributed algorithm, proven in Theorem 3.4. In Figure 4 (right) the difference $\sum_{i=1}^N \left(g_s^i(\mathbf{x}_s^i(t)) - P^i(t)\right)$ is also shown, which is always nonnegative consistently with equation (21).

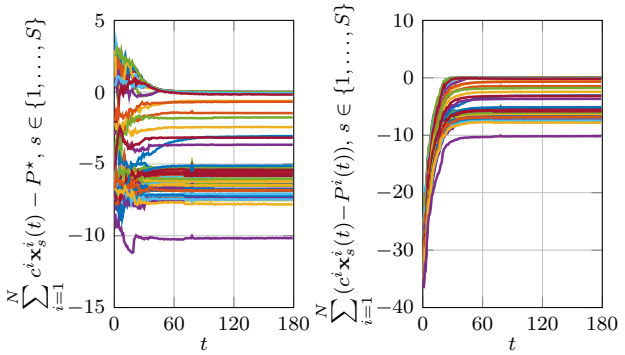

Fig. 4. Evolution of primal violations of solutions $\mathbf{x}^i(t)$, $i \in \{1,\dots,N\}$.

Finally, we propose a numerical comparison between our DDPM distributed algorithm and a plain distributed subgradient method, see [23], applied to the dual of the peak-minimization problem (2). In particular, agents agree on a solution of (5) and build the averaging primal sequence to recover a primal feasible solution. Formally, at every iteration t, each agent i locally computes a dual subgradient of its own q^i at its current dual solution estimate $\boldsymbol{\mu}_{\text{SUB}}^i(t)$ as

$$\frac{\tilde{\partial} q^i(\boldsymbol{\mu}_{\text{SUB}}^i(t))}{\partial \boldsymbol{\mu}^i} = [g_1^i(\mathbf{x}_{1,\text{SUB}}^i(t)) \cdots g_S^i(\mathbf{x}_{S,\text{SUB}}^i(t))]^\top - P_{\text{SUB}}^i(t)\mathbf{1},$$

where $\mathbf{x}_{\text{SUB}}^i(t) \in \operatorname{argmin}_{\mathbf{x}^i \in X^i} \sum_{s=1}^S \boldsymbol{\mu}_{s,\text{SUB}}^i(t) g_s^i(\mathbf{x}_s^i)$ and $P_{\text{SUB}}^i(t) = \max_{s \in \{1,\dots,S\}} g_s^i(\mathbf{x}_{s,\text{SUB}}^i(t))$. Then, it receives

$\boldsymbol{\mu}^j_{\text{SUB}}(t)$ from its neighbors $\mathcal{N}_i$ and performs the distributed subgradient iteration given by

$$\boldsymbol{\mu}^i_{\text{SUB}}(t+1) = \mathcal{P}_{\text{SIMPLEX}}\left[\sum_{j\in\mathcal{N}_i} w^i_j\, \boldsymbol{\mu}^j_{\text{SUB}}(t) + \gamma(t)\frac{\tilde{\partial} q^i(\boldsymbol{\mu}^i_{\text{SUB}}(t))}{\partial \boldsymbol{\mu}^i}\right],$$

where $\mathcal{P}_{\text{SIMPLEX}}$ denotes the Euclidean projection onto the simplex $\{\boldsymbol{\mu} \succeq \mathbf{0} \mid \mathbf{1}^\top\boldsymbol{\mu} = 1\}$ and w^i_j are entries of a doubly stochastic matrix matching the communication graph. The primal sequences computed through the running average are $\widehat{\mathbf{x}}^i(t) = \frac{1}{t}\sum_{\tau=0}^{t}\mathbf{x}^i_{\text{SUB}}(\tau)$, $i \in \{1,\dots,N\}$.

We randomly generate 50 problem instances and run both algorithms to evaluate the cost errors $|P^\star - \sum_{i=1}^N \max_{s\in\{1,\dots,S\}} g^i_s(\mathbf{x}^i_s(t))|$, for the DDPM, and $|P^\star - \sum_{i=1}^N \max_{s\in\{1,\dots,S\}} g^i_s(\widehat{\mathbf{x}}^i_{s,\text{SUB}}(t))|$, for the dual distributed subgradient. As shown in Figure 5, our DDPM algorithm is faster than the plain distributed subgradient. Moreover, DDPM appears to reach finite time convergence for this problem set-up (in the 92% of the cases within 2000 iterations, while in all the other cases in no more than 9000). On the other hand, the plain distributed subgradient after 10000 iterations is still at an accuracy of about 10^{-1}.

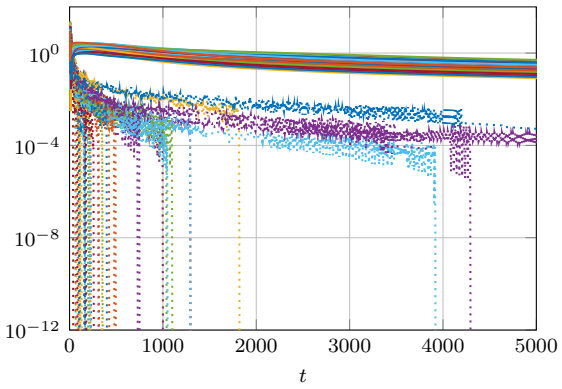

Fig. 5. Evolution of the cost error for the DDPM (dotted lines) and for the plain distributed subgradient method (solid lines).

V. Conclusions

In this paper we studied a distributed min-max optimization framework motivated by peak minimization problems in smart grids. Standard distributed optimization algorithms cannot be applied to this problem set-up due to a coupling in the objective function and in the constraints. We proposed a distributed algorithm based on the combination of duality methods and properties from min-max optimization. Despite a complex derivation, the algorithm has a very simple structure at each node. Moreover, we showed that the primal variable estimate at each node converges to the corresponding component of the optimal solution. Theoretical results are corroborated by a numerical example on peak minimization in Demand Side Management showing that our approach is significantly faster that a plain distributed subgradient algorithm applied to the dual problem.

References

[1] I. Notarnicola, M. Franceschelli, and G. Notarstefano, "A duality-based approach for distributed min-max optimization with application to demand side management," in *IEEE Conf. on Decision and Control (CDC)*, 2016, pp. 1877–1882.

[2] M. Alizadeh, X. Li, Z. Wang, A. Scaglione, and R. Melton, "Demand-side management in the smart grid: Information processing for the power switch," *IEEE Signal Processing Magaz.*, vol. 29, no. 5, pp. 55–67, 2012.

[3] A.-H. Mohsenian-Rad, V. W. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Trans. on Smart Grid*, vol. 1, no. 3, pp. 320–331, 2010.

[4] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa, "Demand-side management via distributed energy generation and storage optimization," *IEEE Trans. on Smart Grid*, vol. 4, no. 2, pp. 866–876, 2013.

[5] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1439–1451, 2006.

[6] B. Yang and M. Johansson, "Distributed optimization and games: a tutorial overview," in *Netw.Control Sys.* Springer, 2010, pp. 109–148.

[7] M. Zhu and S. Martínez, "On distributed convex optimization under inequality and equality constraints," *IEEE Trans. on Automatic Control*, vol. 57, no. 1, pp. 151–164, 2012.

[8] H. D. Sherali and G. Choi, "Recovery of primal solutions when using subgradient optimization methods to solve lagrangian duals of linear programs," *Operations Research Lett.*, vol. 19, no. 3, pp. 105–113, 1996.

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

[10] J. Lu and M. Johansson, "Convergence analysis of primal solutions in dual first-order methods," in *IEEE Conf. on Decision and Control (CDC)*, 2013, pp. 6861–6867.

[11] E. Gustavsson, M. Patriksson, and A.-B. Strömberg, "Primal convergence from dual subgradient methods for convex optimization," *Mathematical Programming*, vol. 150, no. 2, pp. 365–390, 2015.

[12] T.-H. Chang, A. Nedić, and A. Scaglione, "Distributed constrained optimization by consensus-based primal-dual perturbation method," *IEEE Trans. on Automatic Control*, vol. 59, no. 6, pp. 1524–1538, 2014.

[13] A. Simonetto and H. Jamali-Rad, "Primal recovery from consensus-based dual decomposition for distributed convex optimization," *J. of Optim. Theory and Applications*, vol. 168, no. 1, pp. 172–197, 2016.

[14] A. Falsone, K. Margellos, S. Garatti, and M. Prandini, "Dual decomposition for multi-agent distributed optimization with coupling constraints," *Automatica*, vol. 84, pp. 149–158, 2017.

[15] A. Nedić and A. Ozdaglar, "Subgradient methods for saddle-point problems," *J. of Optim. Theory and Applications*, vol. 142, no. 1, pp. 205–228, 2009.

[16] K. Srivastava, A. Nedić, and D. Stipanović, "Distributed min-max optimization in networks," in *IEEE Conf. on Digital Signal Processing (DSP)*, 2011, pp. 1–8.

[17] D. Mateos-Núñez and J. Cortés, "Distributed subgradient methods for saddle-point problems," in *IEEE Conf. on Decision and Control (CDC)*, 2015, pp. 5462–5467.

[18] A. Simonetto, T. Keviczky, and M. Johansson, "A regularized saddle-point algorithm for networked optimization with resource allocation constraints," in *IEEE Conf. on Decision and Control (CDC)*, 2012, pp. 7476–7481.

[19] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "Regret bounds of a distributed saddle point algorithm," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 2969–2973.

[20] G. Notarstefano and F. Bullo, "Distributed abstract optimization via constraints consensus: Theory and applications," *IEEE Trans. on Automatic Control*, vol. 56, no. 10, pp. 2247–2261, 2011.

[21] M. Bürger, G. Notarstefano, and F. Allgöwer, "A polyhedral approximation framework for convex and robust distributed optimization," *IEEE Trans. on Automatic Control*, vol. 59, no. 2, pp. 384–395, 2014.

[22] D. P. Bertsekas, *Nonlinear programming*. Athena Scient., 1999.

[23] A. Nedić and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[24] D. P. Bertsekas, *Convex Optimization Algorithms*. Athena Scient., 2015.

[25] ——, "Min common/max crossing duality: A geometric view of conjugacy in convex optimization," *Lab. for Information and Decision Systems, MIT, Report LIDS-P-2796*, 2009.

[26] M. Alizadeh, A. Scaglione, A. Applebaum, G. Kesidis, and K. Levitt, "Reduced-order load models for large populations of flexible appliances," *IEEE Trans. on Power Sys.*, vol. 30, no. 4, pp. 1758–1774, 2015.