Distributed Constraint-Coupled Optimization over Random Time-Varying Graphs via Primal Decomposition and Block Subgradient Approaches

This is the post peer-review accepted manuscript of:

A. Camisa, F. Farina, I. Notarnicola and G. Notarstefano, "Distributed Constraint-Coupled Optimization over Random Time-Varying Graphs via Primal Decomposition and Block Subgradient Approaches," 2019 IEEE Conference on Decision and Control (CDC), Nice, France, 2019, pp. 6374-6379.

Distributed Constraint-Coupled Optimization over Random Time-Varying Graphs via Primal Decomposition and Block Subgradient Approaches

Andrea Camisa, Francesco Farina, Ivano Notarnicola, Giuseppe Notarstefano

***Abstract*— In this paper, we consider a network of processors that want to cooperatively solve a large-scale, convex optimization problem. Each processor has knowledge of a local cost function that depends only on a local variable. The goal is to minimize the sum of the local costs, while making the variables satisfy both local constraints and a global coupling constraint. We propose a simple, fully distributed algorithm, that works in a random, time-varying communication model, where at each iteration multiple edges are randomly drawn from an underlying graph. The algorithm is interpreted as a primal decomposition scheme applied to an equivalent problem reformulation. Almost sure convergence to the optimal cost of the original problem is proven by resorting to approaches from block subgradient methods. Specifically, the communication structure is mapped to a block structure, where the blocks correspond to the graph edges and are randomly selected at each iteration. Moreover, an almost sure asymptotic primal recovery property, with no averaging mechanisms, is shown. A numerical example corroborates the theoretical analysis.**

I. Introduction

This paper addresses a challenging optimization scenario arising in distributed contexts, where independent smart devices can communicate with each other to cooperatively perform tasks. For instance, in distributed optimal control, one usually considers several, spatially distributed systems that must be controlled. The optimization set-up arising from such contexts is large scale, i.e., consists of several optimization variables, each one constrained in a local set, and the overall goal is to minimize the sum of local cost functions, where each one depends only on the local variable. The problem becomes challenging when additional coupling constraints among the variables are considered. This optimization set-up has been widely investigated in the distributed optimization literature, and is usually tackled by means of so-called dual decomposition approaches, which are built up as distributed optimization algorithms applied to the dual of the considered problem.

In [1] a consensus-based primal-dual perturbation algorithm is proposed to solve smooth constraint-coupled optimization problems. A distributed saddle-point algorithm with laplacian averaging is proposed in [2] for a class of min-max problems. In [3], a distributed algorithm based on cutting planes is formulated. In the centralized literature, some effort has been spent in formulating primal recovery mechanisms (see, e.g., [4]), which allow one to recover feasibility of the primal solution from a dual optimal solution. These approaches have been then shifted to distributed optimization. In [5] a primal recovery mechanism for consensus-based dual decomposition is considered, whereas [6] considers a distributed dual algorithm, based on proximal minimization, with a primal recovery technique. Nevertheless, averaging mechanisms can slow down convergence of algorithms, thus we focus on an alternative approach, introduced in [7], where a distributed algorithm for constraint-coupled optimization, without averaging schemes, has been formulated. In this paper, starting from [7], we propose a distributed algorithm that can be run over random, time-varying graphs. The algorithm analysis resorts to randomized block subgradient methods (see, e.g., [8] for a survey). A unified framework for nonsmooth block methods is [9]. In [10], a randomized block coordinate descent method is formulated, while [11] investigates a stochastic block mirror descent method with random block update. Subgradient methods are among the first methodologies for distributed optimization. However, they were applied to a different problem set-up, where the objective is to minimize the sum of cost functions depending on a common optimization variable. Pioneering works are [12], [13], while a more up-to-date review is [14].

The contributions of this paper are as follows. We propose a distributed algorithm to solve constraint-coupled optimization problems over random, time-varying communication graphs, with underlying connected graph. This network set-up is general and can model several scenarios with, e.g., heterogeneous devices and communication links. The algorithm builds on a primal decomposition scheme applied to an equivalent, relaxed version of the problem. We prove that almost surely the objective value converges to the optimal cost and any limit point of the sequence of solution estimates is an optimal (hence feasible) solution of the given problem. To prove almost sure convergence, the problem is reformulated as an equivalent unconstrained problem through an appropriate, graph-induced change of variables. Then, the algorithm is recast as a randomized block subgradient method applied to the latter problem, with each block corresponding to an edge of the graph. We point out that convergence of the proposed algorithm requires to extend the analysis of block subgradient methods to the case in which multiple blocks are updated at once. Notice that the scheme studied here extends the algorithm in [7], which is formulated for fixed communication graphs and is derived by means of several duality steps (rather than resorting to

This result is part of a project that 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).

A. Camisa, F. Farina, I. Notarnicola and G. Notarstefano are with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy, a.camisa@unibo.it, franc.farina@unibo.it, ivano.notarnicola@unibo.it and giuseppe.notarstefano@unibo.it.

primal decomposition). The results of this paper pave the way for more general algorithms, to be implemented, e.g., in an asynchronous context, with unreliable communication.

The paper is organized as follows. In Section II, we introduce the problem set-up and we recall some preliminaries. In Section III, the random communication model and the distributed algorithm are introduced. A sketch of analysis is provided in Section IV, while Section V presents a numerical computation on a microgrid control problem. For the sake of space, all the proofs are omitted and will be provided in a future document.

II. Optimization Set-up and Preliminaries

In this section, we introduce the problem set-up investigated in this paper, together with some preliminaries.

A. Constraint-Coupled Optimization

Consider the *constraint-coupled* optimization problem

$$\begin{aligned} \min_{\mathbf{x}_1,\dots,\mathbf{x}_N} \ & \sum_{i=1}^N f_i(\mathbf{x}_i) \\ \text{subj. to } & \sum_{i=1}^N \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{0}, \\ & \mathbf{x}_i \in X_i, \qquad i \in \{1,\dots,N\}, \end{aligned} \tag{1}$$

where, for all $i \in \{1,\dots,N\}$, $\mathbf{x}_i$ denotes the i-th portion of the optimization variable, $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$ depends only on the variable $\mathbf{x}_i$, $X_i \subset \mathbb{R}^{n_i}$ is a *local* constraint set for the variable $\mathbf{x}_i$ and $\mathbf{g}_i : \mathbb{R}^{n_i} \to \mathbb{R}^S$ is the i-th contribution to the (vector) *coupling* constraint $\sum_{i=1}^N \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{0}$. The symbol $\mathbf{0}$ denotes the vector of zeroes, and we also use the symbol $\leq$ (and consistently for other sides) to denote component-wise inequality for vectors. Throughout the paper, we make the following standing assumption.

Assumption 2.1: For all $i \in \{1,\dots,N\}$, the set X_i is non-empty, convex and compact, the function f_i is convex and each component of the function $\mathbf{g}_i$ is convex. □

By Assumption 2.1, the optimal cost of problem (1) is finite and at least one optimal solution exists. The following assumption is the well-known Slater's constraint qualification, and is a sufficient condition for the application of duality.

Assumption 2.2: There exist $\bar{\mathbf{x}}_1 \in X_1, \dots, \bar{\mathbf{x}}_N \in X_N$ such that $\sum_{i=1}^N \mathbf{g}_i(\bar{\mathbf{x}}_i) < \mathbf{0}$. □

We look for distributed algorithms to solve problem (1), that is, the problem must be solved by N processors (or agents) in a network, by means of local computation and communication with neighboring agents only. The communication model is formalized in Section III-A. Each node i knows only its local constraint X_i, its local cost function f_i and its own contribution $\mathbf{g}_i$ to the coupling constraints, and is only interested in computing its own portion $\mathbf{x}_i^\star$ of an optimal solution $(\mathbf{x}_1^\star, \dots, \mathbf{x}_N^\star)$ of problem (1).

B. Relaxation and Primal Decomposition

In this subsection, we recall a relaxation and primal decomposition approach for problem (1). Although the approach was introduced in [7], it was first interpreted as a primal decomposition scheme in [15].

To design distributed algorithms solving problem (1), it is worth exploiting the structure of the problem by resorting to tailored decomposition techniques. A typical choice is *dual* decomposition, in which the coupling constraints $\sum_{i=1}^N \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{0}$ are removed from the problem and a "price" for their violation is added to the cost function. By solving the so-called dual problem, one can obtain the optimal "price" for the coupling constraint violation. However, owing to Assumption 2.1, the cost functions are not strictly convex, so that averaging mechanisms, that are typically slow, are required to recover a solution of the original problem (see, e.g., [4]).

In this paper, we instead rely on a *primal* decomposition scheme, also called right-hand side allocation [16], [17]. In a primal decomposition scheme, the coupling constraints $\sum_{i=1}^N \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{0}$ are interpreted as a limited resource to be shared among nodes. Then, a two-level hierarchical structure is formulated, with several (independent) subproblems having a fixed resource allocation, and a master problem, coordinating the overall resource allocation process. We will apply such approach to an equivalent, relaxed version of problem (1), as this allows for the formulation of a distributed optimization algorithm.

Formally, consider the following relaxation of problem (1),

$$\begin{aligned} \min_{\mathbf{x}_1,\dots,\mathbf{x}_N,\rho} \ & \sum_{i=1}^N f_i(\mathbf{x}_i) + M\rho \\ \text{subj. to } & \sum_{i=1}^N \mathbf{g}_i(\mathbf{x}_i) \leq \rho\mathbf{1}, \\ & \rho \geq 0, \ \mathbf{x}_i \in X_i, \quad i \in \{1,\dots,N\}, \end{aligned} \tag{2}$$

where $M > 0$ is a scalar and $\mathbf{1}$ denotes the vector of ones. The variable ρ allows for a violation of the coupling constraints, but is penalized in the cost by the term $M\rho$. If the constant M is large enough, problem (2) is equivalent to (1), as formalized next.

Lemma 2.3: Let Assumptions 2.1 and 2.2 hold. Moreover, let M be such that $M > \|\boldsymbol{\mu}^\star\|_1$, with $\boldsymbol{\mu}^\star \in \mathbb{R}^S$ a Lagrange multiplier for problem (1) associated to the constraint $\sum_{i=1}^N \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{0}$. Then, the optimal solutions of the relaxed problem (2) are in the form $(\mathbf{x}_1^\star, \dots, \mathbf{x}_N^\star, 0)$, where $(\mathbf{x}_1^\star, \dots, \mathbf{x}_N^\star)$ is an optimal solution of (1), i.e., the solutions of (2) must have $\rho = 0$. Moreover, the optimal costs of (2) and (1) are equal.

Proof: See [7, Proposition III.3]. ■

Then, the primal decomposition scheme can be formulated as follows. For all $i \in \{1,\dots,N\}$ and $\mathbf{y}_i \in \mathbb{R}^S$, the i-th *subproblem* is

$$\begin{aligned} p_i(\mathbf{y}_i) \triangleq \min_{\mathbf{x}_i,\rho_i} \ & f_i(\mathbf{x}_i) + M\rho_i \\ \text{subj. to } & \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{y}_i + \rho_i\mathbf{1} \\ & \rho_i \geq 0, \ \mathbf{x}_i \in X_i, \end{aligned} \tag{3}$$

where $\mathbf{y}_i \in \mathbb{R}^S$ is the local *allocation* for node i and $p_i(\mathbf{y}_i)$ denotes the optimal cost of the problem for a fixed $\mathbf{y}_i$. The

local allocations are coordinated by the *master* problem, i.e.,

$$\begin{aligned} \min_{\mathbf{y}_1,\dots,\mathbf{y}_N} \quad & \sum_{i=1}^{N} p_i(\mathbf{y}_i) \\ \text{subj. to} \quad & \sum_{i=1}^{N} \mathbf{y}_i = \mathbf{0}. \end{aligned} \tag{4}$$

In the next, we will denote the cost function of (4) as $p(\mathbf{y}) = \sum_{i=1}^{N} p_i(\mathbf{y}_i)$, where $\mathbf{y} \in \mathbb{R}^{SN}$ is the vector stacking all $\mathbf{y}_i$. It is worth noting that the primal decomposition for problem (1) can be obtained by setting $\rho_i = 0$ in problem (3). However, if the primal decomposition scheme was applied to the original problem (1), then, in order to guarantee feasibility of the subproblems, the master problem (4) would have additional constraints that are difficult to include in a distributed algorithm. Viceversa, subproblem (3) is always feasible for all $\mathbf{y}_i \in \mathbb{R}^S$.

The following lemma establishes the equivalence between the master problem (4) and the relaxed problem (2).

Lemma 2.4 ([16]): Let Assumption 2.1 hold. Then, problems (2) and (4) are equivalent, in the sense that *(i)* the optimal costs are equal, *(ii)* if $(\mathbf{x}_1^\star, \dots, \mathbf{x}_N^\star)$ is an optimal solution of (2) and $(\mathbf{y}_1^\star, \dots, \mathbf{y}_N^\star)$ is an optimal solution of (4), then $(\mathbf{x}_i^\star, 0)$ is an optimal solution of (3), with $\mathbf{y}_i = \mathbf{y}_i^\star$, for all $i \in \{1, \dots, N\}$. □

III. Distributed Primal Decomposition

In this section, we first formalize the communication model, then we describe the distributed algorithm and give its convergence result.

A. Random Time-Varying Communication Model

We use a random, time-varying communication model for the N processors solving problem (1), which can model general set-ups (e.g., with heterogeneous devices and communication links). We assume there exists an underlying undirected, *connected* graph $\mathcal{G}_u = (\{1, \dots, N\}, \mathcal{E}_u)$, where $\mathcal{E}_u \subseteq \{1, \dots, N\} \times \{1, \dots, N\}$ denotes the set of edges, from which the actual graph edges are drawn. The agents are assumed to communicate according to an undirected, time-varying graph $\mathcal{G}^t = (\{1, \dots, N\}, \mathcal{E}^t)$, where $t \in \mathbb{N}$ denotes the (universal) time and $\mathcal{E}^t \subseteq \mathcal{E}_u$ is the set of edges at time t, randomly drawn from $\mathcal{E}_u$. If $(i, j) \in \mathcal{E}^t$, then also $(j, i) \in \mathcal{E}^t$, and nodes i and j can exchange information at time t. Denote by $\mathcal{N}_i^t$ the set of *neighbors* of node i in $\mathcal{G}^t$, i.e., $\mathcal{N}_i^t = \{j \in \{1, \dots, N\} \mid (i, j) \in \mathcal{E}^t\}$ (and, consistently, denote by $\mathcal{N}_{i,u}$ the set of neighbors of node i in $\mathcal{G}_u$).

To obtain the graph $\mathcal{E}^t$, it is assumed that, at each iteration t, a number of $b^t \leq |\mathcal{E}_u|/2$ undirected edges of the underlying graph $\mathcal{G}_u$ are randomly activated. Denote by ν_b the probability that exactly b undirected edges activate, for all $b \in \{1, \dots, |\mathcal{E}_u|/2\}$. Then, after the number of activated edges at time t is known, to obtain their identity, it is assumed that b^t undirected edges are drawn from $\mathcal{E}_u$. Denote by σ_{ij} the probability that the undirected edge (i, j) is activated, for all $i, j \in \{1, \dots, N\}$ with $j > i$. The probabilities ν_b and σ_{ij} are assumed to be constant for all t and the random variables modeling the number and the indices of the activated edges are assumed to be independent from each other. The probabilities ν_b can be arbitrary, while the edge probabilities σ_{ij} are assumed to be uniform, as formally stated next.

Assumption 3.1: It holds $\nu_b \geq 0$ for all $b \in \{1, \dots, |\mathcal{E}_u|/2\}$ and $\sum_{b=1}^{|\mathcal{E}_u|/2} \nu_b = 1$. Moreover, it holds $\sigma_{ij} = \frac{1}{|\mathcal{E}_u|/2}$ for all $i, j \in \{1, \dots, N\}$ with $j > i$. □

The assumption of uniform probabilities σ_{ij} can be relaxed to any distribution, however in this paper we focus on the uniform case.

B. Distributed Algorithm Description

In this subsection, we introduce our Distributed Primal Decomposition algorithm. Informally, the algorithm is as follows. Each node maintains a local allocation estimate that is iteratively adjusted based on the solution of a local problem and on the information gathered from neighbors. Eventually, the local allocations are meant to reach optimal solutions of problem (4), so that the solutions of the local problems are expected to approach optimal solutions of (1).

In order to gain insights, we first consider how a centralized optimization algorithm applied to the master problem (4) works. A projected subgradient method for problem (4) can be explicitly written as

$$\begin{aligned} \bar{\mathbf{y}}_i^{t+1} &= \bar{\mathbf{y}}_i^t + \alpha^t \big(\bar{\boldsymbol{\mu}}_i^t - \frac{1}{N}\sum_{j=1}^{N} \bar{\boldsymbol{\mu}}_j^t\big), \\ &= \bar{\mathbf{y}}_i^t + \alpha^t \frac{1}{N}\sum_{j=1}^{N} \big(\bar{\boldsymbol{\mu}}_i^t - \bar{\boldsymbol{\mu}}_j^t\big), \qquad i \in \{1, \dots, N\}, \end{aligned} \tag{5}$$

where α^t denotes the step-size and each $\bar{\boldsymbol{\mu}}_i^t$ is a Lagrange multiplier of problem (3), associated to the constraint $\mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{y}_i + \rho_i \mathbf{1}$, with $\mathbf{y}_i = \bar{\mathbf{y}}_i^t$ (here we are denoting centralized quantities with a bar). In fact, it can be shown, see, e.g., [17], that $-\bar{\boldsymbol{\mu}}_i^t$ is a subgradient of the function p_i at $\bar{\mathbf{y}}_i^t$, and $\bar{\boldsymbol{\mu}}_i^t - \frac{1}{N}\sum_{j=1}^{N} \bar{\boldsymbol{\mu}}_j^t$ results from of the (Euclidean) projection onto the constraint set of (4).

Clearly, the update direction in (5) requires the collection of $\bar{\boldsymbol{\mu}}_j^t$ from all the agents in the network. To obtain a distributed algorithm, we restrict the sum to the agents in the neighborhood only, i.e., the update (5) is replaced by

$$\mathbf{y}_i^{t+1} = \mathbf{y}_i^t + \alpha^t \sum_{j \in \mathcal{N}_i^t} \big(\boldsymbol{\mu}_i^t - \boldsymbol{\mu}_j^t\big), \quad i \in \{1, \dots, N\}, \tag{6}$$

where each $\boldsymbol{\mu}_i^t$ is a Lagrange multiplier of problem (3) with $\mathbf{y}_i = \mathbf{y}_i^t$ (the constant $1/N$ is absorbed in the step-size). Notice that, as in (5), the update (6) preserves the sum of $\mathbf{y}_i$, i.e., $\sum_{i=1}^{N} \mathbf{y}_i^{t+1} = \sum_{i=1}^{N} \mathbf{y}_i^t$, maintaining feasibility of the iterates provided that $\sum_{i=1}^{N} \mathbf{y}_i^0 = 0$. Update (6) has a sparse structure matching the (time-varying) communication graph, and reduces to (5) in the case of a (fixed) complete graph.

The Distributed Primal Decomposition algorithm is summarized in the next table from the perspective of node i.

Algorithm Distributed Primal Decomposition

Initialization: $\mathbf{y}_i^0$ such that $\sum_{i=1}^N \mathbf{y}_i^0 = \mathbf{0}$

Evolution: FOR $t = 0, 1, \dots$

Compute $((\mathbf{x}_i^t, \rho_i^t), \boldsymbol{\mu}_i^t)$ as a primal-dual optimal solution pair of

$$\begin{aligned} \min_{\mathbf{x}_i, \rho_i} \quad & f_i(\mathbf{x}_i) + M\rho_i \\ \text{subj. to} \quad \boldsymbol{\mu}_i : \ & \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{y}_i^t + \rho_i \mathbf{1} \\ & \mathbf{x}_i \in X_i, \ \ \rho_i \geq 0 \end{aligned} \tag{7}$$

Gather $\boldsymbol{\mu}_j^t$ from $j \in \mathcal{N}_i^t$ and update

$$\mathbf{y}_i^{t+1} = \mathbf{y}_i^t + \alpha^t \sum_{j \in \mathcal{N}_i^t} \left(\boldsymbol{\mu}_i^t - \boldsymbol{\mu}_j^t \right) \tag{8}$$

Note that, the algorithm in [7] can be reformulated as the Distributed Primal Decomposition over a fixed communication graph. However, the algorithm in [7] is derived by means of several duality steps rather than using primal decomposition.

Remark 3.2: The Distributed Primal Decomposition can be formulated through the use of the graph Laplacian matrix. For a single coupling constraint ($S = 1$), the update (8) reads

$$\begin{aligned} \mathbf{y}^{t+1} &= \mathbf{y}^t + \alpha^t L^t \boldsymbol{\mu}^t \\ &= \mathbf{y}^t - \alpha^t L^t \widetilde{\nabla} p(\mathbf{y}^t), \end{aligned}$$

where $L^t \in \mathbb{R}^{N \times N}$ is the time-varying Laplacian matrix of $\mathcal{G}^t$, $\boldsymbol{\mu}^t$ is the vector stacking all $\boldsymbol{\mu}_i^t$ and $\widetilde{\nabla} p(\mathbf{y}^t)$ denotes a subgradient of p at $\mathbf{y}^t$. Similarly, the centralized update (5) reads

$$\bar{\mathbf{y}}^{t+1} = \bar{\mathbf{y}}^t - \alpha^t L_{\text{CO}} \widetilde{\nabla} p(\bar{\mathbf{y}}^t).$$

where the constant $1/N$ is absorbed in the step size and L_{CO} is the laplacian matrix of the complete graph. □

Before stating the central result of the paper, we highlight some appealing features of Distributed Primal Decomposition. First, agents do not need to communicate any of their private information, i.e., the local cost function f_i, the local constraint set X_i, the local solution estimate $\mathbf{x}_i^t$. Instead, only dual information is exchanged with neighbors. Moreover, the algorithm is scalable: as the network size grows, the amount of local computation stays constant. We highlight that the algorithm does not employ any slow averaging mechanism typically used in dual algorithms. Finally, the only synchronous feature of the algorithm is the step size. This prepares the ground for asynchronous extensions, see, e.g., [18].

Next, we state the convergence properties of Distributed Primal Decomposition, under the assumption of non-summable, square-summable step-size.

Assumption 3.3: The sequence $\{\alpha^t\}_{t \geq 0}$, with each $\alpha^t \geq 0$, satisfies $\sum_{t=0}^{\infty} \alpha^t = \infty$, $\sum_{t=0}^{\infty} (\alpha^t)^2 < \infty$. □

Theorem 3.4: Let Assumptions 2.1, 2.2, 3.1 and 3.3 hold. Moreover, let $\boldsymbol{\mu}^\star$ be a Lagrange multiplier of problem (1) associated to the constraint $\sum_{i=1}^N \mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{0}$ and assume $M > \|\boldsymbol{\mu}^\star\|_1$. Consider a sequence $\{\mathbf{x}_i^t, \rho_i^t\}_{t \geq 0}$, $i \in \{1, \dots, N\}$ generated by the Distributed Primal Decomposition algorithm with allocation vectors $\mathbf{y}_i^0$ initialized such that $\sum_{i=1}^N \mathbf{y}_i^0 = \mathbf{0}$. Then, almost surely, it holds

(i) $\sum_{i=1}^N \left(f_i(\mathbf{x}_i^t) + M\rho_i^t \right) \to f^\star$, where $f^\star$ is the optimal cost of (1);

(ii) every limit point of $\{(\mathbf{x}_1^t, \dots \mathbf{x}_N^t)\}_{t \geq 0}$ is an optimal (feasible) solution of (1). □

The proof of Theorem 3.4 is provided in Section IV. We highlight that almost sure asymptotic primal feasibility of the iterates (i.e., Theorem 3.4 *(ii)*) is obtained thanks to the primal decomposition approach. Viceversa, in dual decomposition schemes running averages are typically required to recover feasibility.

IV. Algorithm Analysis

In this section, we provide a sketch of analysis of Distributed Primal Decomposition. First, we introduce a reformulation of problem (4) that allows for the suppression of the constraint $\sum_{i=1}^N \mathbf{y}_i = \mathbf{0}$. Then, we reformulate the algorithm as a randomized block subgradient method. Due to space limitations, only the main points of the analysis are provided here, while the complete derivation is deferred to a forthcoming document.

In this section, we adopt the following notational conventions. The identity matrix is denoted by I, while $\otimes$ denotes the Kronecker product. As regards the notation for blocks, we proceed as follows. Given a vector $\mathbf{z}$, the ℓ-th block (or portion) of $\mathbf{z}$ is denoted as $\mathbf{z}_\ell$. We will also consider portions of a vector $\mathbf{z}$ that are associated to an edge of the graph. Given an edge of the graph, say (i, j), we denote the portion of $\mathbf{z}$ associated to (i, j) by $\mathbf{z}_{(ij)}$.

A. Encoding the Coupling Constraints in Cost Function

In this subsection, we introduce a method for suppressing the coupling constraint from problem (4), while still keeping it implicitly satisfied. This is obtained through an appropriate change of variable. Consider the underlying communication graph $\mathcal{G}_u$. Assuming an arbitrary ordering of the edges, let $\Gamma \in \mathbb{R}^{|\mathcal{E}_u| \times N}$ denote the incidence matrix of $\mathcal{G}_u$, where each row, (corresponding to an edge in the graph) contains all zero entries except the column corresponding to the edge tail (equal to 1), and the column corresponding to the edge head (equal to -1). For instance, if the k-th row of Γ corresponds to the edge (i, j), then the entry (k, ℓ) of Γ is

$$(\Gamma)_{k\ell} = \begin{cases} 1 & \text{if } \ell = i, \\ -1 & \text{if } \ell = j, \\ 0 & \text{otherwise,} \end{cases}$$

for all $k, \ell \in \{1, \dots, N\}$. In the following, two well-known properties of Γ will be used: *(i)* the vector of ones is an eigenvector of Γ with eigenvalue 0, i.e., $\Gamma \mathbf{1} = \mathbf{0}$, *(ii)* since $\mathcal{G}_u$ is connected, then $\text{rank}(\Gamma) = N - 1$.

Now, let us introduce the following change of variable

$$\mathbf{y} = \Pi \mathbf{z}, \qquad \mathbf{z} \in \mathbb{R}^{S|\mathcal{E}_u|}, \tag{9}$$

where $\mathbf{z}$ is the stack of vectors $\mathbf{z}_{(ij)} \in \mathbb{R}^S$ for all edges $(i,j) \in \mathcal{E}_u$, with the same ordering as in Γ, and the matrix Π is defined as

$$\Pi \triangleq (\Gamma^\top \otimes I) \in \mathbb{R}^{SN \times S|\mathcal{E}_u|}. \tag{10}$$

By using the properties of the Kronecker product, the components of $\mathbf{y}$ can be written as

$$\mathbf{y}_i = [\Pi \mathbf{z}]_i = \sum_{j \in \mathcal{N}_{i,u}} (\mathbf{z}_{(ij)} - \mathbf{z}_{(ji)}), \qquad \forall\, i \in \{1, \dots, N\},$$

where $[\Pi \mathbf{z}]_i$ denotes the i-th portion of $\Pi \mathbf{z}$ and $\mathcal{N}_{i,u}$ is the set of neighbors of node i in the underlying graph $\mathcal{G}_u$. It turns out that the change of variable (9) can be successfully used to encode the coupling constraint of problem (4) in the cost function, as a consequence of the following result.

Lemma 4.1: The matrix Π satisfies:

(i) $\sum_{i=1}^N [\Pi \mathbf{z}]_i = \mathbf{0}$ for all $\mathbf{z} \in \mathbb{R}^{S|\mathcal{E}_u|}$;

(ii) for all $\tilde{\mathbf{y}} \in \mathbb{R}^{SN}$ satisfying $\sum_{i=1}^N \tilde{\mathbf{y}}_i = \mathbf{0}$ there exists $\tilde{\mathbf{z}} \in \mathbb{R}^{S|\mathcal{E}_u|}$ such that $\tilde{\mathbf{y}} = \Pi \tilde{\mathbf{z}}$. □

We now plug the change of variable (9) into problem (4). Formally, for all $i \in \{1, \dots, N\}$, define the functions

$$\tilde{p}_i\big(\{\mathbf{z}_{(ij)}, \mathbf{z}_{(ji)}\}_{j \in \mathcal{N}_{i,u}}\big) \triangleq p_i\big([\Pi \mathbf{z}]_i\big), \qquad \mathbf{z} \in \mathbb{R}^{S|\mathcal{E}|}.$$

By Lemma 4.1, we obtain the following result.

Corollary 4.2: Problem (4) is equivalent to the unconstrained problem

$$\min_{\mathbf{z}} \sum_{i=1}^N \tilde{p}_i\big(\{\mathbf{z}_{(ij)}, \mathbf{z}_{(ji)}\}_{j \in \mathcal{N}_{i,u}}\big), \tag{11}$$

in the sense that: *(i)* the optimal costs are equal, and *(ii)* if $\mathbf{z}^\star$ is an optimal solution of (11), then $\mathbf{y}^\star = \Pi \mathbf{z}^\star$ is an optimal solution of (4). □

In the following, we denote the cost function of (11) as $\tilde{p}(\mathbf{z}) = \sum_{i=1}^N \tilde{p}\big(\{\mathbf{z}_{(ij)}, \mathbf{z}_{(ji)}\}_{j \in \mathcal{N}_{i,u}}\big) = p(\Pi \mathbf{z})$.

B. Reformulation of Distributed Primal Decomposition as Randomized Block Subgradient Method

In this section, we recast the Distributed Primal Decomposition algorithm as an equivalent, randomized block subgradient method for solving problem (11).

First, let us define the block structure for the optimization variable $\mathbf{z}$ of problem (11). We associate each block of $\mathbf{z}$ to an undirected edge of $\mathcal{G}_u$. Thus, a total of $B = |\mathcal{E}_u|/2$ blocks have to be considered. Formally, assume an arbitrary ordering of the undirected edges. If a block $\ell \in \{1, \dots, B\}$ is associated to the undirected edge $(i,j) \in \mathcal{E}_u$, with $j > i$, then the ℓ-th block of $\mathbf{z}$, denoted by $\mathbf{z}_\ell \in \mathbb{R}^{2S}$, consists of the stack of $\mathbf{z}_{(ij)}, \mathbf{z}_{(ji)} \in \mathbb{R}^S$, i.e.,

$$\mathbf{z}_\ell = \begin{bmatrix} \mathbf{z}_{(ij)} \\ \mathbf{z}_{(ji)} \end{bmatrix}.$$

By analogy, we will use the shorthand $\sigma_\ell = \sigma_{ij}$ to denote the probability associated to the ℓ-th block. Clearly, by Assumption 3.1, it holds $\sigma_\ell = 1/B$, i.e., blocks have uniform probability to be picked. Without loss of generality, we will suppose that the blocks are ordered such that $\mathbf{z}^\top = [\mathbf{z}_1^\top \ \cdots \ \mathbf{z}_B^\top]$.

Next, we formulate a randomized block subgradient method applied to problem (11). In view of the communication structure (cf. Section III-A), at each iteration t, a number of $b^t \leq B$ blocks activate, corresponding to the undirected edges in the graph that are active at that iteration. As per Assumption 3.1, the probability that exactly b blocks activate is ν_b, for all $b \in \{1, \dots, B\}$. We denote by $B^t \subseteq \{1, \dots, B\}$ the index set of the blocks activated at time t. The algorithm is basically a subgradient method, where at each iteration t we only update the blocks in B^t, i.e.,

$$\mathbf{z}_\ell^{t+1} = \begin{cases} \mathbf{z}_\ell^t - \alpha^t [\widetilde{\nabla} \tilde{p}(\mathbf{z}^t)]_\ell, & \text{if } \ell \in B^t, \\ \mathbf{z}_\ell^t, & \text{if } \ell \notin B^t, \end{cases} \tag{12}$$

where $\widetilde{\nabla} \tilde{p}(\mathbf{z}^t) \in \mathbb{R}^{S|\mathcal{E}_u|}$ denotes a subgradient of $\tilde{p}$ at $\mathbf{z}^t$, $[\widetilde{\nabla} \tilde{p}(\mathbf{z}^t)]_\ell$ denotes the ℓ-th block of $\widetilde{\nabla} \tilde{p}(\mathbf{z}^t)$ and α^t is the stepsize. Notice that, differently from standard block subgradient methods, algorithm (12) updates multiple blocks at once.

In order to show that algorithm (12) is equivalent to Distributed Primal Decomposition, let us elaborate on (12) from a computational point of view. For this purpose, let us discuss on the computation of a subgradient of $\tilde{p}$ at any $\mathbf{z} \in \mathbb{R}^{S|\mathcal{E}_u|}$. By definition, it holds $\tilde{p}(\mathbf{z}) = p(\Pi \mathbf{z})$. Thus, by using the subgradient property for affine transformations of the domain, we conclude that

$$\begin{aligned} \widetilde{\nabla} \tilde{p}(\mathbf{z}) &= \Pi^\top \widetilde{\nabla} p(\Pi \mathbf{z}) \\ &= (\Gamma \otimes I) \widetilde{\nabla} p(\Pi \mathbf{z}). \end{aligned} \tag{13}$$

Recall that $\mathbf{y} = \Pi \mathbf{z}$. On the one hand, by exploiting the separability of p, the i-th component of $\widetilde{\nabla} p(\mathbf{y})$ is equal to $\frac{\tilde{\partial} p(\mathbf{y})}{\tilde{\partial} \mathbf{y}_i} = \widetilde{\nabla} p_i(\mathbf{y}_i)$. On the other hand, as problem (3) enjoys strong duality, according to [17, Section 5.4.4], a subgradient of p_i at $\mathbf{y}_i$ can be computed as $\widetilde{\nabla} p_i(\mathbf{y}_i) = -\boldsymbol{\mu}_i$, where $\boldsymbol{\mu}_i$ is a Lagrange multiplier of problem (3) associated to the constraint $\mathbf{g}_i(\mathbf{x}_i) \leq \mathbf{y}_i + \rho_i \mathbf{1}$. By collecting this facts together with (13), we see that the components of $\widetilde{\nabla} \tilde{p}(\mathbf{z})$ can be computed as

$$\begin{aligned} \frac{\tilde{\partial} \tilde{p}(\mathbf{z})}{\partial \mathbf{z}_{(ij)}} &= \widetilde{\nabla} p_i\Big([\Pi \mathbf{z}]_i\Big) - \widetilde{\nabla} p_j\Big([\Pi \mathbf{z}]_j\Big) \\ &= \boldsymbol{\mu}_j - \boldsymbol{\mu}_i, \qquad \forall\, (i,j) \in \mathcal{E}_u, \end{aligned} \tag{14}$$

where $\frac{\tilde{\partial} \tilde{p}(\mathbf{Z})}{\partial \mathbf{z}_{(ij)}}$ denotes the component of $\widetilde{\nabla} \tilde{p}(\mathbf{z})$ associated to $\mathbf{z}_{(ij)}$ and $\boldsymbol{\mu}_k$ denotes a Lagrange multiplier for the problem

$$\begin{aligned} \min_{\mathbf{x}_k, \rho_k} \ & f_k(\mathbf{x}_k) + M \rho_k \\ \text{subj. to } & \mathbf{g}_k(\mathbf{x}_k) \leq [\Pi \mathbf{z}]_k + \rho_k \mathbf{1} \\ & \rho_k \geq 0, \ \mathbf{x}_k \in X_k, \end{aligned} \tag{15}$$

associated to the constraint $\mathbf{g}_k(\mathbf{x}_k) \leq [\Pi \mathbf{z}]_k + \rho_k \mathbf{1}$, for all $k \in \{1, \dots, N\}$. By using (14), the update (12) can be recast as

$$\mathbf{z}_{(ij)}^{t+1} = \begin{cases} \mathbf{z}_{(ij)}^t + \alpha^t \big(\boldsymbol{\mu}_i^t - \boldsymbol{\mu}_j^t\big), & \text{if } (i,j) \in \mathcal{E}^t, \\ \mathbf{z}_{(ij)}^t, & \text{if } (i,j) \notin \mathcal{E}^t, \end{cases} \tag{16}$$

where $\boldsymbol{\mu}_k^t$ is a Lagrange multiplier for problem (15), with $\mathbf{z} = \mathbf{z}^t$. If we now let $\mathbf{y}^t = \Pi \mathbf{z}^t$, we see that algorithm (16) is equivalent to (7)–(8).

V. Numerical example

In this section, we provide numerical computations to validate the theoretical results. We apply the Distributed Primal Decomposition algorithm to a microgrid control problem, in the form (1). We do not provide the problem formulation, which can be found in [7]. The same parameters considered in [7] are used, i.e., $N = 10$ agents, consisting of 4 generators, 3 storage devices, 2 controllable loads and 1 connection to the main grid. The prediction horizon of the optimal control problem is equal to $T = 8$ slots, so that $S = 8$ coupling constraints are considered. First, we generate a random underlying Erdős-Rényi graph with edge probability 0.2. The probabilities ν_b (cf. Assumption 3.1) are generated randomly and the local allocations are initialized to $\mathbf{y}_1^0 = \ldots = \mathbf{y}_N^0 = \mathbf{0}$. We perform a Montecarlo simulation with a total of 50 graph realizations, and we run the Distributed Primal Decomposition algorithm over each realization. In Figure 1, the evolution of the algorithm is shown. On the left, the scaled cost error $(\sum_{i=1}^N (f_i(\mathbf{x}_i^t) + M\rho_i^t) - f^\star)/f^\star$ on a logarithmic scale is provided, where $(\mathbf{x}_i^t, \rho_i^t)$ are the primal quantities generated by the algorithm and $f^\star$ is the optimal cost of the problem, computed by a centralized solver. On the right, a plot of the maximum component of the coupling constraints, i.e., $\max_{s\in\{1,\ldots,S\}} \sum_{i=1}^N \mathbf{g}_{i,s}(\mathbf{x}_i^t)$, is shown. It can

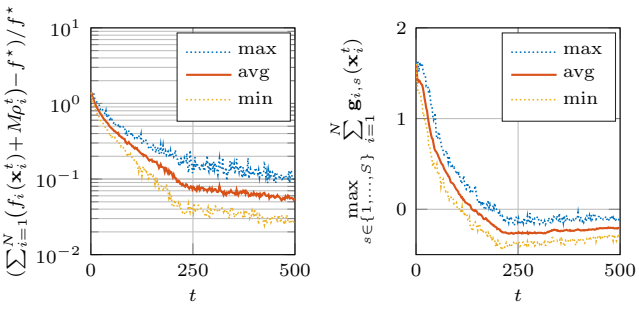

Fig. 1. Evolution of the cost error (left) and of the maximum component of the coupling constraints (right). The maximum, minimum and average are computed with respect to 50 graph realizations.

be seen from the figures that the objective value converges to the optimal cost of the original problem, consistently with the theoretical result of Theorem 3.4 *(i)*. Moreover, the quantities computed by the algorithm are asymptotically feasible for the coupling constraints, confirming the primal feasibility property *(ii)* of Theorem 3.4.

VI. Conclusions

In this paper, we proposed a distributed optimization algorithm, named Distributed Primal Decomposition, for a network of agents in a random communication network. The algorithm, based on a relaxation and primal decomposition approach, allows the agents to minimize the sum of local cost functions, subject to local constraints and global coupling constraints. To analyze the algorithm, we use an appropriate change of variables, so that distributed update can be reformulated as a randomized block subgradient method applied to a suitable equivalent problem. The communication structure is mapped to blocks that randomly activate at each iteration. The results of this paper pave the way for more comprehensive frameworks, which can run over general communication networks. Numerical computations on microgrid control substantiated the theoretical results.

References

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

[2] D. Mateos-Núnez and J. Cortés, "Distributed saddle-point subgradient algorithms with laplacian averaging," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2720–2735, 2017.

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

[4] 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.

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

[6] 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.

[7] I. Notarnicola and G. Notarstefano, "Constraint-coupled distributed optimization: a relaxation and duality approach," *IEEE Transactions on Control of Network Systems*, vol. PP, no. 99, pp. 1–10, 2019.

[8] A. Beck and L. Tetruashvili, "On the convergence of block coordinate descent type methods," *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.

[9] M. Razaviyayn, M. Hong, and Z.-Q. Luo, "A unified convergence analysis of block successive minimization methods for nonsmooth optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1126–1153, 2013.

[10] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, vol. 144, no. 1-2, pp. 1–38, 2014.

[11] C. D. Dang and G. Lan, "Stochastic block mirror descent methods for nonsmooth and stochastic optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 856–881, 2015.

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

[13] A. Nedić, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[14] A. Nedić and J. Liu, "Distributed optimization for control," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 77–103, 2018.

[15] A. Camisa, I. Notarnicola, and G. Notarstefano, "A primal decomposition method with suboptimality bounds for distributed mixed-integer linear programming," in *IEEE Conference on Decision and Control*, 2018, pp. 3391–3396.

[16] G. J. Silverman, "Primal decomposition of mathematical programs by resource allocation: I – basic theory and a direction-finding procedure," *Operations Research*, vol. 20, no. 1, pp. 58–74, 1972.

[17] D. P. Bertsekas, *Nonlinear programming*. Athena Scientific, 1999.

[18] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Asynchronous gossip algorithms for stochastic optimization," in *IEEE Conference on Decision and Control*, 2009, pp. 3581–3586.