Primal decomposition and constraint generation for asynchronous distributed mixed-integer linear programming

Primal Decomposition and Constraint Generation for Asynchronous Distributed Mixed-Integer Linear Programming

Andrea Camisa, Giuseppe Notarstefano

Abstract— **In this paper, we deal with large-scale Mixed Integer Linear Programs (MILPs) with coupling constraints that must be solved by processors over networks. We propose a finite-time distributed algorithm that computes a feasible solution with suboptimality bounds over asynchronous and unreliable networks. As shown in a previous work of ours, a feasible solution of the considered MILP can be computed by resorting to a primal decomposition of a suitable problem convexification. In this paper we reformulate the primal decomposition resource allocation problem as a linear program with an exponential number of unknown constraints. Then we design a distributed protocol that allows agents to compute an optimal allocation by generating and exchanging only few of the unknown constraints. Each allocation is iteratively used to compute a candidate feasible solution of the original MILP. We establish finite-time convergence of the proposed algorithm under very general assumptions on the communication network. A numerical example corroborates the theoretical results.**

I. Introduction

In different network contexts such as, e.g., energy management, smart grid, cooperative task allocation in robotic networks, Mixed-Integer Linear Programs (MILPs) need to be solved in real time. In these applications, problem data is typically scattered throughout the network, and collecting the required information at a central node is not possible (or at least convenient). Thus, distributed optimization algorithms are required. Moreover, as MILPs are NP-hard problems, fast algorithms providing (feasible) suboptimal solutions are attractive, as e.g., in the case of dynamic optimization problems arising in Model Predictive Control [1].

In [2], a fast parallel dual decomposition approach based on the restriction of the coupling constraints has been introduced to obtain a feasible solution, however a coordinating unit is required. The method has been improved in [3], where a time-varying restriction allows for a higher solution quality. To date, there are few works tackling MILPs within a distributed framework. In [4], a distributed version of [3] has been formalized. A different idea has been explored in [5], where a *primal* decomposition approach is exploited to devise a fast distributed algorithm that computes a (feasible) suboptimal solution, where simulations on random problems have highlighted tighter suboptimality bounds compared to [2]. Other approaches to solve MILPs over networks include [6], where a distributed cutting-plane algorithm is devised to solve a MILP with common cost. In [7], a distributed column generation approach is used to solve a task-target assignment problem. In [8], a multi-assignment problem is solved by means of an auction-based algorithm. All the methods mentioned so far involve the iterative solution of suitably constructed Linear Programs (LPs). As regards distributed algorithms for LPs, in [9] a simplex algorithm is formulated to solve degenerate LPs. In [10] a continuous-time algorithm to obtain an optimal primal-dual pair of a LP is considered, while in [11] a method based on event-triggered communication is analyzed with tools from switched and hybrid systems. In [12], a distributed Dantzig-Wolfe decomposition method with online column generation is used to solve in finite time a LP with a coupling constraint among the variables. Notice that [9]–[12] are not immediately applicable to the LP convexification of the MILP addressed in this paper, since an explicit description of the local convexified constraints is not available, as we will show in the following.

The contributions of the paper are as follows. We propose a distributed algorithm that converges, in a finite number of communication rounds, to a feasible solution, with suboptimality bounds, of a large-scale MILP with coupling constraints among the variables. Following [5], we adopt a primal decomposition approach applied to a convexification of the target MILP with restricted coupling constraints. In this paper, we exploit piece-wise linearity of the cost functions to suitably reformulate the primal decomposition (resource allocation) problem into a LP with exponentially many constraints. Then, to solve the LP, we propose a distributed algorithm with constraint exchange among neighbors (whose general idea was introduced in [13]), based on the local online generation of constraints at each node, so that the simultaneous processing of all the constraints is avoided. The optimal allocation is then used to compute a (feasible) suboptimal solution to the original MILP. We are able to establish finite-time convergence of the algorithm under very general assumptions on the communication network, i.e., asynchronous, unreliable (e.g., subject to packet loss), directed time-varying graph. Our algorithm requires, as a computation step, the calculation of the lexicographically minimal Lagrange multiplier of a small local linear program. However, since this problem is not trivial to solve in practice, because of the (unknown) exponentially many constraints, we provide a local routine, based on outer approximations, to perform this computation step. Notably, our algorithm preserves information privacy, i.e., the solution estimates are not shared with neighbors. Finally, we provide numerical computations, confirming the theoretical analysis.

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 and G. Notarstefano are with the Department of Electrical, Electronic and Information Engineering, University of Bologna, Bologna, Italy, a.camisa@unibo.it and giuseppe.notarstefano@unibo.it.

This paper extends [5]. Indeed, here we achieve finite-time convergence by exploiting piece-wise linearity of the costs of the primal decomposition problem and by using a different algorithmic structure. On the other hand, in [5], only finite-time feasibility can be obtained, but it requires further restriction of the coupling constraints, with a degradation of the solution performance. The idea of combining local online constraint generation and exchange is inspired by the distributed column generation approach in [12]. However, the algorithm of the current paper is structurally different, since it is based on primal decomposition instead of Dantzig-Wolfe decomposition. Moreover, the algorithm in [12] addresses LPs and its extension to MILPs is not straightforward.

The paper is organized as follows. In Section II we describe the problem set-up and we recall several results on primal decomposition. In Section III we reformulate the problem and we introduce our distributed algorithm together with a local routine. The algorithm is analyzed in Section IV, while in Section V we provide numerical computations. For the sake of space, all proofs are omitted and will be provided in a forthcoming document.

II. Problem Set-up and Primal Decomposition

We consider a network of N agents that communicate according to a time-varying digraph $\mathcal{G}_c^t = (\{1,\ldots,N\}, \mathcal{E}^t)$, where t denotes time and $\mathcal{E}^t \subseteq \{1,\ldots,N\}^2$ is the edge set at time t. We will denote by $\mathcal{N}_i^t$ the in-neighbor set of agent i at time t, i.e., $\mathcal{N}_i^t = \{j \mid (j,i) \in \mathcal{E}^t\}$. We make the following assumption on the communicaton graph.

Assumption 2.1: The communication graph $\mathcal{G}_c^t$ is jointly strongly connected, i.e., the graph $\mathcal{G}_\infty^t \triangleq (\{1,\ldots,N\}, \mathcal{E}_\infty^t)$, with $\mathcal{E}_\infty^t = \bigcup_{\tau=t}^{\infty} \mathcal{E}^\tau$, is strongly connected for all $t \geq 0$. □

Assumption 2.1 is very general. We will see that in the local iterates of our algorithm the agents need not know the universal time t, which means that it can also be implemented in unreliable, asynchronous communication networks, and, in particular, in networks subject to packet loss.

We suppose that the agents want to solve the mixed-integer linear program

$$\begin{aligned} \min_{\mathbf{x}_1,\ldots,\mathbf{x}_N} \quad & \sum_{i=1}^N c_i^\top \mathbf{x}_i \\ \text{subj. to} \quad & \sum_{i=1}^N A_i \mathbf{x}_i \leq b \\ & \mathbf{x}_i \in P_i \cap (\mathbb{Z}^{Z_i} \times \mathbb{R}^{R_i}), \qquad i \in \{1,\ldots,N\}, \end{aligned} \tag{1}$$

where $P_i \subset \mathbb{R}^{n_i}$ are nonempty compact polyhedra with $n_i = Z_i + R_i$, and $A_i \in \mathbb{R}^{S \times n_i}$ and $b \in \mathbb{R}^S$ describe coupling constraints among the variables. Throughout the paper, we use inequality symbols also for vectors, meaning that inequality holds for each component of the vectors. To streamline the notation, we denote the local mixed-integer sets as $X_i \triangleq P_i \cap (\mathbb{Z}^{Z_i} \times \mathbb{R}^{R_i})$. We assume that problem (1) is feasible and that each agent i knows only c_i, A_i, X_i and that everyone knows b.

For large-scale problems, where $\sum_{i=1}^N n_i \gg S$, it is of interest to look for fast distributed algorithms providing (feasible) suboptimal solutions of problem (1), with guaranteed suboptimality bounds. Indeed, being (1) NP-hard, computing an optimal solution may be computationally prohibitive. To this end, we employ a method based on the convexification of the local sets X_i and on the restriction of the coupling constraints [2]. In this paper, however, we will not rely on the *dual* decomposition approach used in [2], but instead consider the *primal* decomposition idea introduced in our previous work [5], for which tighter restrictions and lower suboptimality levels can be obtained (see [5, Section V.B]). Formally, the convexified and restricted problem, for a given a restriction $\boldsymbol{\sigma} \geq \mathbf{0}$, is

$$\begin{aligned} \min_{\mathbf{z}_1,\ldots,\mathbf{z}_N} \quad & \sum_{i=1}^N c_i^\top \mathbf{z}_i \\ \text{subj. to} \quad & \sum_{i=1}^N A_i \mathbf{z}_i \leq b - \boldsymbol{\sigma} \\ & \mathbf{z}_i \in \text{conv}(X_i), \qquad i \in \{1,\ldots,N\}, \end{aligned} \tag{2}$$

where $\text{conv}(X_i)$ denotes the convex hull of X_i and we adopted the notational convention that the variables $\mathbf{z}_i \in \mathbb{R}^{n_i}$ are the convex counterpart of $\mathbf{x}_i \in \mathbb{Z}^{Z_i} \times \mathbb{R}^{R_i}$. As done in other works, we make the following assumption.

Assumption 2.2: For a given $\boldsymbol{\sigma} \geq \mathbf{0}$, problem (2) is feasible and its optimal solution is unique. □

A slight perturbation of the cost vector is sufficient to guarantee uniqueness of the solution, while a sufficiently tight restriction can likely allow for the feasibility of the restricted problem [5]. Also, notice that we are not requiring feasibility of the restricted version of (1). Problem (2) is convex and exhibits a decomposable structure, as we show in the next subsection.

A. Preliminaries on Relaxation and Primal Decomposition

Primal decomposition can be directly applied to problem (2). However, as it will be clear from the forthcoming discussion, this approach would lead to an optimization problem with constraints that are difficult to handle. To overcome this issue, we consider the following *relaxed* version of problem (2)

$$\begin{aligned} \min_{\substack{\mathbf{z}_1,\ldots,\mathbf{z}_N,\\ v_1,\ldots,v_N}} \quad & \sum_{i=1}^N (c_i^\top \mathbf{z}_i + R v_i) \\ \text{subj. to} \quad & \sum_{i=1}^N A_i \mathbf{z}_i \leq b - \boldsymbol{\sigma} + \sum_{i=1}^N v_i \mathbf{1} \\ & \mathbf{z}_i \in \text{conv}(X_i), \ v_i \geq 0, \qquad i \in \{1,\ldots,N\}, \end{aligned} \tag{3}$$

where $\mathbf{1}$ is the vector with all components equal to 1 and we added the v_i variables, used to model *violations* of the coupling constraints. These violations, however, are penalized in the cost by an appropriate parameter $R > 0$. The following lemma establishes the relationship between the optimal solutions of problems (2) and (3).

Lemma 2.3: There exists a sufficiently large $R > 0$ such that the optimal solutions of problem (3) are of the form $(\mathbf{z}_1^\star,\ldots,\mathbf{z}_N^\star,0,\ldots,0)$, with $(\mathbf{z}_1^\star,\ldots,\mathbf{z}_N^\star)$ being an optimal solution of problem (2). Thus, the optimal solutions of (3) must have $v_i = 0$ for all $i \in \{1,\ldots,N\}$. □

The proof of the statement is very similar to [14, Proposition III.3]. We now wish to solve (3) by exploiting the decomposable structure of the problem. In a primal decomposition approach, it is customary to consider, for all $i \in \{1, \ldots, N\}$, the function $p_i : \mathbb{R}^S \to \mathbb{R}$, defined as the optimal cost of the i-th subproblem

$$\begin{aligned} p_i(\mathbf{y}_i) = \min_{\mathbf{z}_i, v_i} \; & c_i^\top \mathbf{z}_i + R v_i \\ \text{subj. to } \; & A_i \mathbf{z}_i \leq \mathbf{y}_i + v_i \mathbf{1} \\ & \mathbf{z}_i \in \text{conv}(X_i), \; v_i \geq 0, \end{aligned} \tag{4}$$

where $\mathbf{y}_i \in \mathbb{R}^S$ is the i-th allocation vector, and the goal is to find an optimal allocation by solving the *master problem*

$$\begin{aligned} \min_{\mathbf{y}_1, \ldots, \mathbf{y}_N} \; & \sum_{i=1}^N p_i(\mathbf{y}_i) \\ \text{subj. to } \; & \sum_{i=1}^N \mathbf{y}_i = b - \boldsymbol{\sigma}. \end{aligned} \tag{5}$$

Let $(\mathbf{y}_1^\star, \ldots, \mathbf{y}_N^\star)$ be an optimal solution of problem (5) and let $(\mathbf{z}_i^\star, v_i^\star)$ be an optimal solution of problem (4) with $\mathbf{y}_i = \mathbf{y}_i^\star$. Then, by using Lemma 2.3, it follows that $v_i^\star = 0$ for all i and that $(\mathbf{z}_1^\star, \ldots, \mathbf{z}_N^\star)$ is an optimal solution of problem (2).

B. Feasible Mixed-Integer Solution Computation

The primal decomposition approach discussed in Section II-A can be used to compute a (feasible) suboptimal solution of problem (1). In [5], we introduced a method based on the restriction of the coupling constraints that allows for the computation of a feasible mixed-integer solution. Formally, we first define the vector $\mathbf{L}_i \in \mathbb{R}^S$ with components

$$\mathbf{L}_i^s = \min_{\mathbf{x}_i \in X_i} A_i^s \mathbf{x}_i, \qquad s \in \{1, \ldots, S\},$$

where A_i^s denotes the s-th row of A_i and the $\mathbf{L}_i$ vectors represent the minimum amount of allocation that an agent requires for a feasible solution. Then, a tight a-priori restriction can be computed as

$$\boldsymbol{\sigma} \triangleq \mathbf{1}(S+1) \max_{i \in \{1, \ldots, N\}} \max_{s \in \{1, \ldots, S\}} \Big(A_i^s \mathbf{x}_i^{\text{L}} - \mathbf{L}_i^s \Big), \tag{6}$$

where, for all i, $\mathbf{x}_i^{\text{L}}$ is an optimal solution of

$$\min_{\mathbf{x}_i \in X_i} \max_{s \in \{1, \ldots, S\}} (A_i^s \mathbf{x}_i - \mathbf{L}_i^s).$$

Given an optimal solution of the master problem (5), with $\boldsymbol{\sigma}$ defined as in (6), a feasible mixed-integer solution of problem (1) can be computed as specified by the next proposition, where we denote by lex-min the lexicographically minimal optimal solution of an optimization problem.

Proposition 2.4 ([5], Theorem 4.1): Let Assumption 2.2 hold and let $(\mathbf{y}_1^\star, \ldots, \mathbf{y}_N^\star)$ be an optimal solution of (5), with $\boldsymbol{\sigma}$ equal to (6), and $R > 0$ sufficiently large. Moreover, for all i, let $(\varphi_i^\star, \xi_i^\star, \mathbf{x}_i^\star)$ be the optimal solution of

$$\begin{aligned} \operatorname*{lex-min}_{\varphi_i, \xi_i, \mathbf{x}_i} \; & \varphi_i \\ \text{subj. to } \; & c_i^\top \mathbf{x}_i \leq \xi_i \\ & A_i \mathbf{x}_i \leq \mathbf{y}_i^\star + \varphi_i \mathbf{1} \\ & \mathbf{x}_i \in X_i, \; \varphi_i \geq 0. \end{aligned}$$

Then, $(\mathbf{x}_1^\star, \ldots, \mathbf{x}_N^\star)$ is a feasible solution of (1). □

It is worth noting that in Proposition 2.4 is the first place where Assumption 2.2 comes into play, and is required to guarantee feasibility of the computed solution for MILP (1).

Remark 2.5 (Suboptimality bounds): The feasible mixed-integer solution defined in Proposition 2.4 is, in general, suboptimal. However, tight suboptimality bounds can be computed as suggested by [5, Theorem 4.3]. □

III. A DISTRIBUTED FINITE-TIME ALGORITHM

Proposition 2.4 allows us to focus on the solution of problem (5) as a key step for obtaining a feasible solution of (1). For time-varying networks and for smooth p_i functions, one can apply distributed algorithms such as the one considered in [15], which enjoys asymptotic convergence. However, the p_i are not smooth in general. In this section we introduce a distributed algorithm with finite-time convergence that solves problem (5) and computes a (possibly suboptimal) feasible solution of (1). Before describing the algorithm, we discuss a linear program reformulation of problem (5).

A. LP Reformulation of Master Problem

It is well known that, since X_i is bounded, conv(X_i) is a bounded polyhedron [16]. This implies that each subproblem (4) is in fact a Linear Program. This property allows us to prove the following important result.

Lemma 3.1: For all $i \in \{1, \ldots, N\}$, the function p_i is piece-wise linear, i.e., there exist $a_i^1, \ldots, a_i^{|L_i|} \in \mathbb{R}^S$ and $f_i^1, \ldots, f_i^{|L_i|} \in \mathbb{R}$ such that

$$p_i(\mathbf{y}_i) = \max_{\ell \in L_i} \big(\mathbf{y}_i^\top a_i^\ell + f_i^\ell \big). \qquad \square$$

A consequence of Lemma 3.1 is that $p_i(\mathbf{y}_i)$ can be alternatively defined as the smallest number ρ_i such that $(a_i^\ell)^\top \mathbf{y}_i + f_i^\ell \leq \rho_i$ for all $\ell \in L_i$. Thus, problem (5) can be equivalently recast as the linear program

$$\begin{aligned} \min_{\substack{\mathbf{y}_1, \ldots, \mathbf{y}_N \\ \rho_1, \ldots, \rho_N}} \; & \sum_{i=1}^N \rho_i \\ \text{subj. to } \; & \sum_{i=1}^N \mathbf{y}_i = b - \boldsymbol{\sigma} \\ & \mathbf{y}_i^\top a_i^\ell + f_i^\ell \leq \rho_i, \quad \forall \ell \in L_i, \, \forall i \in \{1, \ldots, N\}. \end{aligned} \tag{7}$$

In the following, we will compactly denote the optimization variable as $(\mathbf{y}, \boldsymbol{\rho})$. We point out that the number of inequality constraints of (7) is $\sum_{i=1}^N |L_i|$, which can be extremely large, since $|L_i|$ is usually exponential in the number of integer variables of X_i. Thus, it is not affordable to solve the problem by enumerating all the constraints. In the next subsection we introduce an algorithm, based on *online* constraint generation, to solve (7). Such algorithm is specifically designed to solve MILP (1), since the resulting LP has a huge number of constraints, not necessarily known a priori. To guarantee convergence of all the nodes to the same solution, we focus on the lex-optimal solution of problem (7).

Given a vertex $(\mathbf{y}, \boldsymbol{\rho})$ of the feasible set of problem (7), it is well known from linear programming that a *basis* associated to $(\mathbf{y}, \boldsymbol{\rho})$ consists of $N(S+1)$ active constraints, i.e., satisfied with equality at $(\mathbf{y}, \boldsymbol{\rho})$, such that the relaxed LP, containing only the constraints in the basis, has the same optimal solution of (7).

B. Distributed Algorithm: Description and Discussion

In this subsection, we introduce our algorithm *Distributed Primal Decomposition with Constraint Generation for MILP* (DiP-COGEN-MILP). We first introduce the notation. We use the abbreviation "lex-min multiplier" to denote the lexicographically minimal Lagrange multiplier. The subscript $[j]$ indicates the agent that computed a given quantity, while the subscript i is used to indicate the i-th component of a vector (when both are necessary, we write $i, [j]$). We denote the basis of agent i at time t with $B^t_{[i]}$ and we say that the tuple (a, f, j), with $a \in \mathbb{R}^S$, $f \in \mathbb{R}$ and $j \in \{1, \dots, N\}$, belongs to $B^t_{[i]}$ if the constraint $\mathbf{y}_j^\top a + f \leq \rho_i$ is in the basis of agent i at time t. If $B^t_{[i]}$ is a basis, then $(\mathbf{y}^t_{[i]}, \boldsymbol{\rho}^t_{[i]})$ must be the lex-optimal solution of problem (9) with $H_{\text{TMP}} = B^t_{[i]}$.

At each communication round t, agent i has a current guess $\mathbf{y}^t_{[i]}$ of the global allocation vector, that may violate some of the constraints of problem (7). Thus, a new constraint is generated as follows. It computes a Lagrange multiplier $\boldsymbol{\mu}^t_{[i]}$ of problem (8). According to [17, Section 5.4.4], $-\boldsymbol{\mu}^t_{[i]}$ is a subgradient of p_i at $\mathbf{y}^t_{i,[i]}$, and a feasible vector $(\mathbf{y}, \boldsymbol{\rho})$ of problem (7) must satisfy the subgradient inequality, i.e., it must hold $(\mathbf{y}^t_{i,[i]} - \mathbf{y}_j)^\top \boldsymbol{\mu}^t_{[i]} + p^t_{[i]} \leq \rho_j$. Thus, agent i generates this constraint and solves a local version of problem (7) to find a new guess of the global allocation, and a basis of the solution is communicated to neighbors. Finally, the updated allocation is used to compute a tentative mixed-integer solution for problem (1).

Our algorithm DiP-COGEN-MILP is summarized in the table from the perspective of node i, while the convergence properties are formalized in Section IV. We want to stress that problem (8) is a small linear program, but it is not possible to use standard techniques to solve it. Indeed, $\text{conv}(X_i)$ has an exponential number of constraints, not explicitly known. In Section III-C, we provide a routine that can be used to obtain the lex-min multiplier (needed to guarantee finite-time convergence) without enumerating all the constraints. As for problem (9), any lexicographic solver for LPs can be used. Finally, as regards problem (10), it can be solved as described in [5], i.e., by first determining the smallest φ_i such that the constraint $A_i \mathbf{x}_i \leq \mathbf{y}^{t+1}_{i,[i]} + \varphi_i \mathbf{1}$ is satisfied, then by minimizing $c_i^\top \mathbf{x}_i$. In order to cope with unbounded problems at an early stage of the algorithm execution, we consider the additional bounding box $-M\mathbf{1} \leq \mathbf{y}, \boldsymbol{\rho} \leq M\mathbf{1}$ in problem (9). The "big-M" initialization indicated in the table consists of computing $\mathbf{y}^0_{[i]}$ by solving problem (9) with $H_{\text{TMP}} = \emptyset$.

Remark 3.2 (Basis dimension): Since the S equality constraints of problem (7) are always satisfied at any feasible vector, the size of communicated bases consists of $N(S+1) - S$ active inequality constraints (except the bounding box constraints that are common to all the agents).

Distributed Algorithm DiP-COGEN-MILP

Initialization: $\mathbf{y}^0_{i,[i]}$ obtained via big-M and $B^0_{[i]} = \emptyset$

Evolution: for all $t = 0, 1, \dots$

Generate $h^t_{[i]} = \big((-\boldsymbol{\mu}^t_{[i]}),\ (p^t_{[i]} + \boldsymbol{\mu}^{t\top}_{[i]} \mathbf{y}^t_{i,[i]}),\ i\big)$ as a constraint tuple with $p^t_{[i]}$ the optimal cost and $\boldsymbol{\mu}^t_{[i]}$ the lex-min multiplier of

$$\begin{aligned} \min_{\mathbf{z}_i, v_i} \quad & c_i^\top \mathbf{z}_i + R v_i \\ \text{subj. to} \quad & \boldsymbol{\mu}_{[i]} : A_i \mathbf{z}_i \leq \mathbf{y}^t_{i,[i]} + v_i \mathbf{1} \\ & \mathbf{z}_i \in \text{conv}(X_i), \quad v_i \geq 0 \end{aligned} \tag{8}$$

Receive $B^t_{[j]}$ from $j \in \mathcal{N}^t_i$ and set

$$H_{\text{TMP}} = B^t_{[i]} \cup \big(\cup_{j \in \mathcal{N}^t_i} B^t_{[j]} \big) \cup \big\{ h^t_{[i]} \big\}$$

Compute $(\mathbf{y}^{t+1}_{[i]}, \boldsymbol{\rho}^{t+1}_{[i]})$ as the optimal solution and $B^{t+1}_{[i]}$ as the corresponding basis of

$$\begin{aligned} \operatorname*{lex-min}_{\substack{\mathbf{y}_1, \dots, \mathbf{y}_N \\ \rho_1, \dots, \rho_N}} \quad & \sum_{i=1}^N \rho_i \\ \text{subj. to} \quad & \sum_{i=1}^N \mathbf{y}_i = b - \boldsymbol{\sigma} \\ & \mathbf{y}_j^\top a + f \leq \rho_j, \qquad \forall\, (a, f, j) \in H_{\text{TMP}} \\ & -M\mathbf{1} \leq \mathbf{y}, \boldsymbol{\rho} \leq M\mathbf{1} \end{aligned} \tag{9}$$

Compute $\mathbf{x}^{t+1}_{[i]}$ as the optimal solution of

$$\begin{aligned} \operatorname*{lex-min}_{\varphi_i, \xi_i, \mathbf{x}_i} \quad & \varphi_i \\ \text{subj. to} \quad & c_i^\top \mathbf{x}_i \leq \xi_i \\ & A_i \mathbf{x}_i \leq \mathbf{y}^{t+1}_{i,[i]} + \varphi_i \mathbf{1} \\ & \mathbf{x}_i \in X_i, \ \ \varphi_i \geq 0. \end{aligned} \tag{10}$$

Comparison with previous work. In [5], a distributed algorithm has been proposed to obtain an optimal solution of problem (5), used to retrieve a feasible mixed-integer solution of the original MILP (1). The algorithm in [5] is based on a subgradient method with diminishing step size. In this paper, instead, we use a constraint generation and exchange technique, although the size of the optimization variable in problem (9) depends on N. We highlight that the constraint exchange method and the piece-wise linearity of the p_i functions (cf. Lemma 3.1) allow us to guarantee finite-time convergence of the algorithm under very general assumptions on the communication network.

C. Routine for Local Problem

In this subsection, we discuss how to practically compute the lex-min Lagrange multiplier of problem (8). For a fixed $\bar{\mathbf{y}}_i \in \mathbb{R}^S$, we can compute the dual of problem (8) when dualizing only the constraint $A_i \mathbf{z}_i \leq \bar{\mathbf{y}}_i + v_i \mathbf{1}$. The dual problem, in minimization form, is

$$\begin{aligned} \min_{\boldsymbol{\mu}_i} \quad & q_i(\boldsymbol{\mu}_i) \\ \text{subj. to} \quad & \boldsymbol{\mu}_i \geq \mathbf{0} \\ & \boldsymbol{\mu}_i^\top \mathbf{1} \leq R, \end{aligned} \tag{11}$$

where $q_i(\boldsymbol{\mu}_i) = \boldsymbol{\mu}_i^\top \bar{\mathbf{y}}_i - \min_{\mathbf{x}_i \in X_i} \big((c_i^\top + \boldsymbol{\mu}_i^\top A_i)\mathbf{x}_i\big)$.

We are interested in finding the lex-optimal solution of problem (11). To this end, we now formulate a finite-time algorithm based on outer approximations. To differentiate the notation with the distributed algorithm, here we denote the iterations with the letter k.

Algorithm 2 Local routine for problem (9)

```
**Initialization:** Set $k = 0$ and start at a feasible $\boldsymbol{\mu}_i^0 \in \mathbb{R}^S$
**Evolution:** Repeat until $q_i^{k-1}(\boldsymbol{\mu}_i^k) = q_i(\boldsymbol{\mu}_i^k)$
  **Compute** $g_i^k = \bar{\mathbf{y}}_i - A_i \mathbf{x}_i^k$ (subgradient of $q_i(\boldsymbol{\mu}_i^k)$) with
      $\mathbf{x}_i^k \in \operatorname*{argmin}_{\mathbf{x}_i \in \text{vert}(X_i)} \big(c_i^\top + (\boldsymbol{\mu}_i^k)^\top A_i\big)\mathbf{x}_i$      (12)
  **Update**
      $\boldsymbol{\mu}_i^{k+1} = \operatorname*{lex-min}_{\boldsymbol{\mu}_i \geq \mathbf{0},\, \boldsymbol{\mu}_i^\top \mathbf{1} \leq R} \; q_i^k(\boldsymbol{\mu}_i)$      (13)
   where
      $q_i^k(\boldsymbol{\mu}_i) \triangleq \max_{\tau \in \{0,\dots,k\}} \big(q_i(\boldsymbol{\mu}_i^\tau) + (\boldsymbol{\mu}_i - \boldsymbol{\mu}_i^\tau)^\top g_i^\tau\big)$
  **Set** $k \leftarrow k + 1$
```

Notice that the lex-optimal solution of problem (13) can be found by using any lexicographic solver for linear programs applied to the epigraph form of the problem. We now give the convergence result of Algorithm 2.

Proposition 3.3: Let $\bar{\mathbf{y}}_i \in \mathbb{R}^S$ be given and let the sequence $\{\boldsymbol{\mu}_i^k\}_{k \geq 0}$ be generated by Algorithm 2. Moreover, let $\boldsymbol{\mu}_i^{\text{LEX}}$ denote the lex-optimal solution of problem (11). Then, Algorithm 2 converges in finite time to $\boldsymbol{\mu}_i^{\text{LEX}}$, i.e., there exists $K_0 \in \mathbb{N}$ such that $\boldsymbol{\mu}_i^{K_0} = \boldsymbol{\mu}_i^{\text{LEX}}$ and

$$q_i^{K_0 - 1}(\boldsymbol{\mu}_i^{K_0}) = q_i(\boldsymbol{\mu}_i^{K_0}). \qquad \square$$

IV. Analysis of DiP-COGEN-MILP

In this section we analyze DiP-COGEN-MILP. We first provide an intermediate result, which states that the set of all Lagrange multipliers of problem (7) is a compact polyhedron, and each vertex is associated to a linear portion of the piece-wise linear p_i (cf. Lemma 3.1). We will denote by $\mathcal{L}_i(\mathbf{y}_i)$ the indices ℓ for which the maximum of the piece-wise linear p_i (cf. Lemma 3.1) is attained, i.e.,

$$\mathcal{L}_i(\mathbf{y}_i) \triangleq \{\ell \in L_i \mid \mathbf{y}_i^\top a_i^\ell + f_i^\ell = p_i(\mathbf{y}_i)\}.$$

Lemma 4.1: For a fixed $\mathbf{y}_i$, let $\mathcal{S}_i$ denote the set of all Lagrange multipliers of problem (4). Then,

(i) $\mathcal{S}_i$ is a compact polyhedron;

(ii) for all $\bar{\boldsymbol{\mu}}_i \in \text{vert}(\mathcal{S}_i)$, there exists an $\ell \in \mathcal{L}_i(\mathbf{y}_i)$ such that $-\bar{\boldsymbol{\mu}}_i = a_i^\ell$. □

A consequence of Lemma 4.1 is that the lexicographically minimal Lagrange multiplier of problem (4) can be used to compute a constraint of problem (7).

Corollary 4.2: Let $\mathbf{y}_i \in \mathbb{R}^S$ be given and let $p_i(\mathbf{y}_i)$ be the optimal cost and $\boldsymbol{\mu}_i^{\text{LEX}}$ the lex-min multiplier of problem (4). Then, there exists a $\ell \in \mathcal{L}_i(\mathbf{y}_i) \subseteq L_i$ such that $a_i^\ell = -\boldsymbol{\mu}_i^{\text{LEX}}$ and $f_i^\ell = p_i(\mathbf{y}_i) + \mathbf{y}_i^\top \boldsymbol{\mu}_i^{\text{LEX}}$. □

Notice that DiP-COGEN-MILP generates constraints as suggested by Corollary 4.2. But since L_i is a finite set for all i, this implies that the set of generated constraints during the algorithm evolution is finite. This key fact allows us to obtain finite-time convergence (cf. Theorem 4.4).

A. Convergence Result

In this section, we formalize the convergence result of DiP-COGEN-MILP, under the following assumption.

Assumption 4.3: The lex-optimal solution $(\mathbf{y}^\star, \boldsymbol{\rho}^\star)$ of problem (7) exists. □

We now state the main convergence result of the paper.

Theorem 4.4: Let Assumptions 2.1, 2.2 and 4.3 hold and let $\boldsymbol{\sigma}$ be defined as in (6) and $R, M > 0$ be sufficiently large. Consider the allocation sequences $\{\mathbf{y}_{[i]}^t\}_{t \geq 0}$ and the mixed-integer sequences $\{\mathbf{x}_{[i]}^t\}_{t \geq 0}$, $i \in \{1, \dots, N\}$, generated by DiP-COGEN-MILP. There exists a sufficiently large (finite) time $T > 0$ such that, for all $t \geq T$,

(i) for all $i \in \{1, \dots, N\}$, it holds $\mathbf{y}_{[i]}^t = \bar{\mathbf{y}} \in \mathbb{R}^{NS}$, optimal solution of problem (5);

(ii) The vector $(\mathbf{x}_{[1]}^t, \dots, \mathbf{x}_{[N]}^t) \in \mathbb{R}^{\sum_{i=1}^N n_i}$ is a feasible solution for problem (1), i.e., $\mathbf{x}_{[i]}^t \in X_i$ for all $i \in \{1, \dots, N\}$ and $\sum_{i=1}^N A_i \mathbf{x}_{[i]}^t \leq b$. □

The mixed-integer solution computed by the agents for $t \geq T$ is feasible for problem (1). However, in general, it is suboptimal, but a tight suboptimality bound can be computed by using [5, Theorem 4.3]. We highlight that a side contribution of the paper is that the agents reach consensus in finite time on an optimal solution of (5).

A remarkable property of the algorithm is that the agents can detect convergence in a fully distributed way, so that the following stopping criterion can be used: for uniformly jointly strongly connected graphs with period B, each agent i can conclude that convergence has occurred if its local solution $(\mathbf{y}_{[i]}^t, \boldsymbol{\rho}_{[i]}^t)$ has not changed after $2BN + 1$ communication rounds [18, Theorem 1].

V. Numerical computations

In this section, we provide a numerical example that validates the theoretical analysis of Section IV. For the sake of space, we only show how DiP-COGEN-MILP behaves on a single instance of problem (1). By using the same generation model as [5] (with $\hat{c}_i \in [0, 0.1]$), we generate a random MILP with $N = 30$ agents, $S = 3$ coupling constraints and resource vector b with entries in $[20, 120]$. Moreover, we set $R = 10 \cdot \|\boldsymbol{\mu}^\star\|_1$, with $\boldsymbol{\mu}^\star$ a Lagrange multiplier of problem (2) (so that Lemma 2.3 applies), and $M = 2 \cdot 10^3$. As for the communication network, we randomly generate an Erdős-Rényi undirected connected graph with edge probability equal to 0.1.

In Figure 1 (left), the evolution of the optimal cost of problem (9) (for all i), compared to the optimal cost of problem (7), is shown. In an outer approximation fashion, the algorithm selects infeasible points $(\mathbf{y}^t_{[i]}, \boldsymbol{\rho}^t_{[i]})$ that eventually become feasible (for problem (7)) and equal to each other. The figure highlights also that in the early iterations there are still insufficient constraints in the network, so that the global allocation estimates are attained at the bounding box. We point out that, by construction, the points $(\mathbf{y}^t_{[i]}, \boldsymbol{\rho}^t_{[i]})$ are always feasible for problem (5). In Figure 1 (right), we compare, for all i, the evolution of the cost of $(\mathbf{y}^t_{[i]}, \boldsymbol{\rho}^t_{[i]})$ for problem (5), with respect to its optimal cost. We see that initially the points have a high cost, but they become optimal in finite time, as expected from Theorem 4.4.

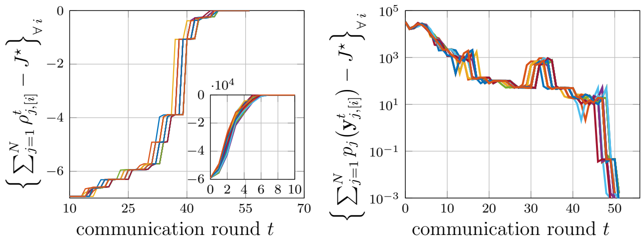

Fig. 1. On the left, the evolution of the optimal cost of problem (9), compared with the optimal cost $J^\star$ of problem (7) is reported for all i,. The inset figure shows the behavior of the algorithm in the early iterations. On the right, the evolution of the cost of problem (5) at the computed vectors $\mathbf{y}^t_{[i]}$, compared with $J^\star$, is shown on a logarithmic scale.

Now, we consider the sequence of distributed mixed-integer solutions $(\mathbf{x}^t_{[1]}, \ldots, \mathbf{x}^t_{[N]})$. In Figure 2, the evolution of primal feasibility (with respect to the coupling constraint) is shown. Notice that the algorithm is allowed to violate the constraints during the evolution, but, according to Proposition 2.4, it becomes feasible when convergence to an optimal solution of problem (5) occurs. We highlight that the algorithm in [5] could require a larger number of iterations than DiP-COGEN-MILP until feasibility is reached. However, such comparison is not completely meaningful, as one should also compare the local computation time and the quantity of exchanged information among agents.

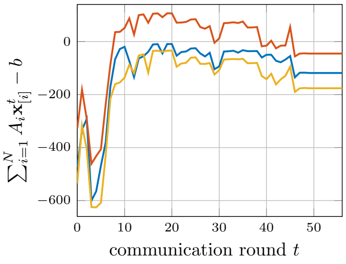

Fig. 2. Evolution of the distributed coupling constraint value. As soon as convergence is reached, the mixed-integer solution is guaranteed to be feasible for problem (1).

VI. Conclusions

In this paper, we proposed a novel finite-time distributed algorithm to compute a feasible solution of MILP (1), with suboptimality bounds. We considered a convexification of the original MILP, with restricted coupling constraints, to which we applied the primal decomposition approach of [5]. After a reformulation of the primal decomposition (resource allocation) problem to a LP with an exponential number of unknown constraints, we devised a distributed algorithm, in which agents generate constraints of the LP and exchange solution bases, until they eventually converge to the lex-optimal solution of the LP. The allocation estimates are used to compute tentative mixed-integer solutions, which are guaranteed to be feasible for problem (1) when convergence of the distributed algorithm has occurred. In order to guarantee finite-time convergence, we provided a local routine that allows for the generation of constraints by using the lexicographically minimal Lagrange multiplier of a small problem at each node. Finally, we provided a numerical example to validate the results.

References

[1] A. Richards and J. How, “Mixed-integer programming for control,” in *Proceedings of the American Control Conference*. IEEE, 2005, pp. 2676–2683.

[2] R. Vujanic, P. M. Esfahani, P. J. Goulart, S. Mariéthoz, and M. Morari, “A decomposition method for large scale MILPs, with performance guarantees and a power system application,” *Automatica*, vol. 67, no. 5, pp. 144–156, 2016.

[3] A. Falsone, K. Margellos, and M. Prandini, “A decentralized approach to multi-agent MILPs: finite-time feasibility and performance guarantees,” *preprint arXiv:1706.08788*, 2017.

[4] ——, “A distributed iterative algorithm for multi-agent MILPs: finite-time feasibility and performance characterization,” *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 563–568, 2018.

[5] A. Camisa, I. Notarnicola, and G. Notarstefano, “A primal decomposition method with suboptimality bounds for distributed mixed-integer linear programming,” *preprint arXiv:1811.03657*, 2018.

[6] A. Testa, A. Rucco, and G. Notarstefano, “Distributed mixed-integer linear programming via cut generation and constraint exchange,” *preprint arXiv:1804.03391*, 2018.

[7] S. Karaman and G. Inalhan, “Large-scale task/target assignment for UAV fleets using a distributed branch and price optimization scheme,” *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 13 310–13 317, 2008.

[8] H.-L. Choi, L. Brunet, and J. P. How, “Consensus-based decentralized auctions for robust task allocation,” *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 912–926, 2009.

[9] M. Bürger, G. Notarstefano, F. Bullo, and F. Allgöwer, “A distributed simplex algorithm for degenerate linear programs and multi-agent assignments,” *Automatica*, vol. 48, no. 9, pp. 2298–2304, 2012.

[10] D. Richert and J. Cortés, “Robust distributed linear programming,” *IEEE Transactions on Automatic Control*, vol. 60, no. 10, pp. 2567–2582, 2015.

[11] D. Richert and J. Cortes, “Distributed linear programming with event-triggered communication,” *SIAM Journal on Control and Optimization*, vol. 54, no. 3, pp. 1769–1797, 2016.

[12] M. Bürger, G. Notarstefano, and F. Allgöwer, “Locally constrained decision making via two-stage distributed simplex.” in *CDC-ECE*, 2011, pp. 5911–5916.

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

[14] I. Notarnicola and G. Notarstefano, “Constraint coupled distributed optimization: Relaxation and duality approach,” *preprint arXiv:1711.09221*, 2017.

[15] H. Lakshmanan and D. P. De Farias, “Decentralized resource allocation in dynamic networks of agents,” *SIAM Journal on Optimization*, vol. 19, no. 2, pp. 911–940, 2008.

[16] R. R. Meyer, “On the existence of optimal solutions to integer and mixed-integer programming problems,” *Mathematical Programming*, vol. 7, no. 1, pp. 223–235, 1974.

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

[18] M. Chamanbaz, G. Notarstefano, and R. Bouffanais, “Randomized constraints consensus for distributed robust linear programming,” *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 4973–4978, 2017.