A Primal Decomposition Method with Suboptimality Bounds for Distributed Mixed-Integer Linear Programming

This is the post peer-review accepted manuscript of:

A. Camisa, I. Notarnicola and G. Notarstefano, "A Primal Decomposition Method with Suboptimality Bounds for Distributed Mixed-Integer Linear Programming," 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, 2018, pp. 3391-3396.

The published version is available online at:

https://doi.org/10.1109/CDC.2018.8619597

A Primal Decomposition Method with Suboptimality Bounds for Distributed Mixed-Integer Linear Programming

Andrea Camisa[1], Ivano Notarnicola[1], Giuseppe Notarstefano[2]

***Abstract*— In this paper we deal with a network of agents seeking to solve in a distributed way Mixed-Integer Linear Programs (MILPs) with a coupling constraint (modeling a limited shared resource) and local constraints. MILPs are NP-hard problems and several challenges arise in a distributed framework, so that looking for suboptimal solutions is of interest. To achieve this goal, the presence of a linear coupling calls for tailored decomposition approaches. We propose a fully distributed algorithm based on a *primal decomposition* approach and a suitable tightening of the coupling constraints. Agents repeatedly update local allocation vectors, which converge to an optimal resource allocation of an approximate version of the original problem. Based on such allocation vectors, agents are able to (locally) compute a mixed-integer solution, which is guaranteed to be feasible after a sufficiently large time. Asymptotic and finite-time suboptimality bounds are established for the computed solution. Numerical simulations highlight the efficacy of the proposed methodology.**

I. Introduction

In this paper we consider constraint-coupled Mixed-Integer Linear Programs (MILP) with the following structure

$$\begin{aligned} \min_{\mathbf{x}_1,\dots,\mathbf{x}_N} \quad & \sum_{i=1}^{N} c_i^\top \mathbf{x}_i \\ \text{subj. to} \quad & \sum_{i=1}^{N} A_i \mathbf{x}_i \preceq b \\ & \mathbf{x}_i \in X_i, \qquad\qquad i \in \{1,\dots,N\} \end{aligned} \tag{1}$$

where $A_i \in \mathbb{R}^{S\times n_i}$, and $b \in \mathbb{R}^S$ describe S coupling constraints, and the local constraints X_i are mixed-integer polyhedral, compact and nonempty sets $X_i = \{\mathbf{x}_i \in \mathbb{Z}^{Z_i} \times \mathbb{R}^{R_i} \mid D_i\mathbf{x}_i \preceq d_i\}$, with $Z_i + R_i = n_i$. Throughout the paper, we use the symbols $\preceq$ and $\succeq$ to indicate element-wise inequality for vectors. As customary, we assume (1) is feasible. We want to address problem (1) in a distributed computation framework in which agents of a network communicate only with neighbors. Since these problems typically arise as instances of dynamic optimization problems that need to be repeatedly solved in real-time, we look for *fast* distributed algorithms to compute feasible (possibly suboptimal) solutions. Several problems in cyber-physical network systems as, e.g., in cooperative robotics or smart grids, can be cast as (1). An interesting scenario arises in Distributed Model Predictive Control, where a large set of dynamical systems, described by local constraints X_i (with both continuous and integer variables), needs to cooperatively solve a common control task and their states, outputs and/or inputs are coupled through coupling constraints.

We organize the relevent literature in two parts. First, we review primal and dual decomposition methods for significant types of problems. In the tutorial papers [1], [2], primal and dual decomposition techniques are reviewed. A distributed primal decomposition approach is proposed in [3] to solve smooth resource allocation problems. In [4] a regularized saddle-point algorithm for convex networked optimization problems is analyzed. In [5] distributed algorithms based on Laplacian-gradient dynamics are used to solve economic dispatch over digraphs. In [6] novel convergence rates for distributed resource allocation algorithms are proven. In [7] the convergence rate of a distributed algorithm for the minimization of a common cost under a resource constraint is established. Distributed algorithms for constraint-coupled problems have been proposed in [8], [9]. Second, we review parallel and distributed algorithms for MILPs. In [10] a Lagrange relaxation approach is used to decompose MILPs arising in demand response control in smart grids. In [11] a heuristic for embedded mixed-integer programming is proposed to obtain approximate solutions. First attempts of proposing a distributed approximate solution for MILPs are [12], [13]. Recently, a distributed algorithm, based on cutting-planes, has been proposed in [14] to solve common cost MILPs. Although this method could be applied to (1), each agent would know the entire solution vector (without, e.g., preserving privacy). A pioneering work on fast, master-based parallel algorithms to find approximate solutions of problem (1) is [15]. Here the key idea is to tighten the coupling constraint and then apply a dual decomposition method to get mixed-integer points violating the restricted coupling constraint but not the original one. In [16], an improved *iterative* tightening procedure has been proposed to obtain enhanced performance guarantees. In both works [15], [16], a master processing unit is needed. The first, fully distributed implementation of the above dual decomposition based methodology is proposed in [17].

In this paper we pursue the same main goal as in [15]–[17], namely to provide a fast algorithm to compute a feasible mixed-integer solution of (1) with guaranteed suboptimality bounds. In particular, we consider a *distributed computation framework over networks*, even though we will point out how to implement the method in a parallel architecture. Differently from the above works, based on dual decomposition, we propose a distributed algorithm

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).

[1]A. Camisa and I. Notarnicola are with the Department of Engineering, Università del Salento, Lecce, Italy, name.lastname@unisalento.it.

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

relying on *primal* decomposition. A building block for the proposed scheme is a distributed algorithm to solve convex programs. It is based on a relaxation approach combined with a distributed primal decomposition scheme. Although the new algorithm strongly relies on the scheme proposed in [9], it represents a contribution per se. Indeed, we give a new insightful interpretation as a primal decomposition scheme with distributed negotiation among the agents of optimal local allocations. This distributed algorithm is used to solve a LP approximation of (1) with restricted coupling constraints. The resulting (local) allocation vectors allow agents to retrieve a mixed-integer solution by solving a local MILP with minimal violation of the allocation vector. We are able to: (i) establish tight (asymptotic and finite-time) restrictions of the coupling constraint such that the computed mixed-integer solutions are feasible for (1), and (ii) provide proper (asymptotic and finite-time) suboptimality bounds. Preliminary numerical computations highlight the tightness of our restriction and show low suboptimality gap.

The paper unfolds as follows. In Section II we recall some preliminaries. In Section III we derive our distributed algorithm, analyzed in Section IV. Numerical computations are provided in Section V. Due to space constrains all proofs are omitted and will be provided in a forthcoming document.

II. Preliminaries and Distributed Framework

A. LP Approximation of the Target MILP and its Properties

In order to design fast algorithms to find approximate solutions of (1), following [15], it is useful to introduce an approximate version of the problem. The approximation is a linear program obtained by replacing the local (mixed-integer) constraints X_i with their polyhedral convex hull denoted by $\text{conv}(X_i)$, and the total resource vector b with a restricted resource vector $b-\boldsymbol{\sigma}$, with $\boldsymbol{\sigma} \succeq \mathbf{0} \in \mathbb{R}^S$. In order to clearly distinguish the decision variables of the original (MILP) and approximated (LP) problems, from now on, we will use $\mathbf{z}_i \in \mathbb{R}^{n_i}$ to denote the continuous counterpart of the mixed-integer variable $\mathbf{x}_i \in \mathbb{Z}^{Z_i} \times \mathbb{R}^{R_i}$. The LP approximation can thus be written as

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

We now briefly discuss some properties of this approximation. First of all, for $\boldsymbol{\sigma} = \mathbf{0}$, problem (2) is a relaxation of (1). A well-known property of such a relaxed problem is that its dual problem coincides with the dual of (1), see e.g., [18]. Since in general MILP (1) has a duality gap and problem (2) enjoys strong duality, then the costs of (2), with $\boldsymbol{\sigma} = \mathbf{0}$, and (1) differ exactly by the MILP duality gap.

The high-level idea motivating the restriction $\boldsymbol{\sigma}$ is to exploit problem (2) to compute mixed-integer points satisfying the local constraints X_i, but violating the coupling constraint no more than $\boldsymbol{\sigma}$. We will see in the next section that not only the magnitude, but also the meaning of the restriction that we use in our approach is different from the ones proposed in dual decomposition schemes. A common, standing assumption required for methods based on a given restriction $\boldsymbol{\sigma} \succeq \mathbf{0}$ of the coupling constraints is the following.

Assumption 2.1 (On the restricted LP): Problem (2) is feasible and its optimal solution is unique. □

Finally, we recall a result in [19] which will play a key role in our analysis, shows that any vertex (including the optimal one) of (2) is partially mixed-integer, and provides a bound on the number of agents whose solution is not mixed-integer.

Lemma 2.2 ([19], Theorem 1): Let Assumption 2.1 hold and let $(\bar{\mathbf{z}}_1,\dots,\bar{\mathbf{z}}_N)$ be a vertex of problem (2). There exists a set $I_{\text{INT}} \subseteq \{1,\dots,N\}$, with cardinality at least $|I_{\text{INT}}| \geq N - S - 1$, such that $\bar{\mathbf{z}}_i \in X_i$ for all $i \in I_{\text{INT}}$. □

B. Primal Decomposition

Primal decomposition allows to obtain a master-subproblem architecture from constraint-coupled convex programs such as (2). Local *allocation vectors* at each node, adding up to the total resource $b-\boldsymbol{\sigma}$, are iteratively adjusted until they converge to the optimal allocation. Thus, each node can asymptotically retrieve its portion of optimal solution of (2) by using its local allocation, [20], [21].

Formally, in a primal decomposition approach, problem (2) can be restated into a hierarchical master-subproblem formulation, with a *master problem*

$$\begin{aligned} \min_{\mathbf{y}_1,\dots,\mathbf{y}_N} \quad & \sum_{i=1}^{N} p_i(\mathbf{y}_i) \\ & \sum_{i=1}^{N} \mathbf{y}_i = b - \boldsymbol{\sigma} \\ & \mathbf{y}_i \in Y_i, \qquad\qquad i \in \{1,\dots,N\} \end{aligned} \tag{3}$$

where, for each $i \in \{1,\dots,N\}$, $p_i(\mathbf{y}_i)$ is defined as the optimal cost of the i-th *subproblem*

$$\begin{aligned} p_i(\mathbf{y}_i) = \min_{\mathbf{z}_i} \quad & c_i^\top \mathbf{z}_i \\ \text{subj. to} \quad & A_i \mathbf{z}_i \preceq \mathbf{y}_i \\ & \mathbf{z}_i \in \text{conv}(X_i), \end{aligned} \tag{4}$$

and $Y_i \subseteq \mathbb{R}^S$ denotes the set of $\mathbf{y}_i$ such that (4) is well-posed (i.e., the set of $\mathbf{y}_i$ such that there exists at least a $\bar{\mathbf{z}}_i \in \text{conv}(X_i)$ with $A_i \bar{\mathbf{z}}_i \preceq \mathbf{y}_i$).

Due to the presence of Y_i, solving (3) is not trivial, especially in a distributed computation framework. Several works as, e.g., [3], investigate a simplified set-up without local constraints, so that $Y_i \equiv \mathbb{R}^S$. Recently, in [9] a methodology to overcome this issue has been proposed. We will pursue the same idea to devise a distributed primal decomposition approach for (2), that will act as a building block for our distributed algorithm.

C. Distributed Computation Framework

We consider a network of N processors communicating according to a *connected* and *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. If $(i,j) \in \mathcal{E}$, then nodes i and j can

exchange information (and in fact also $(j,i) \in \mathcal{E}$). We denote by $\mathcal{N}_i$ the set of *neighbors* of node i in $\mathcal{G}$, i.e., $\mathcal{N}_i = \{j \in \{1,\dots,N\} \mid (i,j) \in \mathcal{E}\}$. Each node i knows only its local constraint X_i, its portion c_i of the total cost and the matrix A_i of the coupling constraint. The goal is that each agent computes an approximation for its portion $\mathbf{x}_i^\star$ of an optimal solution of (1) by means of local communication with neighboring agents only.

III. Distributed Primal Decomposition for Feasible MILP Solution

In this section we introduce our distributed optimization algorithm and discuss implementation features. We provide a constructive argument leading to the proposed method, consisting of two routines described in the next two subsections.

A. Distributed Primal Decomposition Method for LP Solution over Networks

Following the approach proposed in [9], we can derive a distributed algorithm to solve (2) by combining a (distributed) primal decomposition method with a relaxation approach. The algorithm reads as follows. Each agent updates a local vector $((\mathbf{z}_i^t, v_i^t), \boldsymbol{\mu}_i^t)$ as a primal-dual optimal solution pair of (6), with $M > 0$. Then, it gathers $\boldsymbol{\mu}_j^t$ from $j \in \mathcal{N}_i$ and updates its local estimate of the optimal allocation vector $\mathbf{y}_i^{t+1}$ with (7), where α^t is an appropriate step-size sequence.

Assumption 3.1 (Diminishing Step-size): The step-size sequence $\{\alpha^t\}_{t\geq 0}$, with each $\alpha^t \geq 0$, satisfies the conditions $\sum_{t=0}^{\infty} \alpha^t = \infty$, $\sum_{t=0}^{\infty} \left(\alpha^t\right)^2 < \infty$. □

We can now state the convergence properties of the proposed algorithm, in which agents solve the relaxed (always feasible) version (6) of problem (4), and then update their resource allocation vector $\mathbf{y}_i$ according to a linear update.

Proposition 3.2: Let Assumptions 2.1 and 3.1 hold and let the local allocation vectors $\mathbf{y}_i^0$ be initialized such that $\sum_{i=1}^N \mathbf{y}_i^0 = b - \boldsymbol{\sigma}$. Then, there exists a sufficiently large $M > 0$ for which the distributed algorithm (6), (7) generates an allocation vector sequence $\{\mathbf{y}_1^t, \dots, \mathbf{y}_N^t\}_{t\geq 0}$ such that

(i) $\sum_{i=1}^N \mathbf{y}_i^t = b - \boldsymbol{\sigma}$, for all $t \geq 0$,

(ii) $\lim_{t\to\infty} \|\mathbf{y}_i^t - \mathbf{y}_i^\star\| = 0$ for all $i \in \{1,\dots,N\}$, where $(\mathbf{y}_1^\star, \dots, \mathbf{y}_N^\star)$ is an optimal solution of (3);

(iii) any limit point of the primal sequence $\{\mathbf{z}_1^t, \dots, \mathbf{z}_N^t\}_{t\geq 0}$ associated to $\{\mathbf{y}_1^t, \dots, \mathbf{y}_N^t\}_{t\geq 0}$, say $(\mathbf{z}_1^\infty, \dots, \mathbf{z}_N^\infty)$, is an optimal solution of problem (2), and the corresponding cost $\sum_{i=1}^N c_i^\top \mathbf{z}_i^\infty$ is equal to the optimal cost of (2). □

As a corollary of *(iii)*, the sequences $\{v_i^t\}_{t\geq 0}$, $i \in \{1,\dots,N\}$, converge to zero.

B. Feasible Mixed-Integer Solution Computation

The distributed algorithm (6)-(7), in general, does not provide asymptotically a mixed-integer solution. Thus, if $\mathbf{y}_i^\infty$ is the asymptotic assignment of agent i, the MILP

$$\begin{aligned} \min_{\mathbf{x}_i} \ & c_i^\top \mathbf{x}_i \\ \text{subj. to } & A_i \mathbf{x}_i \preceq \mathbf{y}_i^\infty \\ & \mathbf{x}_i \in X_i, \end{aligned} \tag{5}$$

admits an optimal solution which is also the solution of problem (4) with $\mathbf{y}_i = \mathbf{y}_i^\infty$, for (at least) $N - S - 1$ agents. Thus, if the remaining (at most) $S+1$ agents find a solution to (5), all the agents have a mixed-integer point satisfying both the local constraints and the coupling constraints. To this end, it is sufficient that for each agent i for which the LP solution is not mixed integer, there exists *at least one* feasible point $\bar{\mathbf{x}}_i \in X_i$ such that $A_i \bar{\mathbf{x}}_i \preceq \mathbf{y}_i^\infty$.

However, this does not happen in general because the negotiated local allocation vectors are based on local constraints $\text{conv}(X_i)$ rather than X_i. Thus, we adopt a relaxation approach similar to the one proposed for the LP approximation. That is, we let agents solve a relaxed version of (5) in which the cost $c_i^\top \mathbf{x}_i$ must be minimized while allowing for a minimal violation of the coupling constraint. This can be done by solving (8), where we use the notation lex-min to indicate that ρ_i and ξ_i are minimized in a lexicographic order (we will discuss next how to solve (8)).

C. Distributed Algorithm: Description and Implementation Discussion

In the following table we summarize our Distributed Primal Decomposition for Feasible MILP Solution (DiP-FEAS-MILP) algorithm from the perspective of node i.

Distributed Algorithm DiP-FEAS-MILP

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

Evolution:

Compute $\boldsymbol{\mu}_i^t$ as a dual optimal solution of

$$\begin{aligned} \min_{\mathbf{z}_i, v_i} \ & c_i^\top \mathbf{z}_i + M v_i \\ \text{subj. to } \ \boldsymbol{\mu}_i : \ & A_i \mathbf{z}_i \preceq \mathbf{y}_i^t + v_i \mathbf{1} \\ & \mathbf{z}_i \in \text{conv}(X_i), \ v_i \geq 0 \end{aligned} \tag{6}$$

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

$$\mathbf{y}_i^{t+1} = \mathbf{y}_i^t + \alpha^t \sum_{j\in\mathcal{N}_i} \left(\boldsymbol{\mu}_i^t - \boldsymbol{\mu}_j^t\right) \tag{7}$$

Compute $\mathbf{x}_i^t$ as an optimal solution of

$$\begin{aligned} \text{lex-min}_{\rho_i, \xi_i, \mathbf{x}_i} \ & \rho_i \\ \text{subj. to } \ & c_i^\top \mathbf{x}_i \leq \xi_i \\ & A_i \mathbf{x}_i \preceq \mathbf{y}_i^t + \rho_i \mathbf{1} \\ & \mathbf{x}_i \in X_i, \ \rho_i \geq 0. \end{aligned} \tag{8}$$

Remark 3.3 (Parallel Implementation): We point out that a parallel implementation of the DiP-FEAS-MILP distributed algorithm can be obtained by letting a central unit update the allocation vectors $\mathbf{y}_i^t$. This can be done by means of a centralized subgradient replacing (7). □

From an implementation point of view, in most cases an explicit description of $\text{conv}(X_i)$ in terms of inequalities might not be available. Then, column generation techniques can be used to approximate $\text{conv}(X_i)$, see, e.g., [22]. However, since the algorithm only requires $\boldsymbol{\mu}_i^{t+1}$ to evolve,

agents can obtain an estimate by *locally* running a dual subgradient method to find a dual optimal solution of (6) without resorting to a description of $\text{conv}(X_i)$. Indeed, being the Lagrangian of (6) a linear function of the primal variable $\mathbf{z}_i$, a subgradient of the dual function at $\boldsymbol{\mu}_i^t$ can be easily computed as $A_i\bar{\mathbf{x}}_i - \mathbf{y}_i^t$, with $\bar{\mathbf{x}}_i$ an optimal solution of

$$\min_{\mathbf{x}_i \in X_i} (c_i^\top + (\boldsymbol{\mu}_i^t)^\top A_i)\mathbf{x}_i = \min_{\mathbf{z}_i \in \text{conv}(X_i)} (c_i^\top + (\boldsymbol{\mu}_i^t)^\top A_i)\mathbf{z}_i,$$

where the equality follows from the linearity of the cost.

Moreover, if agent i wants to know $J_i^{\text{LP},t} \triangleq c_i^\top \mathbf{z}_i^t + M v_i^t$ at a given time instant t, by strong duality, can evaluate $\min_{\mathbf{x}_i \in X_i}(c_i^\top + (\boldsymbol{\mu}_i^t)^\top A_i)\mathbf{x}_i - (\boldsymbol{\mu}_i^t)^\top \mathbf{y}_i^t$. The sum of these quantities will appear in the suboptimality bound in finite-time, so that it is computable in a distributed way by using an average consensus algorithm.

We now show a simple way to perform the lex-min optimization in (8). First, agents compute ρ_i^t as the optimal cost of

$$\begin{aligned} \min_{\rho_i, \mathbf{x}_i} \ & \rho_i \\ \text{subj. to } & A_i\mathbf{x}_i \preceq \mathbf{y}_i^t + \rho_i \mathbf{1} \\ & \mathbf{x}_i \in X_i, \ \ \rho_i \geq 0. \end{aligned} \tag{9}$$

Then, they compute $\mathbf{x}_i^t$ as an optimal solution of

$$\begin{aligned} \min_{\mathbf{x}_i} \ & c_i^\top \mathbf{x}_i \\ \text{subj. to } & A_i\mathbf{x}_i \preceq \mathbf{y}_i^t + \rho_i^t \mathbf{1} \\ & \mathbf{x}_i \in X_i. \end{aligned} \tag{10}$$

IV. Feasibility Guarantees and Suboptimality Bounds of DiP-FEAS-MILP

In this section we analyze the performance of DiP-FEAS-MILP distributed algorithm. We first derive the restriction on the coupling constraint needed to ensure asymptotic feasibility. Then we provide (asymptotic and finite-time) feasibility guarantees and suboptimality bounds, which can be computed once the algorithm solution is available.

A. Tight Restriction for Asymptotic Feasibility

We derive an upper bound of the violation ρ_i for any *admissible* LP allocation $\mathbf{y}_i^\infty$. This allows us to find a minimal, a-priori restriction $\boldsymbol{\sigma}$, of the coupling constraint. For all i, consider a "lower bound" vector $\mathbf{L}_i$ with components

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

where A_i^s is the s-th row of A_i. $\mathbf{L}_i^s$ can be equivalently computed by minimizing over $\text{conv}(X_i)$. Thus, each admissible allocation $\mathbf{y}_i^\infty$ satisfies $\mathbf{y}_i^\infty \succeq \mathbf{L}_i$. To compute the maximum violation ρ_i, due to the mismatch between $\mathbf{x}_i^t$ and $\mathbf{z}_i^t$, let $(\mathbf{x}_i^{\text{L}}, \tilde{\ell}_i)$ be an optimal solution to

$$\begin{aligned} \min_{\mathbf{x}_i, \ell_i} \ & \ell_i \\ \text{subj. to } & A_i\mathbf{x}_i - \mathbf{L}_i \preceq \ell_i \mathbf{1} \\ & \mathbf{x}_i \in X_i, \ \ell_i \in \mathbb{R}. \end{aligned}$$

Then, denoting $\mathbf{x}_i^\infty$ a solution to (8) with $\mathbf{y}_i^\infty$, it holds

$$A_i\mathbf{x}_i^\infty - \mathbf{y}_i^\infty \preceq A_i\mathbf{x}_i^\infty - \mathbf{L}_i \preceq \Big(\max_{s \in \{1,\dots,S\}} A_i^s \mathbf{x}_i^{\text{L}} - \mathbf{L}_i^s\Big)\mathbf{1},$$

for any admissible $\mathbf{y}_i^\infty$. Therefore, a restriction $\boldsymbol{\sigma}$, guaranteeing feasibility of $(\mathbf{x}_1^\infty, \dots, \mathbf{x}_N^\infty)$ with respect to MILP (1), can be obtained by setting $\boldsymbol{\sigma} = \sigma^{\text{ASY}}\mathbf{1}$, with

$$\sigma^{\text{ASY}} \triangleq (S+1) \max_{i \in \{1,\dots,N\}} \max_{s \in \{1,\dots,S\}} \Big(A_i^s \mathbf{x}_i^{\text{L}} - \mathbf{L}_i^s\Big). \tag{11}$$

We point out that the restriction (11) can be computed in a distributed way by using a max-consensus algorithm. In Figure 1, we give an illustrative example of the restriction with 3 constraints and 2 agents.

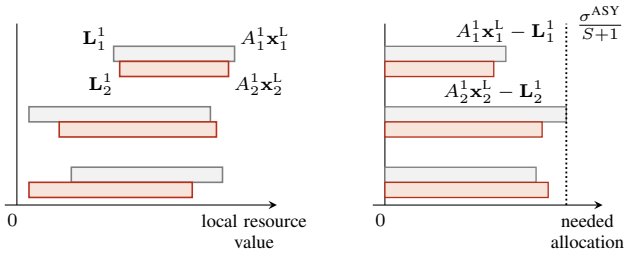

Fig. 1. Graphical representation of the restriction $\boldsymbol{\sigma}$ for a problem with $S = 3$ constraints and $N = 2$ agents.

Before proceeding with the analysis, we compare our restriction (11) with the one proposed in [15], i.e.,

$$\boldsymbol{\sigma}_s = S \max_{i \in \{1,\dots,N\}} \Big(\max_{\mathbf{x}_i \in X_i} A_i^s \mathbf{x}_i - \mathbf{L}_i^s\Big), \tag{12}$$

with $s \in \{1, \dots, S\}$. Notice that, in place of the worst-case $\max_{\mathbf{x}_i \in X_i} A_i^s \mathbf{x}_i$, we use $A_i^s \mathbf{x}_i^{\text{L}}$, obtained at the "lowest" feasible point $\mathbf{x}_i^{\text{L}} \in X_i$ closest to $\mathbf{L}_i$. On the other hand, our scaling factor $S+1$ is larger than S. Finally, we notice that when all the A_i^s are equal for all $s \in \{1, \dots, S\}$, then the two restrictions coincide, except for the scaling factor. This situation occurs for very structured problems, such as in the case study analyzed in [15], for which our restriction can be obtained by scaling (12) by $(S+1)/S$. Finally, in certain problem set-ups, such as in partial shipments [19], the local constraint sets are such that $\mathbf{0} \in X_i$ and $A_i\mathbf{x}_i \succeq \mathbf{0}$ for all $\mathbf{x}_i \in X_i$. Then, by construction, $\mathbf{L}_i = \mathbf{0}$ and $\tilde{\ell}_i = 0$ for all $i \in \{1, \dots, N\}$. It follows that, for these special problems, σ^{ASY} boils down to zero, so that no restriction is needed. This holds true in general for problems in which $A_i\mathbf{x}_i^{\text{L}} = \mathbf{L}_i$ for all $i \in \{1, \dots, N\}$ for some $\mathbf{x}_i^{\text{L}} \in X_i$.

B. Asymptotic Guarantees

In this subsection we assume the allocation vectors are initialized such that $\sum_{i=1}^N \mathbf{y}_i^0 = b - \boldsymbol{\sigma}$ with $\boldsymbol{\sigma} = \sigma^{\text{ASY}}\mathbf{1}$.

Theorem 4.1 (Feasibility): Let Assumption 2.1 and Assumption 3.1 hold. Consider the allocation vector sequence $\{\mathbf{y}_1^t, \dots, \mathbf{y}_N^t\}_{t \geq 0}$ generated by DiP-FEAS-MILP converging to $(\mathbf{y}_1^\infty, \dots, \mathbf{y}_N^\infty)$. Let $(\rho_i^\infty, \xi_i^\infty, \mathbf{x}_i^\infty)$ be a lex-optimal solution of (8) corresponding to $\mathbf{y}_i^\infty$ for all $i \in \{1, \dots, N\}$. Then, $(\mathbf{x}_1^\infty, \dots, \mathbf{x}_N^\infty)$ is a feasible solution for MILP (1), i.e., $\mathbf{x}_i^\infty \in X_i$ for all $i \in \{1, \dots, N\}$ and $\sum_{i=1}^N A_i\mathbf{x}_i^\infty \preceq b$. $\square$

Before stating the suboptimality bound, we need constraint qualification on the restricted coupling constraint.

Assumption 4.2 (Slater Constraint Qualification): There exists a point $(\widehat{\mathbf{z}}_1, \dots, \widehat{\mathbf{z}}_N)$ such that $\widehat{\mathbf{z}}_i \in \text{conv}(X_i)$

$$\zeta_{\widehat{\mathbf{z}}} \triangleq \min_{s\in\{1,\dots,S\}} \Big(b_s - \boldsymbol{\sigma}_s - \sum_{i=1}^{N} A_i^s \widehat{\mathbf{z}}_i\Big) > 0. \tag{13}$$

The following result establishes an asymptotic suboptimality bound for DiP-FEAS-MILP.

Theorem 4.3 (Suboptimality Bound): Consider the same assumptions and quantities of Theorem 4.1 and let also Assumption 4.2 hold. Then, $(\mathbf{x}_1^\infty, \dots, \mathbf{x}_N^\infty)$ satisfies the suboptimality bound

$$\begin{aligned}\sum_{i=1}^{N} c_i^\top \mathbf{x}_i^\infty - J^{\text{MILP}} \le \sum_{s=1}^{S+1} (c_{i_s}^\top \mathbf{x}_{i_s}^\infty - p_{i_s}(\mathbf{y}_{i_s}^\infty)) \\ + \frac{\sigma^{\text{ASY}}}{\zeta_{\widehat{\mathbf{z}}}} \sum_{i=1}^{N} \Big(c_i^\top \widehat{\mathbf{z}}_i - p_i(\mathbf{y}_i^\infty)\Big)\end{aligned} \tag{14}$$

where J^{MILP} is the optimal cost of (1), $p_i(\mathbf{y}_i)$ is defined in (4), $(\widehat{\mathbf{z}}_1, \dots, \widehat{\mathbf{z}}_N)$ is a Slater point and i_s is the index sequence of the (at most) $S+1$ agents with $\mathbf{z}_i^\infty \notin X_i$ (where $\mathbf{z}_i$ is the local optimal solution corresponding to $\mathbf{y}_i^\infty$). □

C. Finite-Time Guarantees

In this subsection, we establish finite-time guarantees for DiP-FEAS-MILP. For the asymptotic results, we used the restriction (11). Here, we consider an augmented restriction $\boldsymbol{\sigma} = (\sigma^{\text{ASY}} + \delta)\mathbf{1}$, with an arbitrary (small) $\delta > 0$.

Theorem 4.4 (Finite-time Feasibility): Let Assumptions 2.1 and 3.1 hold. Consider the mixed-integer sequence $\{\mathbf{x}_1^t, \dots, \mathbf{x}_N^t\}_{t\ge 0}$ generated by DiP-FEAS-MILP with $\mathbf{y}_i^0$ initialized such that $\sum_{i=1}^N \mathbf{y}_i^0 = b - (\sigma^{\text{ASY}} + \delta)\mathbf{1}$ for a given $\delta > 0$. There exists a sufficiently large (finite) time $T_\delta > 0$ such that the vector $(\mathbf{x}_1^t, \dots, \mathbf{x}_N^t)$ is a feasible solution for problem (1), i.e., $\mathbf{x}_i^t \in X_i$ for all $i \in \{1, \dots, N\}$ and $\sum_{i=1}^N A_i \mathbf{x}_i^t \preceq b$, for all $t \ge T_\delta$. □

In general, the amount of time T_δ needed to ensure feasibility increases as δ approaches zero. Conversely, the faster to guarantee feasibility the larger the restriction should become. Now, we introduce a suboptimality bound on the computed solution that can be evaluated when the algorithm is halted prematurely.

Theorem 4.5 (Finite-time Suboptimality Bound): Consider the same assumptions and quantities of Theorem 4.4. Moreover, let Assumption 4.2 hold and let $\epsilon_i > 0$ be arbitrary small numbers, for $i \in \{1, \dots, N\}$. Then, there exists a sufficiently large (finite) time $T_\delta > 0$ such that the vector $(\mathbf{x}_1^t, \dots, \mathbf{x}_N^t)$ satisfies for all $t \ge T_\delta$ the suboptimality bound

$$\begin{aligned}\sum_{i=1}^{N} c_i^\top \mathbf{x}_i^t - J^{\text{MILP}} \le \sum_{i=1}^{N} (c_i^\top \mathbf{x}_i^t - J_i^{\text{LP},t}) \\ + \sum_{i=1}^{N} \|\boldsymbol{\mu}_i^t\|_1 \epsilon_i + \Gamma(\sigma^{\text{ASY}} + \delta),\end{aligned} \tag{15}$$

where $\Gamma = \frac{1}{\zeta_{\widehat{\mathbf{z}}}} \sum_{i=1}^{N} \Big(\max_{\mathbf{x}_i \in X_i} c_i^\top \mathbf{x}_i - \min_{\mathbf{x}_i \in X_i} c_i^\top \mathbf{x}_i\Big)$, with $\widehat{\mathbf{z}}$ a Slater point, J^{MILP} is the optimal cost of (1) and $J_i^{\text{LP},t}$ is the cost of (6) at time t. □

V. Numerical Computations

In this section, we corroborate the theoretical work with numerical computations. First, we show that the distributed algorithm achieves feasibility in finite time (cf. Theorem 4.4) on a random MILP with duality gap. Second, we compare our approach with methods based on dual decomposition by generating (unstructured) random MILPs with duality gap.

The generation model for random problems consists of two phases. First, N feasible LPs are generated with a model inspired by [25] to get the local sets $X_i \subset \mathbb{Z} \times \mathbb{R}$ and the cost vectors c_i. We generate polyhedral constraints $D_i \mathbf{x}_i \preceq d_i$ with random entries uniformly in $[0, 1]$ for $D_i \in \mathbb{R}^{6\times 2}$ and entries in $[0, 40]$ for $d_i \in \mathbb{R}^6$. To make sure that X_i is compact, we add box constraints $-60 \cdot \mathbf{1} \le \mathbf{x}_i \le 60 \cdot \mathbf{1}$. The cost vector is then calculated as $c_i = D_i^\top \hat{c}_i$, where $\hat{c}_i$ has entries in $[0, 5]$. Second, we add coupling constraints by generating random A_i matrices with entries in $[0, 1]$ and a resource vector b with values in different intervals.

A. Finite-time feasibility

We generated a random MILP with $N = 100$ agents, $S = 10$ coupling constraints and resource vector b with entries in $[-600, -500]$. In order to apply Theorem 4.4, we set the restriction to $\boldsymbol{\sigma} = (\sigma^{\text{ASY}} + \delta)\mathbf{1}$, with $\delta = 0.4$ and σ^{ASY} computed with (11). Remarkably, the solution computed by the algorithm is feasible for the original MILP for all t. In Figure 2 (left), we evaluated the quantity $\sum_{i=1}^N \rho_i^t - \sigma^{\text{ASY}}$ which in less than 50 communication rounds went below 0, providing a sufficient condition for feasibility The coupling constraint value is shown in Figure 2 (right), where the horizontal dashed line represents $-\sigma^{\text{ASY}}$. The figure highlights two important facts: (i) the algorithm seems to evolve in an interior-point fashion, and (ii) the coupling constraint value is always under $-\sigma^{\text{ASY}}$, which suggests that there is still room for a tighter restriction.

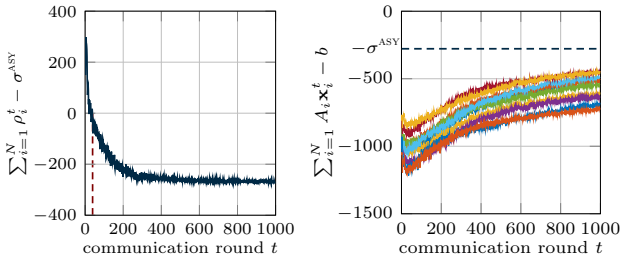

Fig. 2. Evolution of the sum of local MILP violations compared to the restriction σ^{ASY} (left) and coupling constraint value for the computed MILP solution (right).

B. Performance comparison

We compared the performance of our algorithm with [15] whose restriction is defined in (12). We did not run the distributed algorithm to compute an optimal allocation $(\mathbf{y}_1^\infty, \dots, \mathbf{y}_N^\infty)$, but rather we explicitly solved problem (3) and then computed $(\rho_i^\infty, \mathbf{x}_i^\infty)$ with (8). We generated problems with $N = 50$ agents and $S = 10$ coupling constraints.

In the numerical computations, we observed that the method in [15] was not applicable in several generated instances since the restricted LPs were unfeasible. Therefore, we first evaluated the fraction of generated problems for which each methodology was applicable. It turned out that, for resource vectors with components randomly chosen in $[-400, -300]$, our methodology could be applied in 99.33% of cases, out of 300 instances, whereas the method [15] never satisfied the needed assumptions (since the restriction resulted in 0% of feasible restricted LPs).

Then, we generated problems with entries of b in $[300, 400]$ in order to make both methods applicable. This made it difficult to find meaningful problems (i.e., with duality gap), which were only 20.59% out of 2778 feasible problems. For those meaningful problems, we solved the centralized MILP and we compared the solution performance of DiP-FEAS-MILP and [15]. This could be done for 17.66% problems that were feasible for both methods. In particular, we evaluated the relative suboptimality $|(\sum_{i=1}^{N} c_i^\top \mathbf{x}_i^\star - J^{\text{MILP}})/J^{\text{MILP}}|$, where $\mathbf{x}_i^\star$ is the solution found by either DiP-FEAS-MILP or [15]. We also evaluated the relative restriction magnitude, i.e., $\|\boldsymbol{\sigma}\|/\|b\|$. In Figure 3, comparison histograms of the relative suboptimality of both methods and of the relative restriction magnitude are shown. A further investigation to be carried out consists of comparing the restriction magnitude of DiP-FEAS-MILP with the time-varying restriction proposed in [16].

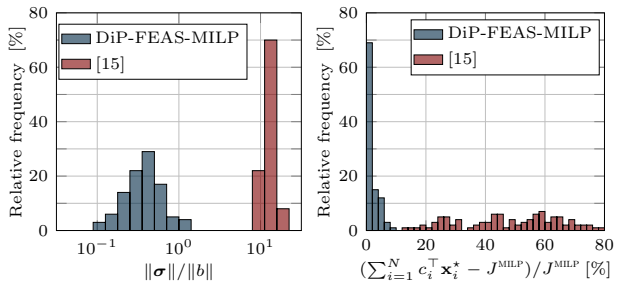

Fig. 3. Relative restriction magnitude (left) and solution suboptimality (right) comparison for 100 random MILPs having nonzero duality gap with $S = 10$ coupling constraints and $N = 50$ agents.

VI. Conclusions

In this paper we proposed a distributed algorithm for multi-agent MILPs by employing a primal decomposition approach. Based on a proper tightening of the coupling constraints, agents update local allocation vectors that asymptotically converge to an optimal resource allocation of a convexified version of the original MILP. Such vectors allow agents to compute a mixed-integer solution satisfying both the local constraints and the (original) coupling constraint. Suboptimality bounds and feasibility guarantees for asymptotic and finite-time solutions are also established. Numerical simulations corroborate our approach showing pros and cons with respect to state-of-the-art methods.

References

[1] 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.

[2] I. Necoara, V. Nedelcu, and I. Dumitrache, "Parallel and distributed optimization methods for estimation and control in networks," *Journal of Process Control*, vol. 21, no. 5, pp. 756–766, 2011.

[3] 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.

[4] 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.

[5] A. Cherukuri and J. Cortés, "Distributed generator coordination for initialization and anytime optimization in economic dispatch," *IEEE Trans. on Control of Network Sys.*, vol. 2, no. 3, pp. 226–237, 2015.

[6] A. Nedić, A. Olshevsky, and W. Shi, "Improved convergence rates for distributed resource allocation," *preprint arXiv:1706.05441*, 2017.

[7] I. Necoara, "Random coordinate descent algorithms for multi-agent convex optimization over networks," *IEEE Trans. on Automatic Control*, vol. 58, no. 8, pp. 2001–2012, 2013.

[8] 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.

[9] I. Notarnicola and G. Notarstefano, "Constraint coupled distributed optimization: a relaxation and duality approach," *preprint arXiv:1711.09221*, 2017.

[10] S.-J. Kim and G. B. Giannakis, "Scalable and robust demand response with mixed-integer constraints," *IEEE Trans. on Smart Grid*, vol. 4, no. 4, pp. 2089–2099, 2013.

[11] R. Takapoui, N. Moehle, S. Boyd, and A. Bemporad, "A simple effective heuristic for embedded mixed-integer quadratic programming," *International Journal of Control*, pp. 1–11, 2017.

[12] Y. Kuwata and J. P. How, "Cooperative distributed robust trajectory optimization using receding horizon MILP," *IEEE Trans. on Control Systems Tech.*, vol. 19, no. 2, pp. 423–431, 2011.

[13] M. Franceschelli, D. Rosa, C. Seatzu, and F. Bullo, "Gossip algorithms for heterogeneous multi-vehicle routing problems," *Nonlinear Analysis: Hybrid Systems*, vol. 10, pp. 156–174, 2013.

[14] A. Testa, A. Rucco, and G. Notarstefano, "A finite-time cutting plane algorithm for distributed mixed integer linear programming," in *56th IEEE Conf. on Decision and Control (CDC)*, 2017, pp. 3847–3852.

[15] 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.

[16] 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.

[17] A. Falsone, "Distributed decision making with application to energy systems," *Ph.D. dissertation, Politecnico di Milano*, 2018.

[18] A. M. Geoffrion, "Lagrangean relaxation for integer programming," in *Mathematical programming study*. Springer, 1974, vol. 2, pp. 82–114.

[19] R. Vujanic, P. M. Esfahani, P. Goulart, and M. Morari, "Large scale mixed-integer optimization: A solution method with supply chain applications," in *IEEE 22nd Mediterranean Conf. of Control and Automation (MED)*, 2014, pp. 804–809.

[20] 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.

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

[22] F. Vanderbeck, "Implementing mixed integer column generation," in *Column generation*. Springer, 2005, pp. 331–358.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] 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.

[25] M. J. Todd, "Probabilistic models for linear programming," *Mathematics of Operations Research*, vol. 16, no. 4, pp. 671–693, 1991.