Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

A finite-Time cutting plane algorithm for distributed mixed integer linear programming

(Article begins on next page)

16 April 2024

A Finite-Time Cutting Plane Algorithm for
Distributed Mixed Integer Linear Programming

Andrea Testa, Alessandro Rucco, Giuseppe Notarstefano

Abstract— **Many problems of interest for cyber-physical network systems can be formulated as Mixed Integer Linear Programs in which the constraints are distributed among the agents. In this paper we propose a distributed algorithm to solve this class of optimization problems in a peer-to-peer network with no coordinator and with limited computation and communication capabilities. In the proposed algorithm, at each communication round, agents solve locally a small LP, generate suitable cutting planes, namely intersection cuts and cost-based cuts, and communicate a fixed number of active constraints, i.e., a candidate optimal basis. We prove that, if the cost is integer, the algorithm converges to the lexicographically minimal optimal solution in a finite number of communication rounds. Finally, through numerical computations, we analyze the algorithm convergence as a function of the network size.**

I. INTRODUCTION

Mixed Integer Linear Programming (MILP) plays an important role in many problems in control, including control of hybrid systems [1], trajectory planning [2], and task assignment [3]. For example, thanks to the simultaneous presence of equality/inequality constraints depending on some integer variables, nonlinear optimal control problems can be approximated by means of MILP. In this paper, we consider a distributed optimization setup in which the constraints of the MILP are distributed among agents of a network, and propose a distributed algorithm to solve it.

We organize the relevant literature to our paper in two main blocks: centralized and parallel approaches to solve MILP problems in control applications, and distributed algorithms solving linear programs and convex problems that can be seen as relaxations or special versions of mixed integer programs. First, a centralized Model Predictive Control (MPC) scheme for solving constrained multivariable control problems is proposed in [4] and [5]. The MPC is formulated as a multi-parametric MILP which avoids solving (expensive) MILPs on-line. In [6] the authors propose an algorithm to solve parametric mixed integer quadratic and linear programs. The algorithm uses a branch-and-bound procedure, where relaxations are solved in the nodes of a binary search tree. In [7] the authors show how to formulate a centralized trajectory optimization problem for multiple UAVs to a finite dimensional MILP which is solved by using a commercial branch and bound algorithm. In [8] the authors address the multi-robot routing problem under connectivity

constraints. The authors show that such a routing problem can be formulated as an integer program with binary variables, and then solve its LP relaxation.

As for parallel methods, in [9] a Lagrange relaxation approach is used in order to solve the overall MILP through a master-subproblem architecture. The proposed solution is applied to the demand response control problem in smart grids. Although processors are spatially distributed, the computation is parallel since it makes use of a central coordinator. In [10] a dual decomposition technique is proposed for the charging control problem of electric vehicles. Here an aggregator is required in order to assign charging slots to each individual electric vehicle.

Second, regarding distributed optimization algorithms, we concentrate on schemes solving linear programs and convex programs that represent a relaxation of suitable mixed-integer programs. In [11] the authors design a robust, distributed algorithm to solve linear programs over networks with event-triggered communication. Based on state-based rules, the agents decide when to broadcast state information to their neighbors in order to ensure asymptotic convergence to a solution of the linear program. In [12] the authors propose a distributed algorithm to find valid solutions for the bargaining problem by means of a LP relaxation. In [13] the authors address the Utility Maximization problem which is, in its general formulation, a mixed-integer nonlinear programming problem. The proposed solution is based on a convex relaxation, i.e., the integer constraint on the rates is relaxed thus yielding a convex program. In [14] and [15] the authors propose a Newton-type fast converging algorithm to solve, under the assumption that the utility functions are self-concordant, the Network Utility Maximization problem. In [16] the authors propose constraints consensus algorithms to solve abstract optimization programs (i.e., a generalization of linear programs). A distributed simplex algorithm is proposed in [17] to solve degenerate LPs and multi-agent assignment problems in asynchronous networks. A distributed version of the Hungarian method is proposed in [18] to solve distributed LP arising in multi-robot assignment problems. In [19] the authors address multi-agent task assignment and routing problems, modeled as MILP, in a distributed fashion. A gossip algorithm exploiting pairwise task exchanges between agents is proposed to find a common feasible assignment. Finally, a distributed trajectory optimization algorithm for cooperative UAVs is proposed in [20]. The algorithm is based on a special sequential computation set-up in which local MILPs are solved by the UAVs in a given sequence.

Andrea Testa, Alessandro Rucco, Giuseppe Notarstefano are with Department of Engineering, Università del Salento, Lecce, Italy {name.lastname}@unisalento.it

The main contribution of this paper is the design of a *Distributed MILP* algorithm, called `DiMILP`, that, solves MILP problems in finite time under the assumption of integer optimal cost. To the best of our knowledge, this is the first distributed algorithm solving MILP problems in asynchronous, directed networks. The algorithm is based on the local generation of suitable cutting planes, namely intersection cuts and cost-based cuts, and the exchange of active (basic) constraints. We consider a peer-to-peer network with no coordinator, in which each agent performs simple computations, i.e., solves small LPs and generates cutting planes, and communicates with other agents only a small number of linear constraints, i.e., a basis of its local LP relaxation. By exploiting the structure of intersection cuts and cost-based cuts, we prove the correctness of the proposed algorithm and its convergence in a finite number of communication rounds. We analyze and discuss a set of simulations to study the evolution and the convergence of the algorithm while varying the network size.

We highlight some meaningful differences with respect to the literature discussed above. Although constraint exchange and cutting plane approaches have been proposed in [16] and [21], in this paper we consider a MILP optimization problem. The presence of variables subject to integer constraints gives raise to new challenges in the algorithm design and analysis. In [19] and [20], the solution to each local MILP is optimal, yet there is no guarantee on the global optimality. On the contrary, we propose an algorithm for distributed MILP with guaranteed finite-time convergence to a global optimum.

The paper is organized as follows. In Section II, we introduce the MILP problem and its distributed formulation. In Section III, we describe the cutting plane approach for MILP. The distributed algorithm is introduced and analyzed in Section IV. Numerical computations are provided in Section V followed by the conclusion in Section VI.

II. PROBLEM SET-UP

We consider the following MILP:

$$\min_{z} c^\top z$$
$$\text{subj. to } a_i^\top z \le b_i \,, \, i = 1, \ldots, n \qquad (1)$$
$$z \in \mathbb{Z}^{d_Z} \times \mathbb{R}^{d_R}$$

where $a_i \in \mathbb{R}^d$, $b_i \in \mathbb{R}$, $c \in \mathbb{R}^d$, and n is the number of inequality constraints. Before formulating the distributed optimization set-up considered in the paper, we provide some useful notation.

Notation: We denote by z the decision variables in $\mathbb{Z}^{d_Z} \times \mathbb{R}^{d_R}$, by x the variables in $\mathbb{Z}^{d_Z}$, i.e., the ones subject to integer constraints, and by y the variables in $\mathbb{R}^{d_R}$. We let $d = d_Z + d_R$. Given an inequality $a^\top z \le b$ for $z \in \mathbb{R}^d$, with $a \in \mathbb{R}^d$ and $b \in \mathbb{R}$, we use the following simplified notation $\{a^\top z \le b\} := \{z \in \mathbb{R}^d : a^\top z \le b\}$ for the related half-space. The polyhedron[1] induced by the inequality constraints $a_i^\top z \le b_i \,, \, i \in \{1, \ldots, n\}$, is $P := \bigcap_{i=1}^{n} \{a_i z \le b_i\}$. Given

---

[1]A polyhedron is a set described by the intersection of a finite number of half-spaces.

two vectors $v, w \in \mathbb{R}^d$, v is lexicographically greater than w, $v >_{lex} w$, if there exists $k \in \{1, \ldots, d\}$ such that $v_k > w_k$ and $v_m = w_m$ for all $m < k$.

In this paper we assume an LP solver is available. In particular, we consider a solver that is able to find the unique *lexicographically minimal optimal solution* of the problem. From now on, we call such a solver LPLEXSOLV and say that it returns the *lex-optimal* solution meaning it is the lexicographically minimal optimal solution of the solved LP problem. LPLEXSOLV also returns an optimal *basis* identifying the lex-optimal solution. Given an LP problem with constraint set $P := \bigcap_{i=1}^{n} P_i$, with each P_i a half-space, a basis B is the intersection of a *minimal* number of half-spaces $P_{\ell_1}, \ldots, P_{\ell_k}$, $k \le d$, such that the solution of the LP over the constraint set B is the same as the one over P. If the lex-optimal solution is considered, it turns out that B is the intersection of exactly d half-spaces ($k = d$).

In our distributed setup, we consider a network composed by a set of agents $V = \{1, \ldots, N_{ag}\}$. In general, the $n \ge N_{ag}$ constraints in Problem 1 are distributed among the agents, so that each agent knows only a small number of constraints. For simplicity, we assume one constraint $\{a_i^\top z \le b_i\}$ is assigned to the i-th agent, so that $N_{ag} = n$, but we will keep the two notations separate to show that the algorithm can be easily implemented also when $N_{ag} < n$. The communication among the agents is modeled by a time-varying digraph $\mathcal{G}_c(t) = (V, E(t))$, with $t \in \mathbb{N}$ being a universal slotted time. A digraph $\mathcal{G}_c(t)$ models the communication in the sense that there is an edge $(i, j) \in E(t)$ if and only if agent i is able to send information to agent j at time t. For each node i, the set of *in-neighbors* of i at time t is denoted by $N_i(t)$ and is the set of j such that there exists an edge $(j, i) \in E(t)$. A static digraph is said to be *strongly connected* if there exist a directed path for each pair of agents i and j. For a time-varying digraph, we require the *joint strong connectivity*, i.e., $\forall t \in \mathbb{N}, \cup_{\tau=t}^{\infty} \mathcal{G}_c(\tau)$ is strongly connected.

III. CUTTING PLANES IN MIXED INTEGER LINEAR PROGRAMMING

In this section we provide a brief description of one of the most used methods to solve a centralized MILP, i.e., the cutting plane approach.

A. Cutting-Plane approach for MILP

Let $S := P \cap (\mathbb{Z}^{d_Z} \times \mathbb{R}^{d_R})$, problem (1) is equivalent, [22], to the following LP

$$\min_{z} c^\top z$$
$$\text{subj. to } z \in \text{conv}(S) \qquad (2)$$

where $\text{conv}(S)$ is the *convex hull* of S. A two dimensional example is shown in Figure 1. It is worth noting that, if P is a bounded polyhedron, by Meyer's Theorem, [23], $\text{conv}(S)$ is a polyhedron (i.e., $\text{conv}(S)$ is the solution set of a finite system of linear inequalities). For this reason, we make the following assumption, which is common in MILP literature.
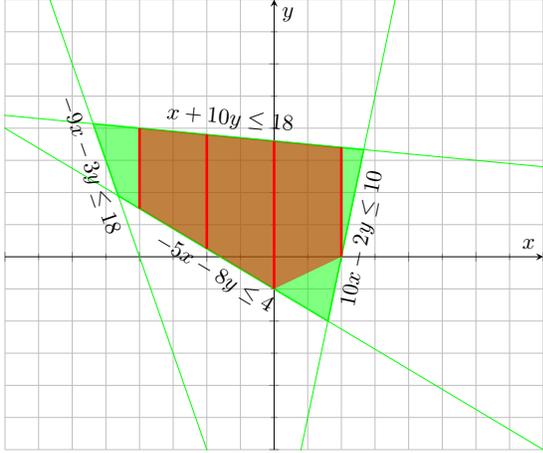
Fig. 1. Mixed integer set for a MILP with $x \in \mathbb{Z}$, $y \in \mathbb{R}$. The polyhedron P (green area) induced by the four inequality constraints (green lines), the set S of feasible solutions of the MILP (red lines), the convex hull of S (red area) are shown.

Assumption 3.1 (Boundedness of P): The polyhedron P is bounded and therefore $\mathrm{conv}(S)$ is a polyhedron. □

The main idea of the cutting plane approach is to neglect the integer constraints on the decision variables x, solve the *relaxed* linear problem (i.e., with $x \in \mathbb{R}^{d_Z}$) and properly tighten (in an iterative manner) the polyhedron P until the solution is in $\mathrm{conv}(S)$. The cutting plane procedure for MILPs can be summarized as follows. Relax the integer constraint in the formulation (1) and solve the optimization problem by using LPLEXSOLV. Let z_{LP} be the lex-optimal solution. If $z_{LP} \in \mathrm{conv}(S)$, then z_{LP} is the lex-optimal solution of (2) and, therefore, of (1). If $z_{LP} \notin \mathrm{conv}(S)$, find a linear inequality (called *Cutting Plane*)

$$\alpha^\top z \leq \alpha_0, \tag{3}$$

where $\alpha \in \mathbb{R}^d$ and $\alpha_0 \in \mathbb{R}$, which is satisfied by all $z \in \mathrm{conv}(S)$ and excluding z_{LP}. Then, update P with the new linear inequality (3) and repeat this approach until the lex-optimal feasible solution for the MILP (1) has been found.

Regarding the convergence property of this approach, under suitable assumptions, cutting plane algorithms obtain convergence after a finite number of iterations. This is the case, for example, of Integer Linear Programs (ILPs) when all the matrices are rational, [23], and of mixed binary programs [24]. For MILPs, if the optimal objective function value is integer, the first cutting plane algorithm converging in a finite number of iterations has been proposed in [25]. These considerations justify the following assumption.

Assumption 3.2 (Feasibility and integer optimal cost): Let $J(z)$ be the objective function, i.e., $J(z) := c^\top z$. There exists a lexicographically-minimal optimal solution $z^\star \in \mathrm{conv}(S)$ such that $J(z^\star) \leq J(z)$, $\forall z \in \mathrm{conv}(S)$. The optimal cost $J^\star := J(z^\star)$ is integer-valued. □

B. Cutting-Plane via Intersection Cuts

Many approaches have been developed to generate valid cutting planes, see [23] for a survey. Next we introduce the

notions of split disjunction, and intersection cut, that will be used to characterize the cuts we use in this paper, i.e., *Mixed Integer Gomory* (MIG) cuts.

Definition 3.3 (Split Disjunction [24]): Given $\pi \in \mathbb{Z}^{d_Z}$ and $\pi_0 \in \mathbb{Z}$, a split disjunction $D(\pi, \pi_0)$ is a set of the form $D(\pi, \pi_0) := \{\pi^\top x \leq \pi_0\} \cup \{\pi^\top x \geq \pi_0 + 1\}$. □
Let B_{LP} be a basis of the lex-optimal solution $z_{LP} = (x_{LP}, y_{LP})$, for a given LP relaxation of (1), and $D(\pi, \pi_0)$ a disjunction with respect to x_{LP}. Let $C(z_{LP})$ be the translated (simplicial) cone formed by the intersection of the halfplanes defining B_{LP} and having apex in z_{LP}[2]. An *intersection cut*, can be derived by considering the intersections between the extreme rays of $C(z_{LP})$ and the hyperplanes defining $D(\pi, \pi_0)$. A more detailed definition can be found in [26].

In this paper we use Mixed-Integer Gomory (MIG) cuts, [25], as cutting planes for our distributed algorithm. As shown in Appendix, MIG cuts can be obtained by working on the tableau of a problem in the form (A.6) equivalent to a given LP relaxation of (1). Specifically, the MIG cut with respect to the k-th row of the tableau, expressed in terms of the decision variable z, is given by:

$$h_{\mathrm{MIG}} := \left\{ \sum_{\ell \in N_+} \bar{a}_{k\ell}[b_B - A_B z]_\ell - \bar{f}_0 \sum_{\ell \in N_-} \bar{a}_{k\ell}[b_B - A_B z]_\ell \geq f_0 \right\}, \tag{4}$$

where A_B and b_B define the constraints of the basis B_{LP} (as in equation (A.5) in Appendix) and $\bar{a}_{k\ell} = [A_B^{-1}]_{k\ell}$, $f_0 = [A_B^{-1}b_B - \lfloor A_B^{-1}b_B \rfloor]_k$, $\bar{f}_0 = \frac{f_0}{1-f_0}$, $N_+ := \{\ell : \bar{a}_{k\ell} \geq 0\}$ and $N_- := \{\ell : \bar{a}_{k\ell} < 0\}$. Here we use the notation $[\cdot]_{k\ell}$ to indicate the (k, ℓ)-th element of a matrix and $[\cdot]_k$ to indicate the k-th element of the vector inside the brackets. Details on how to generate such a MIG cut are given in Appendix.

As recalled in Theorem A.2 in Appendix, a MIG cut with respect to the k-th row (of problem (A.6)) is an intersection cut to the split disjunction $D(e_k, \lfloor x_{LP_k} \rfloor)$ and the basis B_{LP}, with e_k being the k-th vector of the canonical basis (e.g., $e_1 = [1\, 0 \ldots 0]$) and x_{LP_k} the k-th component of x_{LP}. Let us consider now the first non-integer component of z_{LP}, namely $x_{LP_{k^{lex}}}$ where $k^{lex} = \arg\min\{k = 1, \ldots, d_Z : x_{LP_k} \notin \mathbb{Z}\}$. We call MIGORACLE the oracle that generates MIG cut (4) for $k = k^{lex}$.

In addition to the MIG cut, we also consider a constraint based on the actual cost function value $h_c := \{c^\top z \geq \lceil c^\top z_{LP} \rceil\}$. Notice that, by Assumption 3.2, h_c does not cut off any solution of $\mathrm{conv}(S)$. We refer to this inequality constraint as *cost-based cut*.

Next, we recall a centralized algorithm based on MIG and cost-based cuts, which is a reformulation of Gomory's cutting plane algorithm, [25], for MILPs in the form (1). This version is presented, e.g., in [27]. A pseudocode description of this algorithm is given in the table below, Algorithm 1.

It is worth noting that, at each iteration, Algorithm 1 uses the entire set of inequality constraints, P, and all the cuts generated up to that iteration.

---

[2]Given a cone $S \subset \mathbb{R}^d$ and a point $p \in \mathbb{R}^d$, the set $p + S$ is a translated cone with apex in p.

Algorithm 1 Cutting Plane Algorithm for MILP ([25])

Input P, c

(z_{LP}, B_{LP}) =LPLEXSOLV(P, c)

$h_c = \{c^\top z \geq \lceil c^\top z_{LP} \rceil\}$

while $z_{LP} \notin \mathrm{conv}(S)$ *do*

    $h_{\mathrm{MIG}} = \mathrm{MIGORACLE}(z_{LP}, B_{LP})$

    $P = P \cap h_{\mathrm{MIG}} \cap h_c$

    (z_{LP}, B_{LP}) =LPLEXSOLV(P, c)

    $h_c = \{c^\top z \geq \lceil c^\top z_{LP} \rceil\}$

Output $z^\star = z_{LP}$

IV. DISTRIBUTED MILP

In this section we propose a Distributed MILP algorithm, called DiMILP, based on the local generation of cutting planes and the exchange of active constraints. Then we prove its convergence in a finite number of communication rounds under the assumption that the optimal cost is integer.

In contrast to the centralized approach, in the distributed setup, at the first iterations, some agents may not have enough information to properly execute the algorithm (e.g., only one constraint has been assigned to the agent and the local MILP problem is unbounded). For this reason, we initialize the algorithm by assigning to each agent a set of artificial constraints which are inactive for the global problem (1). This method is often referred to as big-M method. Specifically, the decision variable of each agent is delimited by a box constraint. In particular, for a given, sufficiently large $M > 0$, we define the bounding box

$$H_M := \bigcap_{k=1}^{d} (\{z_k \leq M\} \cap \{z_k \geq -M\}).$$

A. Algorithm description

We now describe the proposed distributed algorithm.

Besides the initial constraint $h^{[i]} := \{a_i z \leq b_i\} \cap H_M$, each agent i has two local states, namely $z^{[i]}$, associated to the decision variable of MILP (1), and $B^{[i]}$ being a candidate basis of the problem. We use the subscript k to denote the k-th component of the local state, i.e., $z_k^{[i]}$.

At the generic time t, the i-th agent solves a linear program in which the common objective function $c^\top z$ is minimized subject to the following constraints: the intersection of its neighbors' candidate bases, $\bigcap_{j \in N_i} B^{[j]}(t)$, its own candidate basis, $B^{[i]}(t)$, the inequality constraint set at the initialization step, $h^{[i]}$, and the cost-based cut h_c obtained by rounding up the current cost. Then, agent i generates a MIG cut based on the current (local) lex-optimal solution. Finally, through a pivoting routine, named PIVOT, the agent updates its basis and thus the corresponding solution.

Summing up, at each communication round, each agent has to generate a constraint based on the local optimal cost value, share the basis with its neighbors through local communication (according to the communication graph), solve the LP problem, generate a MIG cut, and update its state. This procedure is formalized in Algorithm 2, where we dropped the dependence on t to highlight that agent i does not need to know it to perform the update.

Algorithm 2 DiMILP

State

$(z^{[i]}, B^{[i]})$

Initialization

$h^{[i]} = \{a_i z \leq b_i\} \cap H_M$

$(z^{[i]}, B^{[i]}) = \mathrm{LPLEXSOLV}(h^{[i]}, c)$

Evolution

$h_c = \{c^\top z \geq \lceil c^\top z^{[i]} \rceil\}$

$H_{tmp} = \left(\bigcap_{j \in N_i} B^{[j]}\right) \cap B^{[i]} \cap h^{[i]} \cap h_c$

$(z_{LP}, B_{LP}) = \mathrm{LPLEXSOLV}(H_{tmp}, c)$

$h_{\mathrm{MIG}} = \mathrm{MIGORACLE}(z_{LP}, B_{LP})$

$(z^{[i]}, B^{[i]}) = \mathrm{PIVOT}(B_{LP} \cap h_{\mathrm{MIG}}, c)$

We highlight that the proposed distributed algorithm is scalable in terms of local memory, computation and communication. Indeed, an agent sends to neighbors a candidate basis, which is a collection of d linear constraints. Consistently, it receives a number of bases equal to its in-degree. Also, in the computation it considers only two more inequality constraints at each iteration (i.e., the MIG cut, h_{MIG}, and one constraint, h_c, based on the local optimal cost-value).

B. Algorithm Analysis

The finite-time convergence and the correctness of the algorithm can be proven in three steps. For the sake of space, the proofs are omitted in this paper and will be provided in a forthcoming document. First, we show that for each agent the local cost and the local state converge in finite time. Second, we prove that consensus among all the agents is attained for the cost functions and for the candidate lex-optimal solutions. Finally, we show that the common candidate solution is indeed the lex-optimal solution of the global problem.

From now on, we denote $J^{[i]}(t)$ the local cost function value related to the decision variable of agent i, i.e., $J^{[i]}(t) = c^\top z^{[i]}(t)$.

Lemma 4.1 (Local convergence): Let $z^{[i]}(t)$ be the local candidate lex-optimal solution and $J(z^{[i]}(t))$ the associated cost of agent i at time $t \geq 0$ executing DiMILP. Then, in a finite number of communication rounds:

    i) the sequence $\{J(z^{[i]}(t))\}_{t \geq 0}$ converges to a constant value $\bar{J}^{[i]}$, and

    ii) the sequence $\{z^{[i]}(t)\}_{t \geq 0}$ converges to $\bar{z}^{[i]} = (\bar{x}^{[i]}, \bar{y}^{[i]})$, where $\bar{x}^{[i]} \in \mathbb{Z}^{d_z}$. $\square$

Lemma 4.2 (Consensus): Assume the communication network, $\mathcal{G}_c(t)$, is jointly strongly connected. Then, $\bar{J}^{[i]} = \bar{J}^{[j]}$ and $\bar{z}^{[i]} = \bar{z}^{[j]}$ for all $i, j \in \{1, \cdots, N_{ag}\}$. $\square$

We are now ready to present the main result of the paper.

Theorem 4.3 (DiMILP convergence): Consider MILP problem (1) in which the constraints are distributed among agents communicating according to a jointly strongly connected communication graph, $\mathcal{G}_c(t)$, $t \geq 0$. Let Assumption 3.2 hold and $M > 0$ be sufficiently large such that the lex-optimal solution of (1) does not change if the constraint set $\bigcap_{i=1}^{n} h^{[i]}$ is replaced by $\left(\bigcap_{i=1}^{n} h^{[i]}\right) \bigcap H_M$.
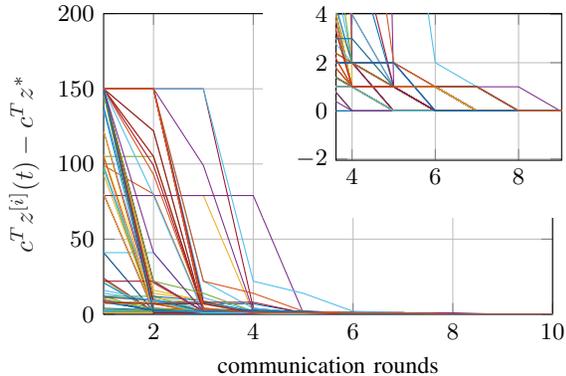
Fig. 2. Difference between the optimal cost and the cost evaluated by each agents at each communication round. The agents reach the optimal cost after 9 communication rounds.
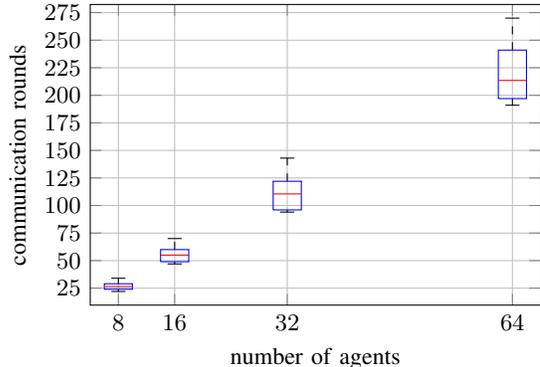

Fig. 3. Communication rounds evolution while varying the graph diameter. Box plot shows the minimum and maximum communication rounds (whiskers), 25% and 75% percentiles (lower and upper limit of the box) and median (red line).

Then `DiMILP` solves MILP problem (1) in a finite number of communication rounds. That is, the sequences $\{J(z^{[i]}(t))\}_{t \geq 0}$ and $\{z^{[i]}(t)\}_{t \geq 0}$, $i = \{1, \ldots N_{ag}\}$, converge, respectively, to the global optimal cost and to the lex-optimal solution of problem (1). $\square$

V. NUMERICAL COMPUTATIONS

In this section we provide numerical computations showing the effectiveness of the proposed algorithm.

We randomly generate the MILP data as follows. We consider a two-dimensional problem, $d = 2$. The decision variable is $z = (x, y)$ where $x \in \mathbb{Z}$ and $y \in \mathbb{R}$. We consider $N_{ag} = 100$ and $n = 100$, i.e., each agent only knows one constraint of the centralized MILP. The inequality constraints are randomly generated from the standard Gaussian distribution (we check feasibility and discard infeasible instances). The cost function is $c = [1, \ 0]^{\top}$. We consider an Erdős-Rényi static digraph with parameter 0.015. We run the algorithm by setting the bounding box H_M with $M = 150$. In Figure 2 we show the difference between the optimal costs of each agent and global optimal cost found by solving the centralized MILP (we use the "intlinprog" function in MATLAB). We obtain convergence to the global optimal solution and the corresponding optimal cost after 9 communication rounds, see zoom-in in Figure 2.

Next, we perform a numerical Monte Carlo analysis of the algorithm convergence while varying the network size and its diameter. We recall that, in a static digraph, the diameter is the maximum distance taken all over the pair of agents (i, j), where the distance is defined as the length of the shortest directed path from i to j. In the following we denote the diameter by $d_{\mathcal{G}}$. We choose a cyclic digraph for which the diameter is proportional to the number of agents, specifically $d_{\mathcal{G}} = N_{ag} - 1$. In particular, we consider the following cases: number of agents equal to 8, 16, 32 and 64. For each case, we generate 50 random MILPs ensuring that each test case has a non-empty set of feasible solutions. The results are shown in Figure 3. The red center line of each box shows the median value of the communication rounds for the 50 random MILPs with fixed diameter. We highlight

that the number of communication rounds needed for the convergence grows linearly with the graph diameter.

VI. CONCLUSION

In this paper, we proposed a distributed algorithm to solve Mixed Integer Linear Programs over peer-to-peer networks. In the proposed distributed setup, the constraints of the MILP are assigned to a network of agents. The agents have a limited amount of memory and computation capabilities and are able to communicate with neighboring agents. Following the idea of centralized cutting plane methods for MILP, each agent solves local (LP) relaxations of the global problem, generates cutting planes, and exchanges active constraints (a candidate basis) with neighbors. We proved that agents reach consensus on the lex-optimal solution of the MILP in a finite number of communication rounds. Finally, we performed a set of numerical computations suggesting that the completion time of the algorithm scales nicely with the number of agents.

APPENDIX
MIXED INTEGER GOMORY CUTTING PLANES

In order to derive a MIG cut, [25], for a generic LP relaxation of problem (1), let z_{LP} be the current lex-optimal solution and B_{LP} an associated basis. From the definition of basis, the lex-optimal solution can be obtained by solving the following LP problem

$$\min_{z} c^{\top} z$$
$$\text{subj. to } A_B z \leq b_B \tag{A.5}$$

where $A_B \in \mathbb{R}^{d \times d}$ and $b_B \in \mathbb{R}^d$ are the matrices obtained by writing in vector form the inequalities associated to the basis B_{LP}. We proceed by rewriting problem (A.5) in standard form. As described in [28], we i) reformulate (A.5) as a maximization problem, ii) replace z with two new decision variables $z_+ \in \mathbb{R}^d$ and $z_- \in \mathbb{R}^d$ having nonnegative components, such that $z = z_+ - z_-$, iii) introduce positive

slack variables $s \in \mathbb{R}^d$. In the new set of variables, we have

$$\max_u \bar{c}^\top u$$

$$\text{subj. to } \bar{A}u = b_B \qquad (A.6)$$

$$u_i \geq 0, k = 1, \ldots, 3d$$

where $u = [z_+^\top, z_-^\top, s^\top]^\top$, $\bar{A} = [A_B, -A_B, I_d]$, $\bar{c} = [-c^\top, c^\top, 0_d^\top]^\top$, I_d is the identity matrix of dimension $d \times d$, 0_d is d-dimensional zero vector. Let the matrix $\bar{A}$ be partitioned as $\bar{A} = [\bar{A}_B, \bar{A}_N]$ where $\bar{A}_B$ is a suitable[3] $d \times d$ non-singular submatrix of $\bar{A}$ and $\bar{A}_N$ consists of the remaining columns of $\bar{A}$. Define the corresponding partition of the vector $u = [u_B^\top, u_N^\top]^\top$ (basic and nonbasic variables). The basic solution corresponding to the basis matrix $\bar{A}_B$ is given by

$$u_B = \bar{b} - \bar{a}u_N , \qquad (A.7)$$

where $\bar{a} = \bar{A}_B^{-1}\bar{A}_N$ and $\bar{b} = \bar{A}_B^{-1}b_B$. The MIG cut is derived from the row of the simplex tableau (A.7) corresponding to a basic variable, let us say the k-th component of (A.7), that is required to be integer but it is not in the current solution. For the k-th row of the tableau, let us define $N_+ := \{\ell : \bar{a}_{k\ell} \geq 0\}$ and $N_- := \{\ell : \bar{a}_{k\ell} < 0\}$, where $\bar{a}_{k\ell}$ is the entry of the (k, ℓ)-th entry of matrix $\bar{a}$. Then the MIG cut is

$$\sum_{\substack{f_{k\ell} \leq f_0 \\ \ell \text{ integer}}} f_{k\ell}u_\ell + \frac{f_0}{1-f_0} \sum_{\substack{f_{k\ell} > f_0 \\ \ell \text{ integer}}} (1-f_{k\ell})u_\ell +$$
$$+ \sum_{\substack{\ell \in N_+ \\ \ell \text{ non-integer}}} \bar{a}_{k\ell}u_\ell - \frac{f_0}{1-f_0} \sum_{\substack{\ell \in N_- \\ \ell \text{ non-integer}}} \bar{a}_{k\ell}u_\ell \geq f_0, \qquad (A.8)$$

where $f_{k\ell}$ is the fractional part of $\bar{a}_{k\ell}$, and f_0 is the fractional part of the k-th component of $A_B^{-1}b_B$. We can rewrite the MIG cut (A.8) with respect to the original decision variables z (as in (4)) by taking in mind that $s = b_B - A_B(z_+ - z_-)$ and $z = z_+ - z_-$.

Theorem A.1 ([27]): The MIG cut (A.8) is a valid cutting plane for $S = \{P \cap \mathbb{Z}^{d_Z} \times \mathbb{R}^{d_R}\}$.

Theorem A.2 ([29]): Let $u^\star$ be a solution of (A.6) and B the associated basis. Let us suppose the k-th element of $u^\star$ is not integer. Then the intersection cut to the split disjunction $D(e_k, \lfloor u_k^\star \rfloor)$ and the basis B is equal to the MIG cut to the k-th component.

References

[1] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[2] A. Richards, T. Schouwenaars, J. P. How, and E. Feron, "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 755–764, 2002.

[3] J. Bellingham, M. Tillerson, A. Richards, and J. P. How, "Multi-task allocation and path planning for cooperating UAVs," in *Cooperative control: models, applications and algorithms.* Springer, 2003, pp. 23–41.

[4] A. Bemporad, F. Borrelli, and M. Morari, "Piecewise linear optimal controllers for hybrid systems," in *American Control Conference*. IEEE, 2000, pp. 1190–1194.

[5] A. Bemporad, F. Borrelli, M. Morari *et al.*, "Model predictive control based on linear programming - The explicit solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[6] D. Axehill, T. Besselmann, D. M. Raimondo, and M. Morari, "A parametric branch and bound approach to suboptimal explicit hybrid MPC," *Automatica*, vol. 50, no. 1, pp. 240–246, 2014.

[7] F. Borrelli, D. Subramanian, A. U. Raghunathan, and L. T. Biegler, "MILP and NLP techniques for centralized trajectory planning of multiple unmanned air vehicles," in *American Control Conference*. IEEE, 2006, pp. 5764–5769.

[8] S. Chopra and M. Egerstedt, "Spatio-temporal multi-robot routing," *Automatica*, vol. 60, pp. 173–181, 2015.

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

[10] 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.

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

[12] ——, "Distributed bargaining in dyadic-exchange networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 3, pp. 310–321, 2016.

[13] C. Fischione, M. D'Angelo, and M. Butussi, "Utility maximization via power and rate allocation with outage constraints in nakagami-lognormal channels," *IEEE Transactions on Wireless Communications*, vol. 10, no. 4, pp. 1108–1120, 2011.

[14] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed newton method for network utility maximization, part I: Algorithm," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2162–2175, 2013.

[15] ——, "A distributed newton method for network utility maximization, part II: Convergence," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2176–2188, 2013.

[16] 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.

[17] 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.

[18] S. Chopra, G. Notarstefano, M. Rice, and M. Egerstedt, "Distributed version of the hungarian method for a multirobot assignment," *IEEE Transactions on Robotics*, 2017.

[19] 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.

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

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

[22] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[23] M. Jünger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, *50 Years of integer programming 1958-2008: From the early years to the state-of-the-art*. Springer Science & Business Media, 2009.

[24] G. Cornuéjols, "Valid inequalities for mixed integer linear programs," *Mathematical Programming*, vol. 112, no. 1, pp. 3–44, 2008.

[25] R. E. Gomory, "An algorithm for the mixed integer problem." *RM-2597, The RAND Corporation*, 1960.

[26] E. Balas and F. Margot, "Generalized intersection cuts and a new cut generating paradigm," *Mathematical Programming*, pp. 1–17, 2013.

[27] M. Jörg, "k-disjunctive cuts and cutting plane algorithms for general mixed integer linear programs," Ph.D. dissertation, Universität München, 2008.

[28] D. G. Luenberger, Y. Ye *et al.*, *Linear and nonlinear programming*. Springer, 1984, vol. 2.

[29] E. Balas, G. Cornuéjols, T. Kis, and G. Nannicini, "Combining lift-and-project and reduce-and-split," *INFORMS Journal on Computing*, vol. 25, no. 3, pp. 475–487, 2013.

[3]If $z_{LP_k} \geq 0 \, \forall k = 1 \ldots d$, then $\bar{A}_B = A_B$. If $z_{LP_k} \leq 0 \, \forall k = 1 \ldots d$, then $\bar{A}_B = -A_B$. In the general case $\bar{A}_B$ is a suitable permutation of columns of A_B and $-A_B$.