Efficient Solution of Distributed MILP in Control of Networked Systems

Zonglin Liu [*] **Olaf Stursberg** [*]

[*] *Control and System Theory, Department of Electrical Engineering and Computer Science, University of Kassel, Germany. (Email: {Z.Liu,stursberg}@uni-kassel.de).*

Abstract: This paper considers the distributed solution of Mixed-Integer Linear Programming (MILP) problems, a class of problems which is of interest, e.g., in optimization-based control of networked systems involving hybrid dynamics. For a larger number of subsystems, the high combinatorial complexity arising from the integer variables usually prohibits the use of centralized solution schemes, and thus requires distributed computational approaches. The proposed approach is inspired by results based on the Shapley-Folkman-Starr theorem, but it relaxes some conservative assumptions in order to enhance the computational efficiency. Numerical experiments for different MILP problems confirm the advantage of the proposed method with respect to computation times.

Keywords: Optimization, distributed control, computational methods, optimal control, networked systems, hybrid systems, complexity, integer programming.

1. INTRODUCTION

In optimization-based control of networked systems, the system size often implies the use of distributed computational approaches. If, in addition, the problem formulation involves logic conditions, the problem class of distributed mixed-integer programming (MIP) arises naturally. Logic conditions may, e.g., stem from hybrid dynamics (possibly the assignment of different dynamics to different regions of the state space, see Groß and Stursberg (2013), or from the modeling of resource constraints involving integer variables, see Liu and Stursberg (2019a). Since MIP problems belong to the complexity class NP-hard, already the solution of optimal control problems for single hybrid systems require dedicated encoding schemes, see e.g. Liu and Stursberg (2018, 2019b). For control problems of networked systems comprising several interacting or coupled subsystems, this is even more true. Instead of centralized solutions for the complete network, the distributed solution with appropriate means to exchange information between the subproblems is a reasonable choice.

To motivate the optimization problem under study in this paper, imagine the following control problem: Let for a set of n_s subsystems the dynamics of any subsystem be specified as discrete-time piecewise affine system, where the assignment of the affine dynamics to polytopic regions can be achieved by mixed-integer formulations, as e.g. described in Bemporad and Morari (1999). Assume further that coupling constraints among the subsystems have been formulated for some continuous or integer variables formulating, e.g., that the system state of only a certain number of subsystems may be contained in a polytopic region at any time. If, in addition, the local cost functions of all subsystems are formulated in linear form, the global problem (containing all subproblems) can be written as MILP task according to:

$$\min_{x_1,\cdots,x_{n_s}} \sum_{i\in N} c_i^{\mathrm{T}} x_i \tag{1}$$

$$\text{s.t.:} \quad \sum_{i\in N} A_i x_i \leq b \tag{2}$$

$$x_i \in X_i \subset \mathbb{R}^{r_i} \times \mathbb{Z}^{z_i},\ \forall i \in N. \tag{3}$$

In here, the vector x_i contains all mixed-integer variables assigned to the subsystem with index $i \in N = \{1,\ldots,n_s\}$, i.e., it contains the continuous states and inputs as well as discrete states and inputs (and possible auxiliary variables) over the complete set of considered discrete points of time. Furthermore, r_i and z_i are the number of continuous and discrete components of x_i, while $c_i \in \mathbb{R}^{r_i+z_i}$, $A_i \in \mathbb{R}^{m\times(r_i+z_i)}$ and $b \in \mathbb{R}^{m\times 1}$ denote matrices of appropriate dimension. The mixed-integer polyhedral set:

$$X_i = \{x_i \in \mathbb{R}^{r_i} \times \mathbb{Z}^{z_i} | D_i x_i \leq d_i\}, \tag{4}$$

in (3) denotes **local** constraints of subsystem i, e.g., the equations and inequalities obtained by encoding the local hybrid dynamics. In contrast, the constraints (2) model the **coupling** among the subsystems – it is worth emphasizing these constraints can represent coupling of states, of inputs, or a mixture of both; The number m of these coupling constraints will play an important role in the further description. This uses $J := \sum_{i\in N} c_i^{\mathrm{T}} x_i$ to refer to the costs of problem (1), as well as $\mathbf{x}^* = [x_1^*;\cdots;x_{n_s}^*]$ to denote the optimal solution of (1), and $J(\mathbf{x}^*)$ for the optimal cost.

Strategies for solving the problem (1) in distributed form have been considered in several publications before, see e.g. Vujanic et al. (2014); Falsone et al. (2019); Camisa et al. (2018); Vujanic et al. (2016) and Camisa and Notarstefano (2019). Most of this work, which is briefly summarized in the following, is based on the Shapley-Folkman-Starr theorem, see Aubin and Ekeland (1976). The key aspect of this theorem is that the vector sum of a large number of non-convex sets tends to be convex, since any vector in its convex hull can be closely approximated by a vector in the non-convex set. As a consequence, the duality gap between the non-convex problem (1) and its dual problem turns out to be relatively small and can often be shown to diminish when the number of subsystems increases (see Chapter 5.7.1 in Bertsekas (2009) for more details). Thus when the mixed-integer set X_i in (1) is replaced by a real-valued set $Conv(X_i)$, denoting the convex hull of all points in X_i, then, the MILP problem (1) can be cast into a Linear Programming (LP) problem:

$$\min_{x_1,\cdots,x_{n_s}} \sum_{i\in N} c_i^{\mathrm{T}} x_i \tag{5}$$
$$\text{s.t.:} \sum_{i\in N} A_i x_i \le b$$
$$x_i \in Conv(X_i),\ x_i \in \mathbb{R}^{r_i+z_i},\ \forall i \in N,$$

by dropping the integrality constraint of all variables in x_i, i.e., $x_i \in \mathbb{R}^{r_i+z_i}$. It must be emphasized that $Conv(X_i)$ does not coincide with the set $\mathcal{R}(X_i) = \{x_i \in \mathbb{R}^{r_i+z_i} | D_i x_i \le d_i\}$, which is obtained by relaxing the integrality constraints to intervals in X_i – in fact, the former is always tighter, see Geoffrion (1974). Furthermore, it is well known that, if the LP problem (5) has a unique optimal solution, then it must be attained in a vertex of the feasible set. By using $\bar{\mathbf{x}}^* = [\bar{x}_1^*, \cdots, \bar{x}_{n_s}^*]$ to denote the unique optimal solution of (5), the following relation is proposed according to the Shapley-Folkman-Starr theorem in Vujanic et al. (2014):

Theorem 1. In $\bar{\mathbf{x}}^*$, a partitioning $N = N_1 \cup N_2$, $|N_1| \ge n_s - m - 1$, of the subsystems can be determined, such that the local solution $\bar{x}_i^*$ in $\bar{\mathbf{x}}^*$ is attained at the vertex of $Conv(X_i)$ for all $i \in N_1$.

Furthermore, as all vertices of $Conv(X_i)$ are also located in the mixed-integer set X_i, the local solution $\bar{x}_i^*$ is also in X_i, $\forall i \in N_1$, although subject to a real-valued set in (5). In case that a large number of subsystems is involved in (1), but only a few coupling constraints (2) need to be considered, i.e., $n_s \gg m$, Theorem 1 implies that the majority of the subsystems can already determine their local feasible solutions of MILP problem (1) through the solution of the LP problem (5).

Thus, the authors in Vujanic et al. (2014) introduced a method to obtain local feasibility of the remaining $|N_2|$ subsystems, while maintaining the satisfaction of the coupling constraints (2). Meanwhile, due to the convexity of problem (5), the dual decomposition of (5) (by dualizing the coupling constraints (2) with a multiplier $\lambda \in \mathbb{R}^{1\times m}$) has also been used to realize a distributed computation of (5) involving the sub-gradient method, see Anstreicher and Wolsey (2009). However, this method faces the following drawbacks, limiting its application:

(1) the condition $A_i x_i \ge 0$, $\forall x_i \in X_i$, $\forall i \in N$ is required in (1), in order to guarantee the success of the computation for the $|N_2|$ subsystems;
(2) the convergence rate towards $\bar{x}_i^*$ by employing the sub-gradient method is extremely slow (mainly due to the vanishing step size in each iteration).

Facing these drawbacks, a new relation between (1) and (5) was established in Vujanic et al. (2016):

Theorem 2. If (5) and its dual problem (obtained by dualizing the coupling constraints (2)) have a unique optimal solution $\bar{\mathbf{x}}^*$ and $\bar{\lambda}^*$, then the local solutions obtained by solving the following local problems
$$\min_{x_i\in X_i} (c_i^{\mathrm{T}} + \bar{\lambda}^* A_i) x_i \tag{6}$$
for all n_s subsystems, differ for at most $m+1$ subsystems with the optimal solution $\bar{\mathbf{x}}^*$ of problem (5).

This enables each subsystem i to solve problem (6) locally, based on the dual optimum $\bar{\lambda}^*$. The resulting local solution satisfies X_i for all $i \in N$, but may eventually violate the coupling constraint (2). Accordingly, the authors in Vujanic et al. (2016) tightened the constraints in (2) in advance, and determined the dual optimum of the substitute problem. Through this approach, the local solutions determined from (6) based on the new dual optimum, only violate the tightened coupling constraint, but not the original one – thus, they constitute a feasible candidate of (1). Here, the condition on $A_i x_i \ge 0$, $\forall x_i \in X_i$, $\forall i \in N$ is relaxed, but the following significant drawbacks still exist:

(1) the convergence rate towards the dual optimum $\bar{\lambda}^*$ when employing the sub-gradient method is as slow as the one towards $\bar{x}_i^*$;
(2) the problem (5) must have a feasible solution after tightening the coupling constraints (2), but the tightening scheme is quite conservative, since each coupling constraint with index j is tightened by:
$$(m+1)\cdot \max_{i\in N}\Big(\max_{x_i\in X_i} A_i(j,:)x_i - \min_{x_i\in X_i} A_i(j,:)x_i\Big), \tag{7}$$
resulting in a substitute constraint which is hard to satisfy.

Thereafter, Falsone et al. (2019) proposed an improved sub-gradient method, aiming at reducing the conservativeness caused by the tightening. In the work of Camisa et al. (2018) and Camisa and Notarstefano (2019), the authors employed the primal decomposition method instead of the dual one for the solution of (5), in order to provide a finite-time suboptimality bound.

Obviously, the common drawback of the work above is the slow convergence rate towards either $\bar{x}_i^*$, or $\bar{\lambda}^*$ when employing the sub-gradient method. In addition, requirements such as $A_i x_i \ge 0$, $\forall x_i \in X_i$, $\forall i \in N$, or the feasibility of (5) with tightened coupling constraints, further limit the applicability of these methods.

Thus, this paper proposes a novel distributed solution scheme aiming at overcoming these issues. The main idea of the proposed method is still based on the Shapley-Folkman-Starr theorem – but unlike the existing methods in which the theorem is employed to directly determine a feasible candidate of (1), here it is applied to iteratively improve feasible candidates for the solution of (1), until the global optimum (or at least a sub-optimum) is found. We show that the conservative assumptions mentioned above are relaxed in the proposed approach, and that the computations can be accelerated.

In the next section, the proposed distributed solution of (1) is introduced and discussed in detail. Then, the procedure is tested on various MILP problems in Sec. 3 in order to illustrate its efficiency. The paper concludes with a discussion and an outlook in Sec. 4.

2. THE DISTRIBUTED SOLUTION

The step of finding a first feasible candidate of an MIP problem usually constitutes the first phase of solution in most of the existing solvers, typically incurring a high cost function value. Only after a certain number of iterations, established search strategies, like branch-and-bound, or branch-and-cut algorithms, the quality of the candidate improves. Specifically for MILP problems according to (1), many tests and applications have shown that – even if the problem is large – the determination of the first feasible candidate is typically much faster than the process of converging to the optimal one. As an example, for the electric vehicle charging problem considered in Vujanic et al. (2016); Falsone et al. (2019), the determination of the globally optimal solution needs over 6 hours, but the first feasible candidate was found in less than one second.

Thus, by assuming that a first feasible candidate of (1) is on hand and denoted by $\mathbf{x}^f = [x_1^f; \cdots; x_{n_s}^f]$, the main task of the algorithm to be proposed for distributed solution is to improve $\mathbf{x}^f$ until the global optimum (or a value close to it) is found. Recall that, when the standard branch-and-bound method is applied to improve $\mathbf{x}^f$, it usually keeps branching on the integer variables (leading to nodes). This process, however, requires a heuristics on the priority of the nodes to be considered first for branching. If the heuristics is not efficient, this causes a large amount of meaningless computation, e.g. when the new nodes are infeasible, or worse than $\mathbf{x}^f$. It also requires to store a large amount of data. Here instead, search directions for continuous cost improvements in **any** iteration are proposed.

2.1 Improving $\mathbf{x}^f$ by the Sub-Gradient Method

The method first starts with determining a new coupling constraint of (1) based on $\mathbf{x}^f$, and taking the form:

$$\sum_{i\in N} A_i x_i \le b^f, \text{ with } b^f = \sum_{i\in N} A_i x_i^f. \tag{8}$$

Since $\mathbf{x}^f$ is a feasible candidate of (1), the vector b^f must satisfy:

$$b^f \le b. \tag{9}$$

Obviously, this implies that (8) is a tighter constraint compared to the original one. Now, by replacing (2) with (8) in problem (1), the following MILP problem is obtained:

$$\min_{x_1,\cdots,x_{n_s}} \sum_{i\in N} c_i^{\mathrm{T}} x_i \tag{10}$$

$$\text{s.t.: } \sum_{i\in N} A_i x_i \le b^f$$

$$x_i \in X_i,\, x_i \in \mathbb{R}^{r_i} \times \mathbb{Z}^{z_i},\, \forall i \in N.$$

A similar MILP problem with tightened coupling constraint was also used in Vujanic et al. (2016), but there, the existence of a feasible solution of the tightened problem was only assumed, while the feasibility of (10) here always holds, since $\mathbf{x}^f$ is feasible. Now, by replacing the mixed-integer set X_i in (10) with the convexified set $Conv(X_i)$, an LP problem is obtained:

$$\min_{x_1,\cdots,x_{n_s}} \sum_{i\in N} c_i^{\mathrm{T}} x_i \tag{11}$$

$$\text{s.t.: } \sum_{i\in N} A_i x_i \le b^f$$

$$x_i \in Conv(X_i),\, x_i \in \mathbb{R}^{r_i+z_i},\, \forall i \in N.$$

Its dual problem (obtained from dualizing the coupling constraint (8)), has the form:

$$\sup_{\lambda} -\lambda b^f + \sum_{i\in N} \min_{x_i \in Conv(X_i)} (c_i^{\mathrm{T}} + \lambda A_i) x_i \tag{12}$$

$$\text{s.t.: } \lambda \ge 0.$$

Here, $\bar{\mathbf{x}}^{*,f} = [\bar{x}_1^{*,f}; \cdots; \bar{x}_{n_s}^{*,f}]$ and $\bar{\lambda}^{*,f}$ denote the optimal solution of (11) and (12).

Assumption 1. Both problems (11) and (12) have unique optimal solutions $\bar{\mathbf{x}}^{*,f}$ and $\bar{\lambda}^{*,f}$.

Note that this assumption is typically not conservative, since even for degenerate cases, in which this assumption may be violated, one can avoid this situation by introducing small perturbations to the cost or constraints in (11), as indicated in Vujanic et al. (2016).

For the series of MILP and LP problems introduced in the last two sections, let their optimal costs be compared: By using $J(\mathbf{x}^*)$, $J(\bar{\mathbf{x}}^*)$, $J(\mathbf{x}^{*,f})$, and $J(\bar{\mathbf{x}}^{*,f})$ to represent the optimal costs of the problems (1), (5), (10), and (11) respectively, as well as $J(\mathbf{x}^f)$ for the global costs of $\mathbf{x}^f$, the following relations can be established:

Theorem 3. For a given $\mathbf{x}^f$, it applies that:

$$J(\bar{\mathbf{x}}^*) \le J(\mathbf{x}^*) \le J(\mathbf{x}^{*,f}) \le J(\mathbf{x}^f), \tag{13}$$

$$J(\bar{\mathbf{x}}^*) \le J(\bar{\mathbf{x}}^{*,f}) \le J(\mathbf{x}^{*,f}). \tag{14}$$

Proof. The first inequality (from left to right) in (13) follows from the relaxed integrality constraint in (5) compared to (1); the second inequality in (13) follows from the tighter coupling constraint in (10) compared to (1); the last inequality in (13) follows from the fact that $\mathbf{x}^f$ represents a feasible candidate of (10) only, but not necessarily the optimal one; the first inequality in (14) is implied by the tighter coupling constraint in (11) compared to (5); the second inequality in (14) results from the relaxed integrality constraint in (11) as opposed to (10). □

For the objective of improving $\mathbf{x}^f$ the relations listed in Theorem 3 point into useful direction: firstly, it can be noticed that $J(\mathbf{x}^*)$ and $J(\bar{\mathbf{x}}^{*,f})$ are both lower than $J(\mathbf{x}^f)$, and the two values are bounded by the same lower bound $J(\bar{\mathbf{x}}^*)$ and the upper bound $J(\mathbf{x}^{*,f})$. This indicates that $\mathbf{x}^*$ and $\bar{\mathbf{x}}^{*,f}$ are leading to similar global costs, which are lower than $J(\mathbf{x}^f)$. Then, since:

- $\mathbf{x}^*$ is the global optimum of (1), which defines the best improved candidate one can find for $\mathbf{x}^f$;
- and $\bar{\mathbf{x}}^{*,f}$ is the global optimum of the LP problem (11), which may not satisfy the local constraint X_i (and thus may not be feasible for (1)),

it is straightforward to assume that a feasible candidate of (1) being located close to $\bar{\mathbf{x}}^{*,f}$ will attain a similar global

Algorithm 1 Distributed computation of $\bar{\mathbf{x}}^{*,f}$

```
1: Initialization: $\rho = 1$, $\lambda^{[\rho]} = 0$, $\mathbf{x}^f$, $Flag = 0$;
2: while $\rho \leq \rho^{max}$ and $Flag = 0$ do
3:     for $i = 1 : n_s$ do
4:         $x_i^{[\lambda^{[\rho]}]} := \arg\min_{x_i \in X_i}(c_i^{\mathrm{T}} + \lambda^{[\rho]} A_i) x_i$
5:         $\overline{x}_i^{[\rho]} := \frac{1}{\rho} \sum_{j=1}^{\rho} x_i^{[\lambda^{[\rho]}]}$
6:     end for
7:     Decompose $x_i^{[\lambda^{[\rho]}]}$ into $[x_{i,r}^{[\lambda^{[\rho]}]}; x_{i,int}^{[\lambda^{[\rho]}]}]$, $\forall i \in N$, and solve the LP problem:
           $\min_{x_{1,r},\cdots,x_{n_s,r}} \sum_{i \in N} c_i^{\mathrm{T}} \cdot [x_{i,r}; \mathbf{0}^{z_i \times 1}]$   (15)
           s.t.: $\sum_{i \in N} A_{i,r} \cdot x_{i,r} \leq b - \sum_{i \in N} A_{i,int} \cdot x_{i,int}^{[\lambda^{[\rho]}]}$   (16)
           $D_{i,r} \cdot x_{i,r} \leq d_i - D_{i,int} \cdot x_{i,int}^{[\lambda^{[\rho]}]}, \forall i \in N$   (17)
           $x_{i,r} \in \mathbb{R}^{r_i}, \forall i \in N$
8:     if (15) - (17) is feasible and the optimized solution $x_{i,r}^*$, $i \in N$ satisfies $\sum_{i \in N} c_i^{\mathrm{T}}[x_{i,r}^*; x_{i,int}^{[\lambda^{[\rho]}]}] < J(\mathbf{x}^f)$ then
9:         $Flag = 1$
10:    else
11:        $Flag = 0$
12:    end if
13:    $\gamma^{[\rho]} := \sum_{i \in N} A_i x_i^{[\lambda^{[\rho]}]} - b^f$
14:    $\lambda^{[\rho+1]} := P_+(\lambda^{[\rho]} + s^{[\rho]} \gamma^{[\rho]})$
15:    $\rho := \rho + 1$
16: end while
```

cost as $J(\bar{\mathbf{x}}^{*,f})$ and $J(\mathbf{x}^*)$. Thus, such a feasible candidate can be regarded as better than $\mathbf{x}^f$, and for determining this candidate the knowledge of $\bar{\mathbf{x}}^{*,f}$ is required.

Hence, the dual decomposition and sub-gradient method are employed to compute $\bar{\mathbf{x}}^{*,f}$ in a distributed fashion, see Algorithm 1: If the lines 7 to 12 were omitted in Algorithm 1, this would represent a standard sub-gradient method, known to suffer from slow convergence rates. In the algorithm, ρ is the iteration counter and ρ^{max} represents the maximal number of iterations allowed to be executed (usually a large number to ensure the convergence to $\bar{\mathbf{x}}^{*,f}$), and $\gamma^{[\rho]}$ is the sub-gradient of the dual function in (12). The symbol $s^{[\rho]}$ is the step length chosen to update the multiplier $\lambda^{[\rho]}$ in each iteration ρ, and the operation P_+ in line 14 denotes the projection onto the positive sub-space of $\mathbb{R}^m$. The computations in line 3 to 6 are carried out in parallel for the subsystems. Note that the index $[\lambda^{[\rho]}]$ in $x_i^{[\lambda^{[\rho]}]}$ is used to clarify that this solution to the sub-problem in line 4 is specific for the present value of $\lambda^{[\rho]}$. In addition, the computation in line 5 is an averaging over the $x_i^{[\lambda^{[\rho]}]}$ obtained in the previous iteration. It leads to $\bar{x}_i^{*,f}$ for $\rho \to \infty$, as long as the following conditions for the step length $s^{[\rho]}$ are satisfied (see Anstreicher and Wolsey (2009) for the proof):

$$s^{[\rho]} \to 0, \sum_{\rho=1}^{\infty} s^{[\rho]} = \infty, \sum_{\rho=1}^{\infty} (s^{[\rho]})^2 < \infty. \tag{18}$$

A simple choice of $s^{[\rho]}$ satisfying these conditions is $s^{[\rho]} = \frac{1}{\rho}$. Finally, it is emphasized that the constraint $x_i \in X_i$ in line 4 differs from the original constraint $x_i \in Conv(X_i)$ in (11) and (12), since the computation of the convexified set $Conv(X_i)$ is hard, especially for a large dimension. The authors in Vujanic et al. (2016) suggested to use column generation techniques, see Barnhart et al. (1998), to construct approximations of $Conv(X_i)$. Here instead, as the term to be minimized in line 4 is linear, i.e., must exist an optimal $x_i^{[\lambda^{[\rho]}]}$ located in the vertices of $Conv(X_i)$, thus also in X_i. Accordingly, the constraint $x_i \in X_i$ is adopted in line 4, since the outcome will not be affected according to Assumption 1. In other words, a small-scale MILP problem with local variables only is solved in line 4, instead of the original LP problem requiring the knowledge of $Conv(X_i)$.

Clearly, without the steps in line 7 to 12, the Algorithm 1 would terminate after averaging to $\bar{\mathbf{x}}^{*,f}$ in line 5. At this stage, an improvement of $\mathbf{x}^f$ can be determined based on $\bar{\mathbf{x}}^{*,f}$. However, since the convergence towards $\bar{\mathbf{x}}^{*,f}$ usually requires many iterations, the computations in line 7 to 12 are carried out in addition, to reduce the number of necessary iterations: Any set of local variables $x_i \in X_i$ can always be decomposed into the real-valued part $x_{i,r} \in \mathbb{R}^{r_i}$, and the integer part $x_{i,int} \in \mathbb{Z}^{z_i}$. Similarly, the matrices A_i and D_i can also be decomposed into $A_i = [A_{i,r}, A_{i,int}]$ and $D_i = [D_{i,r}, D_{i,int}]$, such that $A_i x_i = [A_{i,r}, A_{i,int}] \cdot [x_{i,r}; x_{i,int}]$ and $D_i x_i = [D_{i,r}, D_{i,int}] \cdot [x_{i,r}; x_{i,int}]$ hold. With this scheme for the $x_i^{[\lambda^{[\rho]}]}$ obtained in line 4 in iteration ρ, the newly assigned problem (15) in line 7 fixes the integer part $x_{i,int}^{[\lambda^{[\rho]}]}$, and leaves the real part $x_{i,r}^{[\lambda^{[\rho]}]}$ to be newly selected. This aims at achieving the following goals through the solution of (15):

- reducing the global costs attained by $x_i^{[\lambda^{[\rho]}]}$, $i \in N$ through the variation of their real-valued parts;
- as $x_i^{[\lambda^{[\rho]}]} \in X_i$ applies according to line 4, the constraint (17) aims at preserving the local feasibility during the variation of the real variables;
- by employing the sub-gradient method to solve problem (11), the $x_i^{[\lambda^{[\rho]}]}$ obtained in iteration ρ may, in general, violate the dualized coupling constraints (8) (see Vujanic et al. (2016) for the reasoning, and also the second plot in Fig. 1, which demonstrates the maximal violation to (8) in each iteration). However, as the original coupling constraints (2) determine a larger feasible space than (8) according to (9), the $x_i^{[\lambda^{[\rho]}]}$ may have satisfied (2) even if (8) is not satisfied (see the first plot in Fig. 1, recording the maximal violation of (2)). Thus, as the real-valued part of $x_i^{[\lambda^{[\rho]}]}$ is allowed to be newly selected in (15), the constraint (16) aims at ensuring that the feasibility of (2) is eventually recovered after the optimization.

Let $x_{i,r}^*$, $i \in N$ denote the optimized solution of (15). If problem (15) is feasible in iteration ρ, then a new feasible candidate $\mathbf{x}^{new} = [x_1^{new}; \cdots; x_{n_s}^{new}]$ of (1) is found with $x_i^{new} := [x_{i,r}^*; x_{i,int}^{[\lambda^{[\rho]}]}]$, $i \in N$. Then, the global costs of the new candidate $J(\mathbf{x}^{new})$ is checked and if it is smaller than

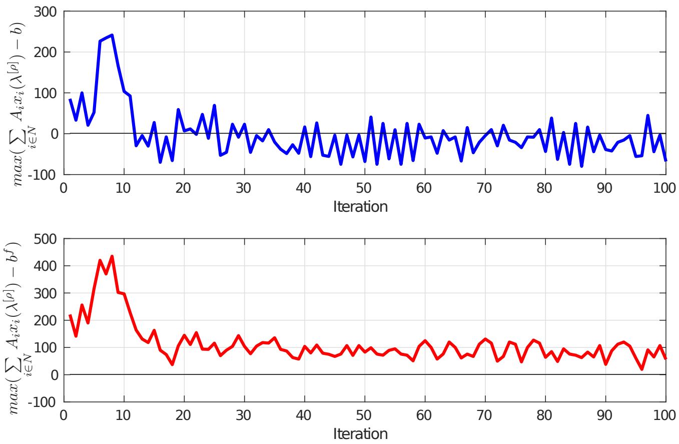

Fig. 1. Outcome of Algorithm 1 for randomly generated instances of problem (1): the upper plot shows the maximal violation of the original coupling constraint (2) by $x_i^{[\lambda^{[\rho]}]}$, $i \in N$, over the iterations. Values below zero indicate that no violation is observed. The lower plot shows, to the contrary, the maximal violation of the tightened coupling constraint (8) over the iterations. Notice that $x_i^{[\lambda^{[\rho]}]}$, $i \in N$ does not satisfy the tightened coupling constraint (8) for the first 100 iterations by applying the sub-gradient method, but has already satisfied the original one (2) already for $\rho = 11$, i.e., the tightened constraint has to be assessed as overly conservative.

$J(\mathbf{x}^f)$, a better candidate than $\mathbf{x}^f$ is found, and Algorithm 1 terminates. As a result, the objective to improve $\mathbf{x}^f$ is realized before $\bar{\mathbf{x}}^{*,f}$ has converged by the additional solution of (15), leading to a significant reduction of the computation time, as will be shown below.

It is remarked that the satisfaction of the conditions in line 8 of Algorithm 1 is not guaranteed in general. However, the probability that the conditions are satisfied rises with increasing ρ: since the multiplier $\lambda^{[\rho]}$ converges towards its optimal value $\bar{\lambda}^{*,f}$ for $\rho \to \infty$, the $x_i^{[\lambda^{[\rho]}]}$, $i \in N$ will eventually differ in at most $m+1$ subsystems from $\bar{\mathbf{x}}^{*,f}$ according to Theorem 2. As $\bar{\mathbf{x}}^{*,f}$ satisfies the tightened coupling constraint (8), and leads to low global costs according to Theorem 3, the $x_i^{[\lambda^{[\rho]}]}$, $i \in N$ thus also tend to satisfy the tightened coupling constraint (8) (and thus the original one (2)) and low global costs for $\rho \to \infty$.

Note that the solution of the LP problem (15) can be carried out in a centralized fashion (e.g. by a central coordinator), since the required computational effort is negligible compared to the MILP problem (1), given that only $\sum_{i\in N} r_i$ real variables are involved

2.2 *Further Improvement of* $\mathbf{x}^f$

Now consider the case that the conditions in line 8 of Algorithm 1 are never satisfied. Then, no feasible candidate better than $\mathbf{x}^f$ is found by the algorithm, and it terminates after $\bar{\mathbf{x}}^{*,f}$ is converged. Thereafter, for $\bar{\mathbf{x}}^{*,f}$, a set of $|N_1| \geq n_s - m - 1$ subsystems can be detected according to Theorem 1, such that for $i \in N_1$ the local solution $\bar{x}_i^{*,f}$ (as contained in $\bar{\mathbf{x}}^{*,f}$) also satisfies X_i. At this stage, in order to determine a feasible candidate of (1) being close to $\bar{\mathbf{x}}^{*,f}$ (thus better than $\mathbf{x}^f$), the following problem is set up for the remaining $|N_2| \leq m+1$ subsystems:

$$\min_{x_i, \forall i \in N_2} \sum_{i\in N_2} c_i^{\mathrm{T}} x_i \tag{19}$$

$$\text{s.t.:} \sum_{i\in N_2} A_i x_i \leq b - \sum_{i\in N_1} A_i \bar{x}_i^{*,f} \tag{20}$$

$$x_i \in X_i,\; x_i \in \mathbb{R}^{r_i} \times \mathbb{Z}^{z_i},\; \forall i \in N_2 \tag{21}$$

The solution of (19) aims at recovering the local feasibility $x_i \in X_i$ of the remaining $|N_2|$ subsystems, while the feasibility of the original coupling constraint (2) is maintained. Note that, a new feasible candidate of (1) can be determined if problem (19) has a feasible solution and the optimized solution $x_i^{*,rec}$, $i \in N_2$ is found. Let the new candidate still be denoted by $\mathbf{x}^{new} = [x_1^{new}; \cdots; x_{n_s}^{new}]$, but $x_i^{new} := \bar{x}_i^{*,f}$, $\forall i \in N_1$ and $x_i^{new} := x_i^{*,rec}$, $\forall i \in N_2$. Then, if $J(\mathbf{x}^{new})$ attains lower global costs than $J(\mathbf{x}^f)$, a candidate better than $\mathbf{x}^f$ is found. However, the feasibility of problem (19) can also not be guaranteed in general, in most cases due to violation of (20). But the following facts indicate that the existence of a feasible solution to (19) is likely:

(1) Due to $|N_2| \ll |N_1|$ (since $m \ll n_s$), only a small fraction of the n_s subsystems need to re-select their local share of $\bar{\mathbf{x}}^{*,f}$. This implies that the left-hand side of the coupling constraint (2), i.e., $\sum_{i\in N} A_i x_i = \sum_{i\in N_1} A_i x_i + \sum_{i\in N_2} A_i x_i$, will not deviate much from $\sum_{i\in N} A_i \bar{x}_i^{*,f}$ after the $|N_2|$ subsystems have re-selected their local candidates in (19);

(2) for the new local candidates of the $|N_2|$ subsystems, the left-hand side of the coupling constraint (2) only has to be smaller than b instead of b_f in (20) (as $\sum_{i\in N} A_i \bar{x}_i^{*,f} \leq b_f \leq b$ holds according to (11)).

Anyhow, if (19) has no feasible solution, or $J(\mathbf{x}^{new}) < J(\mathbf{x}^f)$ fails to hold, no improvement can be obtained for $\mathbf{x}^f$ by the proposed method. Then, the maximal difference between $J(\mathbf{x}^f)$ and the globally optimal costs $J(\mathbf{x}^*)$ can be assessed by the following theorem:

Theorem 4. For given $\mathbf{x}^f$, the difference between $J(\mathbf{x}^f)$ and $J(\mathbf{x}^*)$ is bounded by:

$$J(\mathbf{x}^f) - J(\mathbf{x}^*) \leq J(\mathbf{x}^f) - J(\bar{\mathbf{x}}^{*,f}) + \bar{\lambda}^{*,f}(b - b_f). \tag{22}$$

Proof. According to the relations listed in Theorem 3 the following applies:

$$\begin{aligned} J(\mathbf{x}^f) - J(\mathbf{x}^*) &\leq J(\mathbf{x}^f) - J(\bar{\mathbf{x}}^*) \qquad (23)\\ &\leq J(\mathbf{x}^f) - J(\bar{\mathbf{x}}^{*,f}) + J(\bar{\mathbf{x}}^{*,f}) - J(\bar{\mathbf{x}}^*). \end{aligned}$$

As the LP problem (11) is transformed into (5) by perturbing the dualized constraints according to (9), the difference between their optimal costs are thus bounded by:

$$J(\bar{\mathbf{x}}^{*,f}) - J(\bar{\mathbf{x}}^*) \leq \bar{\lambda}^{*,f}(b - b_f). \tag{24}$$

See Chapter 5.6 in Boyd and Vandenberghe (2004) for a detailed explanation of this inequality. By substituting inequality (24) into (23), the relation (22) is obtained. □

Algorithm 2

```
Initialization: $\mathbf{x}^f$, $flag = 0$;
while $flag = 0$ do
    determine coupling constraint (8) with $\mathbf{x}^f$
    formulate (10), (11) and (12) with (8);
    run Algorithm 1 and:
    if the conditions in line 8 of Algorithm 1 are satisfied before $\rho^{max}$ is reached then
        a better candidate $\mathbf{x}^{new}$ is found and:
            $\mathbf{x}^f := \mathbf{x}^{new}$
    else
        solve (19);
        if a feasible candidate $\mathbf{x}^{new}$ exists for (19) and if it satisfies: $J(\mathbf{x}^{new}) < J(\mathbf{x}^f)$ then
            $\mathbf{x}^f := \mathbf{x}^{new}$
        else
            $flag = 1$
        end if
    end if
end while
```

As $\bar{\lambda}^{*,f}$ and $\bar{\mathbf{x}}^{*,f}$ have both been determined through Algorithm 1, the value of the right-hand side of (22) can be directly calculated, which gives an upper bound of the performance loss of $\mathbf{x}^f$ compared to the global optimum $\mathbf{x}^*$.

2.3 The Overall Procedure

As the last two paragraphs have explained, a better candidate $\mathbf{x}^{new}$ can be obtained by solving different mechanisms, and Algorithm 2 shows how these can be combined to an overall procedure. This procedure does not require conservative assumptions as established in previous work, i.e. the applicability of the proposed method is significantly enhanced. Note that, in the best case for the computation time, $\mathbf{x}^{new}$ is found when the conditions in line 8 of Algorithm 1 are satisfied for the first time. Then, the sub-gradient method does not have to be executed until the primal/dual optimum is reached. If these conditions are not satisfied in any iteration of Algorithm 1, $\mathbf{x}^{new}$ is determined by solving (19). Only if no feasible solution exists also for (19), or the relation $J(\mathbf{x}^{new}) < J(\mathbf{x}^f)$ fails to hold, no candidate better than $\mathbf{x}^f$ is found, and the performance loss of $\mathbf{x}^f$ compared to $\mathbf{x}^*$ is checked by use of (22). Finally, if a better candidate $\mathbf{x}^{new}$ is found, $\mathbf{x}^f$ is set equal to $\mathbf{x}^{new}$, and the iteration is repeated. The whole procedure stops if no improvement is found, or an upper bound on the computation time is reached.

3. NUMERICAL EXAMPLE

In this section, the proposed distributed solution was tested for various MILP problems (1) of different size. The local cost function c_i, local constraints X_i and the coupling constraints (2) are in each test randomly generated, while instances without feasible solutions are discarded. In addition, no particular requirement on the coupling constraints (2) are imposed, as done in Vujanic et al. (2014), nor is any feasibility assumption used for the tightened problem as in Vujanic et al. (2016).

In the first test instance, a number of $n_s = 40$ subsystems was considered, each with $z_i = r_i = 15$ integer and real variables. The number of coupling constraints is $m = 5$. For comparison purposes, the centralized solution of this problem, which involves in total 600 integer variables, is found in $336sec$ by using the solver CPLEX, IBM ILOG (2009) on a 3.4GHZ processor, and the optimal cost is $J(\mathbf{x}^*) = -2.17 \cdot 10^5$. However, the first feasible candidate is determined already after only $3.34sec$, but with a cost of $J(\mathbf{x}^f) = -211.92$.
By employing the proposed distributed solution to improve $\mathbf{x}^f$, a better candidate $\mathbf{x}^{new,1}$ is found after only 6 iterations in Algorithm 1 ($0.35sec$), with a cost of $J(\mathbf{x}^{new,1}) = -2.14 \cdot 10^5$, i.e. a performance loss of only 1.13% compared to $J(\mathbf{x}^*)$. Then, starting from $\mathbf{x}^{new,1}$ by executing the proposed method once more, an even better candidate $\mathbf{x}^{new,2}$ with $J(\mathbf{x}^{new,2}) = -2.16 \cdot 10^5$ is found after only 5 iterations in Algorithm 1, taking $0.48sec$. In this iteration, the performance loss is further reduced to 0.46%. Thereafter, no further improvement can be made.

For a larger problem instance with $n_s = 80$ subsystems, each with $z_i = r_i = 25$ integer and real variables (leading to overall 2000 integer variables in the centralized problem), and with $m = 8$ coupling constraints. The global optimum could not be found by centralized solution within 1 hour using CPLEX, but it only took $77sec$ to find the first feasible candidate $\mathbf{x}^f$ with $J(\mathbf{x}^f) = -2.32 \cdot 10^3$. The proposed method then generates a better candidate $\mathbf{x}^{new,1}$ with $J(\mathbf{x}^{new,1}) = -6.04 \cdot 10^5$ after 6 iterations ($6.27sec$) in Algorithm 1. By employing a further iteration of Algorithms 2 for $\mathbf{x}^{new,1}$, an even better candidate $\mathbf{x}^{new,2}$ was found after 7 iterations in Algorithm 1 (in $9.06sec$) and with $J(\mathbf{x}^{new,2}) = -6.07 \cdot 10^5$. Further improvements were not found, but the difference between $J(\mathbf{x}^{new,2})$ and $J(\mathbf{x}^*)$ is bounded by:

$$J(\mathbf{x}^{new,2}) - J(\mathbf{x}^*) \leq 0.034 \cdot 10^5 \qquad (25)$$

according to (22). Thus, although it is hard to compute the optimal cost $J(\mathbf{x}^*)$ due to the high computational complexity, one can ensure at most 0.55% performance loss for $J(\mathbf{x}^{new,2})$ through (25).

In the third test instance, a number of $n_s = 200$ subsystems was considered, and $z_i = r_i = 10$, $m = 12$. The global optimum could again not be found within 1 hour, but the first feasible candidate $\mathbf{x}^f$ with $J(\mathbf{x}^f) = -1.69 \cdot 10^4$ was obtained in only $7.23sec$. The proposed method produces a better candidate $\mathbf{x}^{new,1}$ with $J(\mathbf{x}^{new,1}) = -7.57 \cdot 10^5$ after $2.09sec$. No better candidate was found afterwards, and the maximal performance loss compared to $J(\mathbf{x}^*)$ is bounded by 3.56% according to (22).

The tests above show that the proposed method in all cases achieves drastic improvements of $\mathbf{x}^f$ within a very short computation time (in particular with the first iteration of Algorithm 2). The obtained candidates attain global costs that are only slightly worse than the global optima. Even for the case that the global optimum cannot be determined in centralized fashion due to the high complexity of (1), the bound defined in (22) still enables one to evaluate the obtained candidate.

A set of additional tests are listed in Table 1.

Table 1. Numerical experiments for different MILP problems with T indicating the time required for the solution of $\mathbf{x}^f$, $\mathbf{x}^{new,1}$ and $\mathbf{x}^*$ ($J(\mathbf{x}^*) = -$ indicates that the global optimum was not found within given time limit).

N	z_i, r_i	m	$T_{\mathbf{x}^f}$	$J(\mathbf{x}^f)$	$T_{\mathbf{x}^{new,1}}$	$J(\mathbf{x}^{new,1})$	$T_{\mathbf{x}^*}$	$J(\mathbf{x}^*)$
10	10	3	$0.20sec$	335.41	$0.11sec$	$-3.27 \cdot 10^4$	$12.23sec$	$-3.40 \cdot 10^4$
10	40	3	$4.99sec$	$2.89 \cdot 10^3$	$2.37sec$	$-1.05 \cdot 10^5$	$> 20min$	$-$
20	10	4	$0.53sec$	-680.07	$0.18sec$	$-8.29 \cdot 10^4$	$3.59sec$	$-8.29 \cdot 10^4$
20	30	4	$4.43sec$	$-2.04 \cdot 10^3$	$0.76sec$	$-1.37 \cdot 10^5$	$> 20min$	$-$
40	10	6	$1.28sec$	$-1.65 \cdot 10^3$	$0.39sec$	$-1.67 \cdot 10^5$	$67.97sec$	$-1.68 \cdot 10^5$
40	20	8	$11.58sec$	$4.66 \cdot 10^3$	$1.98sec$	$-2.53 \cdot 10^5$	$> 20min$	$-$
80	5	5	$0.62sec$	$5.93 \cdot 10^3$	$0.18sec$	$-1.93 \cdot 10^5$	$3.47sec$	$-2.01 \cdot 10^5$
80	8	7	$1.36sec$	$-1.80 \cdot 10^3$	$0.59sec$	$-2.50 \cdot 10^5$	$416sec$	$-2.59 \cdot 10^5$
160	4	8	$0.60sec$	$3.00 \cdot 10^3$	$0.10sec$	$-2.28 \cdot 10^5$	$21.40sec$	$-3.23 \cdot 10^5$
160	20	8	$65sec$	394.33	$2.52sec$	$-6.00 \cdot 10^5$	$> 20min$	$-$
200	5	9	$2.70sec$	$-5.58 \cdot 10^3$	$0.22sec$	$-3.21 \cdot 10^5$	$72sec$	$-5.03 \cdot 10^5$
300	20	15	$231sec$	$-1.85 \cdot 10^3$	$18sec$	$-8.19 \cdot 10^5$	$> 20min$	$-$
500	10	10	$23sec$	$9.26 \cdot 10^3$	$26sec$	$-2.01 \cdot 10^6$	$> 20min$	$-$

4. CONCLUSION

This work has proposed a distributed scheme for the solution of MILP problems of type (1), as encountered, e.g., in the optimal control problem of networked systems involving hybrid dynamics. This work exploits the results established in the Shapley-Folkman-Starr theorem to decompose the centralized problem, such that a set of small-scale problems are to be solved in parallel, thus reducing the overall complexity. Unlike the existing algorithms for the same problem class (which are based as well on the named theorem), the proposed method aims at improving the feasible candidates iteratively, instead of directly computing a sub-optimal solution of (1). It was shown that the proposed procedure does not employ quite conservative assumptions used in previous work, and that the close-to-optimal solutions could be obtained very efficiently for a large number of tests. The current work aims at extending the proposed method to the distributed solution of mixed-integer quadratic and mixed-integer nonlinear programming problems.

REFERENCES

Anstreicher, M. and Wolsey, A. (2009). Two “well-known” properties of subgradient optimization. *Mathematical Programming*, 120(1), 213–220.

Aubin, J. and Ekeland, I. (1976). Estimates of the duality gap in nonconvex optimization. *Mathematics of Operations Research*, 1(3), 225–245.

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., and Vance, P. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations research*, 46(3), 316–329.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Bertsekas, D. (2009). *Convex optimization theory.* Athena Scientific Belmont.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization.* Cambridge university press.

Camisa, A., Notarnicola, I., and Notarstefano, G. (2018). A primal decomposition method with suboptimality bounds for distributed mixed-integer linear programming. In *IEEE Conf on Decision and Control*, 3391–3396.

Camisa, A. and Notarstefano, G. (2019). Primal decomposition and constraint generation for asynchronous distributed mixed-integer linear programming. In *IEEE 18th European Control Conference*, 77–82.

CPLEX, IBM ILOG (2009). User’s manual for cplex, v12.1. *Int. Business Machines Corporation*, 46(53), 157.

Falsone, A., Margellos, K., and Prandini, M. (2019). A decentralized approach to multi-agent milps: finite-time feasibility and performance guarantees. *Automatica*, 103, 141–150.

Geoffrion, M. (1974). Lagrangean relaxation for integer programming. In *Approaches to integer programming*, 82–114. Springer.

Groß, D. and Stursberg, O. (2013). Distributed predictive control for a class of hybrid systems with event-based communication. *IFAC Proceedings Volumes*, 46(27), 383–388.

Liu, Z. and Stursberg, O. (2018). Optimal trajectory planning of hybrid systems by efficient miqp encoding. In *IEEE Conf on Decision and Control*, 1548–1553.

Liu, Z. and Stursberg, O. (2019a). Distributed control of networked systems with coupling constraints. *at-Automatisierungstechnik*, 67(12), 1007–1018.

Liu, Z. and Stursberg, O. (2019b). Efficient optimal control of hybrid systems with conditioned transitions. In *IEEE 15th Int Conf on Control and Automation*, 223–230.

Vujanic, R., Esfahani, M., Goulart, J., Mariéthoz, S., and Morari, M. (2016). A decomposition method for large scale milps, with performance guarantees and a power system application. *Automatica*, 67, 144–156.

Vujanic, R., Esfahani, M., Goulart, P., and Morari, M. (2014). Large scale mixed-integer optimization: A solution method with supply chain applications. In *IEEE 22nd Mediterranean Conf on Control and Automation*, 804–809.