2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

# Tracking control for Petri nets with forbidden states

Raphael Fritz* and Ping Zhang*

***Abstract*— In this paper, an approach for tracking control for Petri nets with forbidden states is presented. The control aim is that the Petri net is forced by a firing sequence from an initial marking into a destination marking while avoiding the forbidden states described by Generalized Mutual Exclusion Constraints (GMEC). A new type of constraints called NOT-GMEC is introduced to allow a compact representation of specific token-place combinations. The firing sequence is determined by transforming the constraints into a suitable form and executing a two-step algorithm.**

## I. INTRODUCTION

Discrete event systems (DES) are a useful tool to model the dynamics of modern flexible manufacturing systems and transportation systems [1]. In this paper we shall consider the tracking control of DES described by peti nets. The control objective is to steer the plant from an initial marking to a desired marking while avoiding forbidden states.

The behavior of DES is often subject to specifications, such as safety or resource constraints. Supervisory control is used to restrict the behavior of DES to an acceptable closed-loop behavior. For Petri nets (PN), Generalized Mutual Exclusion Constraints (GMEC) are an often used tool for the description of such constraints [2]. The focus of supervisory control of PNs lies on the prevention of forbidden states by deactivating transitions. The system should be restricted as little as possible to ensure maximal permissiveness. In contrast, in this paper the objective is to force the system into a desired marking by activating transitions in a suitable and feasible sequence.

The computation of firing sequences has been studied in the framework of reachability analysis [3]–[5] and scheduling [6]–[11]. An optimal firing sequence is found in [3] by solving an integer linear programming (ILP) problem. Supervisory control theory (SCT) and ILP are combined in [4] for a reconfiguration approach under changing resource constraints. Several approaches using heuristics exists in the literature. The reachability graph is partially explored in [6] and [7], a genetic algorithm is used in [8] and in [9] a dynamic search window with best-first algorithm is presented.

A few approaches combine tracking control with the forbidden states problem of supervisory control. A compact representation of the reachability graph called basis reachability graph (BRG) is developed in [5]. The basic principle of model predictive control is applied in [10] and [11] to realize the tracking of a given destination marking by solving an optimization problem at each step.

In comparison to [5], the presented approach can easily handle structural changes of the PN. In [5] structural changes require the recomputation of the BRG. The MPC approach in [11] use the firing count vector for the evaluation of the next transition to fire. This can lead to problems if the firing count vector represents infeasible firing sequences or reaches forbidden markings. The MPC approach has generally the problem that the destination marking can be reached only in a certain percentage of cases.

The approach proposed in this paper has the advantage that it can easily adapt to changing PN structure and guarantees that a feasible firing sequence can be found as long as such a sequence exists. The main contributions of this paper are:

- A tracking control approach is proposed to determine a feasible firing sequence while considering a set of forbidden markings.
- A constraint called NOT-GMEC, which allows to forbid a specific token-place combination, is presented.
- The forbidden states are modelled by GMECs [2], OR-AND GMECs [12] and NOT-GMECs. It is shown how to transform these constraints into a suitable form that can be easily integrated in integer programming problems.
- A two-step algorithm is given, which finds the firing sequence by first restricting the search space and then computing the optimal firing sequence (i.e. the shortest or cost optimal firing sequence).

The paper is organized as follows. Section II introduces some definitions. The constraint transformation for GMECs is introduced in Section III. NOT-GMECs and the constraint transformation is presented in Section IV and V. The tracking control approach for PNs is presented in Section VI and an example is given in Section VII.

## II. PRELIMINARIES

### A. Petri net

This section provides the necessary definitions that will be used later. The used PN theory is based on [13].

A Petri net structure can be represented by the four-tuple $PN = (P, T, N^+, N^-)$, with a set of $m$ places $P = \{p_1, p_2, \dots, p_m\}$ and a set of $n$ transitions $T = \{t_1, t_2, \dots, t_n\}$. The post-incidence matrix $N^+$ (pre-incidence matrix $N^-$) specifies the arcs and their weights from transitions to places (from places to transitions). The $(m \times n)$ matrix $N = N^+ - N^-$ is the incidence matrix.

A transition $t_j \in T$ is enabled at marking $M(k)$, only if

$$N^- q(k) \leq M(k) \tag{1}$$

is fulfilled. Transition $t_j$ is represented by the $n$-dimensional firing vector $q(k) = \begin{pmatrix} q_1(k) & q_2(k) & \cdots & q_n(k) \end{pmatrix}^T$, whose $j$-th entry $q_j(k)$ is 1, while all other entries are 0.

*The authors are with the Institute of Automatic Control, University of Kaiserslautern, Erwin-Schroedinger 12, 67653 Kaiserslautern, Germany (e-mail: fritzr@eit.uni-kl.de; pzhang@eit.uni-kl.de)

The new marking $M(k+1)$, resulting from firing transition $t_j \in T$ at time instant $k$ is determined by the PN state equation

$$M(k+1) = M(k) + Nq(k) \tag{2}$$

and denoted by $M(k)[t_j\rangle M(k+1)$.

Assume that $\sigma = t_{q(0)}t_{q(1)}\dots t_{q(k-1)}$ is a firing sequence of transitions with the length $|\sigma| = k$. The firing count vector $q$ corresponding to the firing sequence $\sigma$ is the sum of all firing vectors $q(i),\ i = 0, 1, \dots, k-1$, thus $q = \sum_{i=0}^{k-1} q(i)$.

Activating the firing sequence $\sigma$ under the initial marking $M(0)$ leads to the marking $M(k)$, described by

$$M(k) = M(0) + Nq \tag{3}$$

This process can be denoted as $M(0)[\sigma\rangle M(k)$.

The Kronecker product of matrices is denoted by $A \otimes B$. The entry-wise square of a matrix $A_{m\times n}$ (also called Hadamard Power [14]) is denoted by $A^{\circ 2}_{m\times n}$. The superscript $T$ denotes the transpose of a matrix. $1_{1\times n} = \begin{pmatrix}1 & \cdots & 1\end{pmatrix}$ is a row vector, $I_{n\times n}$ is an identity matrix and $O_{m\times n}$ is a zero matrix where the subscripts denote the dimensions.

### *B. Forbidden states: GMEC and its derivatives*

Forbidden states can be described by GMECs and define a set of legal markings [2].

*Definition 1 (GMEC [2]):* A GMEC is a pair $(w, b)$ with $w \in \mathbb{Z}^m$ and $b \in \mathbb{Z}$ and defines a set of legal markings

$$\mathcal{L}(w, b) = \{M \in \mathbb{N}^m | w^T M \le b\} \tag{4}$$

This definition can be extended. For an AND-GMEC, every single GMEC must be fulfilled.

*Definition 2 (AND-GMEC [12]):* An AND-GMEC is a set of $s$ GMECs defined by a pair $(W, b)$ with $W{=}\begin{pmatrix}w_1 \cdots w_s\end{pmatrix} \in \mathbb{Z}^{m\times s}$ and $b{=}\begin{pmatrix}b_1 \cdots b_s\end{pmatrix}^T \in \mathbb{Z}^s$. The set of legal markings is

$$\mathcal{L}_{AND}(W, b){=}\{M \in \mathbb{N}^m | \forall i \in \{1, \dots, s\}, w_i^T M \le b_i\} \tag{5}$$

In the case of an OR-GMEC, at least one of the contained single GMEC must hold.

*Definition 3 (OR-GMEC [12]):* An OR-GMEC is a set of $r$ single GMECs defined by the set $W_{OR} = \{(w_1, b_1), \dots, (w_r, b_r)\}$, where at least one of them is fulfilled. The set of legal markings for an OR-GMEC is

$$\mathcal{L}_{OR}(W_{OR}){=}\{M \in \mathbb{N}^m | \exists i \in \{1, \dots, r\}, w_i^T M \le b_i\} \tag{6}$$

For OR-AND GMEC, at least one AND-GMEC must hold.

*Definition 4 (OR-AND GMEC [12]):* An OR-AND GMEC is a set of $r$ AND-GMECs and at least one of them is fulfilled. It is represented by a set $W_{OA} = \{(W_1, b_1), \dots, (W_r, b_r)\}$ with $W_i \in \mathbb{Z}^{m\times s_i}$ and $b_i \in \mathbb{Z}^{s_i}$ and defines a set of legal markings

$$\mathcal{L}_{OA}(W_{OA}){=}\{M \in \mathbb{N}^m | \exists i{\in}\{1, \dots, r\}, W_i^T M {\le} b_i\} \tag{7}$$

### *C. Computation of the firing sequence*

In this section, we briefly review an approach given in [3] that finds a feasible firing sequence $\sigma = t_{q(0)}t_{q(1)}\dots t_{q(d-1)}$ from an initial marking $M(0) = M_0$ to a destination marking $M(d) = M_d$ for a given firing sequence length $|\sigma| = d$.

The solution vector contains the firing vectors $q(k), k = 0, 1, \dots, d-1$ for each time instant, i.e. $x = \begin{pmatrix}q^T(d-1) & q^T(d-2) & \cdots & q^T(0)\end{pmatrix}^T = \begin{pmatrix}x_1 & x_2 & \dots & x_{nd}\end{pmatrix}^T$ with $x_l \in \{0, 1\},\ l = 1, 2, \dots, nd$.

The objective function of [3] is replaced by $F(x) = 1_{1\times nd}x$, which acts as a placeholder objective function and minimizes the number of transitions. In Section VI a more general objective function is proposed.

Only a single transition is fired at each time instant, i.e. $\sum_{i=1}^{n} q_i(k){=}1, k = 0, .., d{-}1$. Rewriting this results in

$$(I_{d\times d} \otimes 1_{1\times n})x = 1_{d\times 1} \tag{8}$$

Due to the state equation of the PN we know that

$$\underbrace{Nq(d-1) + \cdots + Nq(0)}_{(1_{1\times d} \otimes N)x} = M_d - M_0 \tag{9}$$

Based on (2) and $N = N^+ - N^-$ we have $M(k) = M(k+1) - Nq(k) = M(k+1) - (N^+ - N^-)q(k)$. Inserting $M(k)$ into (1) and reformulating, we get $N^+q(k) \le M(k+1)$. For any $k = 0, 1, \dots, d-2$, it holds $M_d = M(k+1) + N\sum_{i=k+1}^{d-1} q(i)$. Therefore $N^+q(k) \le M(k+1) = M_d - N\sum_{i=k+1}^{d-1} q(i)$ and the following inequalities hold

$$\underbrace{\begin{pmatrix} N^+ & O_{m\times n} & \cdots & O_{m\times n} \\ N & N^+ & \cdots & O_{m\times n} \\ \vdots & \vdots & \ddots & \vdots \\ N & N & \cdots & N^+ \end{pmatrix}}_{T_e} x \le \underbrace{\begin{pmatrix} M_d \\ M_d \\ \vdots \\ M_d \end{pmatrix}}_{1_{d\times 1} \otimes M_d} \tag{10}$$

By combining (8), (9), (10) with $F(x)$, a feasible firing sequence can be found by solving the following ILP problem

$$\begin{cases} \min\limits_{x} & F(x) \\ \text{subject to} & \begin{pmatrix} I_{d\times d} \otimes 1_{1\times n} \\ 1_{1\times d} \otimes N \end{pmatrix} x = \begin{pmatrix} 1_{d\times 1} \\ M_d - M_0 \end{pmatrix} \\ & T_e x \le 1_{d\times 1} \otimes M_d \\ & x_l \in \{0, 1\},\ l = 1, 2, \dots, nd \end{cases} \tag{11}$$

The optimal firing sequence $\sigma = t_{q(0)}t_{q(1)}\dots t_{q(d-1)}$ is obtained by transforming the firing vectors $q(0), q(1), \dots, q(d-1)$ in the optimal solution vector $x$ back into the corresponding transitions.

*Remark 1:* ILP (11) finds a firing sequence for a given length $d$. Generally $d$ is unknown and multiple possible lengths need to be tested until a feasible firing sequence is found. In [4], the firing sequence length is bounded by $1 \le d \le d_{\max}$ with $d_{\max} = 5n$ to $10n$. By incremental increase of $d$ a feasible solution can be found. This incremental approach can lead to a high computational effort. In Section VI-B a different method is proposed that reduces the search space to only promising solution candidates.

### *D. Problem formulation*

The firing sequence resulting from ILP (11) can be infeasible if forbidden states are introduced into the system. Hence an ILP problem that also considers forbidden states is required. The problem discussed in this paper is the tracking of a destination marking $M_d$ in a PN with forbidden states by finding an appropriate firing sequence, i.e. $M_0[\sigma_r\rangle M_d$ with $\sigma_r = t_{q(0)}t_{q(1)}\dots t_{q(d-1)}$ and unknown $d$.

## III. Constraints for GMECs

In this section, the constraints of Section II-B are transformed to be used in an ILP problem. We assume that $M_0$ and $M_d$ are legal markings.

**GMEC:** The GMEC (4) can not be directly added to ILP (11). Recalling (3), the requirement imposed by $w^TM(k) \leq b$ is equivalent to $w^T(M_0+Nq) \leq b$. Replacing the firing count vector by the corresponding firing vectors results in

$$w^TN(q(0)+\dots+q(k-1)) \leq b - w^TM_0 \qquad (12)$$

**AND-GMEC:** Derive a constraint in form of (12) for every single GMEC in the AND-GMEC.

**OR-GMEC:** Logical combinations of constraints can not be implemented as shown in (12). A suitable approach for the implementation of logical combinations of constraints in ILP problems is the big-M method [15]. By adding new variables and new constraints to the ILP problem, only a certain number of constraints have to be fulfilled.

For every GMEC a binary variable is necessary, which results in $r$ new binary variables $y_1,\dots,y_r$. For every GMEC $(w_i, b_i)$ in $W_{OR}$ a constraint in the form of (12) is derived and on the right side the term $By_i,\ i = 1,\dots,r$ with $B$ as a large enough positive integer, is added. If $y_i = 0$, then the $i$-th constraint holds. In the case $y_i = 1$ the constraint does not hold. An additional constraint is necessary, so that at least one constraint holds (i.e. at least one $y_i = 0$). That leads to the following constraints

$$\begin{cases} w_1^TN(q(0)+\dots+q(k-1)) \leq b_1 - w_1^TM_0 + By_1 \\ \vdots \\ w_r^TN(q(0)+\dots+q(k-1)) \leq b_r - w_r^TM_0 + By_r \\ \sum_{i=1}^r y_i \leq r-1 \end{cases} \qquad (13)$$

For every OR-GMEC and every time instant $k = 1,\dots,d-1$ a constraint in the form of (13) is required.

**OR-AND GMEC:** The constraints can be modeled based on the constraint modeling for OR-GMECs. Every of the $r$ AND-GMECs in the OR-AND GMEC is associated with the same binary variable $y_i,\ i = 1,\dots r$.

The constraint transformation of OR-GMECs and OR-AND GMECs adds $n_y$ additional variables to the optimization problem. Consider that $|OR|$ is the number of OR-GMECs and $r_i^{OR}$ is the number of GMECs in the $i$-th OR-GMEC. This also applies to OR-AND GMEC (i.e. $|OA|$ and $r_i^{OA}$). The number of additional variables is then $n_y = n_y^{OR} + n_y^{OA}$ with $n_y^{OR} = (d-1)\sum_{i=1}^{|OR|} r_i^{OR}$ and $n_y^{OA} = (d-1)\sum_{i=1}^{|OA|} r_i^{OA}$.

## IV. NOT-GMECs

This section introduces the notion of a new subclass of GMECs called NOT-GMEC and its derivatives.

Forbidden states can represent a multitude of different cases that are undesired or safety-critical. For example adherence of resource constraints or prevention of safety-critical states that can endanger a manufacturing system or human lifes.

The presented constraint allows for a more compact representation if a specified number of tokens in a place is forbidden. This allows to specify an exact token-place combination that should be forbidden.

*Definition 5 (NOT-GMEC):* A NOT-GMEC is a pair $(w, b)$ with $w \in \mathbb{Z}^m$ and $b \in \mathbb{Z}$ and defines a set of legal markings

$$\mathcal{L}_N(w, b) = \{M \in \mathbb{N}^m | w^TM \neq b\} \qquad (14)$$

This definition can be extended into AND-NOT GMEC, OR-NOT GMEC and OR-AND-NOT GMEC.

*Definition 6 (AND-NOT GMEC):* An AND-NOT is a set of $s$ NOT-GMECs defined by a pair $(W, b)$ with $W = (w_1 \cdots w_s) \in \mathbb{Z}^{m\times s}$ and $b = (b_1 \cdots b_s)^T \in \mathbb{Z}^s$. It defines a set of legal markings

$$\mathcal{L}_{AN}(W,b) = \{M \in \mathbb{N}^m | \forall i \in \{1,\dots,s\}, w_i^TM \neq b_i\} \qquad (15)$$

*Definition 7 (OR-NOT GMEC):* An OR-NOT GMEC is a set of $r$ single NOT-GMECs , where at least one of them is fulfilled. It is represented by a set $W_{ON} = \{(w_1, b_1),\dots,(w_r, b_r)\}$. It defines a set of legal markings

$$\mathcal{L}_{ON}(W_{ON}) = \{M \in \mathbb{N}^m | \exists i \in \{1,\dots,s\},\ w_i^TM \neq b_i\} \qquad (16)$$

*Definition 8 (OR-AND-NOT GMEC):* An OR-AND-NOT GMEC is a set of $r$ AND-NOT GMECs , where at least one of them is fulfilled. It is represented by a set $W_{OAN} = \{(W_1, b_1),\dots,(W_r, b_r)\}$ with $W_i \in \mathbb{Z}^{m\times s_i}$ and $b_i \in \mathbb{Z}^{s_i}$ and defines a set of legal markings

$$\mathcal{L}_{OAN}(W_{OAN}) = \{M \in \mathbb{N}^m | \exists i \in \{1,\dots,r\}, W_i^TM \neq b_i\} \qquad (17)$$

The advantage of NOT-GMECs in comparison to GMECs is that the number of required constraints can be reduced. In general for a single NOT-GMEC, an OR-GMEC containing two GMECs is required. The NOT-GMEC $M(p) \neq b,\ p \in P$ can be described by the OR-GMEC $M(p) \leq b-1 \vee M(p) \geq b+1$. In particular the OR-NOT GMEC is a powerful description tool for forbidden states. It allows to forbid a specific token-place combination.

*Remark 2:* In comparison to [12], we allow the right side of the constraints in Sections II-B and IV represented by $b$ to be all intergers, i.e. $b \in \mathbb{Z}$, while [12] restrict $b$ to natural numbers (i.e. non-negative integers). This allows a wider range of possible constraints.

## V. Constraints for NOT-GMECs

The constraints of the NOT-GMECs for ILP are based on the transformation into quadratic contraints to reduce the total number of constraints.

### A. Constraint transformation of NOT-GMEC

To derive a suitable constraint for an ILP problem from (14), the squared difference between the weighted sum of tokens in marking $M(k)$ and the integer $b$ is calculated.

$$(w^TM(k) - b)^2 \geq 1 \qquad (18)$$

Based on (3) the left side of (18) is indeed

$$\underbrace{(w^TNq)^2}_{\substack{\text{quadratically}\\ \text{dependent on } q}} + \underbrace{2w^TN(w^TM_0 - b)q}_{\text{linearly dependent on } q} + \underbrace{(w^TM_0 - b)^2}_{\text{constant}} \qquad (19)$$

where $w^T M_0$ and $b$ are scalars and $w^T N$ is a row vector.

Inserting (19) into quadratic constraint (18) leads to

$$-(w^T Nq)^2 - 2w^T N(w^T M_0 - b)q \leq -1 + (w^T M_0 - b)^2 \tag{20}$$

The markings $M(k),\ k = 1, \ldots, d-1$, that result from the firing sequence $\sigma$ should fulfill the NOT-GMEC and can be computed by (3). Applying the firing count vectors $q = q(0) + \cdots + q(k-1)$, $k = 1, 2, \ldots, d-1$ to the quadratic part of (19) leads to

$$\begin{aligned}-(w^T Nq)^2 = &- \big(w^T Nq(0)\big)^2 - \cdots - \big(w^T Nq(k-1)\big)^2 \\ &- 2\big(w^T Nq(0)\big)\big(w^T Nq(1)\big) - \cdots \\ &- 2\big(w^T Nq(k-2)\big)\big(w^T Nq(k-1)\big)\end{aligned} \tag{21}$$

Recall that the firing vectors $q(0), \cdots, q(k-1)$ have a single entry equal to 1, while all other entries are 0. For those terms in (21) that depend on a single firing vector, the firing vector selects the entry of row vector $w^T N$ which should be squared. This part of (21) can thus be rewritten as

$$\begin{aligned}&- \big(w^T Nq(0)\big)^2 - \cdots - \big(w^T Nq(k-1)\big)^2 \\ &= - (w^T N)^{\circ 2} q(0) - \cdots - (w^T N)^{\circ 2} q(k-1)\end{aligned} \tag{22}$$

Since $w^T Nq(0), \cdots,\ w^T Nq(k-1)$ are scalars, those terms in (21) that depend on two different firing vectors are

$$\begin{aligned}&- 2\big(w^T Nq(0)\big)\big(w^T Nq(1)\big) - \cdots \\ &- 2\big(w^T Nq(k-2)\big)\big(w^T Nq(k-1)\big) \\ = &- 2q^T(0)\big((w^T N)^T (w^T N\big)q(1) - \cdots \\ &- 2q^T(k-2)\big((w^T N)^T \big((w^T N\big)q(k-1)\end{aligned} \tag{23}$$

By substituting (22) and (23) into (20), we get the quadratic constraint

$$\begin{aligned}&\underbrace{\sum_{s_1=0}^{k-2} \sum_{s_2=s_1+1}^{k-1} q^T(s_1)\Big(-2\big(w^T N\big)^T w^T N\Big)q(s_2)}_{(23)} \\ &+ \underbrace{\sum_{s=0}^{k-1} -(w^T N)^{\circ 2} q(s)}_{(22)} + \sum_{s=0}^{k-1} -2w^T N\big(w^T M_0 - b\big)q(s) \\ &\leq -1 + \big(w^T M_0 - b\big)^2\end{aligned} \tag{24}$$

with $k = 1, 2, \ldots, d-1$. A NOT-GMEC can thus be implemented into an ILP problem by adding a quadratic constraint in form of (24) to the optimization problem.

### *B. Constraints transformation for NOT-GMEC derivatives*

The constraints for an AND-NOT GMEC are derived by adding a constraint in form of (24) for every single NOT-GMEC in the AND-NOT GMEC.

An OR-NOT GMEC containing $r$ single NOT-GMECs can be implemented into an ILP problem using (18) and summing up over the $r$ NOT-GMECs. This results in

$$\sum_{i=1}^{r} (w_i^T M(k) - b_i)^2 \geq 1 \tag{25}$$

with integers $b_i,\ i = 1, \ldots, r$. The conversions for (18) can be applied analogously to (25) resulting in

$$\sum_{i=1}^{r} E_l \leq -1 + \sum_{i=1}^{r} \big(w_i^T M_0 - b_i\big)^2 \tag{26}$$

with $k = 1, 2, \ldots, d-1$ and $E_l$ as the left side of (24).

The constraints for an OR-AND-NOT GMEC can be implemented analogously to the OR-AND GMEC in section III by using the Big-M method. The number of OR-AND-NOT GMEC is $|OAN|$ and $r_i^{OAN}$ is the number of AND-GMECs in the $i$-th OR-AND-NOT GMEC. This results in $n_y^{OAN} = (d-1) \sum_{i=1}^{|OAN|} r_i^{OAN}$ additional variables.

*Remark 3:* As mentioned in Section IV, NOT-GMECs and its derivatives can be represented by GMECs and its derivatives. For the constraint of a NOT-GMEC, an OR-GMEC containing two GMECs is required. Recalling the implementation method of an OR-GMEC in section III, three linear constraints are required. In our presented approach only a single quadratic constraint is required. If an exact marking should be forbidden, i.e. modeled by an OR-NOT GMEC containing $r$ NOT-GMECs, our approach still requires only a single quadratic constraint, while the implementation method based on only OR-GMECs requires $2r + 1$ linear constraints. Another advantage is that the size of the solution vector does not increase, because no additional variables are required.

## VI. TRACKING CONTROL APPROACH

In this section the complete tracking control approach is presented. An integer quadratically constraint programming problem combining the constraints of the previous sections with the ILP (11) is shown. This allows the computation of a feasible firing sequence that forces the PN from the initial marking $M_0$ into the destination marking $M_d$ while avoiding forbidden states modeled by the GMECs in Sections II-B and IV.

### *A. Extended integer programming problem*

For the sake of clarity, we introduce the set of constraints $C$. Set $C$ contains all constraints introduced in Sections II-B and IV. Additionally we introduce the transformed constraints $G(c, k)$. This denotes the transformed constraints derived in Sections III and V for a constraint $c \in C$ and a time instant $k = 1, \ldots, d-1$. These constraints are added to the ILP (11), which results in the integer quadratically constraint programming problem IQCP:

$$\begin{cases} \min\limits_x & F(x) \\ \text{subject to} & \begin{pmatrix} I_{d\times d} \otimes 1_{1\times n} \\ 1_{1\times d} \otimes N \end{pmatrix} x = \begin{pmatrix} 1_{d\times 1} \\ M_d - M_0 \end{pmatrix} \\ & T_e x \leq 1_{d\times 1} \otimes M_d \\ & G(c,k), \quad \forall c \in C, k = 1, \ldots, d-1 \\ & x_l, y_i \in \{0,1\},\ \ l, i = 1, \ldots, nd \end{cases} \tag{27}$$

with $n_y = n_y^{OR} + n_y^{OA} + n_y^{OAN}$ as the total number of additional variables $y_i$ that are introduced in the framework of the Big-M method.

## B. Firing sequence

To find a firing sequence $\sigma$, the firing sequence length $d$ is incrementally increased until a feasible solution for IQCP (27) is found ($1 \leq d \leq d_{\max}$, $d_{\max} = 5n$ to $10n$ [4]). To reduce the resulting computational effort of incremental approach, we propose a two step algorithm consisting of IQCP (27) and an additional ILP problem that determines solution candidates.

Note that a feasible firing sequence must satisfy $Nq = M_d - M_0$ and the length of the firing sequence $d = \sum_{j=1}^{n} q_j = 1_{1\times n} q$. Hence, an additional optimization problem can be formulated as follows to find the shortest possible length of the firing sequence.

$$\mathrm{ILP_d}: \begin{cases} \min\limits_{q} & F(q) \\ \text{subject to} & Nq = M_d - M_0 \\ & -1_{1\times n} q \leq -(d_{\min}+1) \\ & q_l \in \mathbb{N}_0,\ l = 1,2,\ldots,n \end{cases} \tag{28}$$

The objective function is $F(q) = d = \sum_{j=1}^{n} q_j = 1_{1\times n} q$ and the inequality constraint is used to specify the minimal length of the firing sequence by the variable $d_{\min}$. Algorithm 1 results from the combination of (27) and (28). The algorithm tries to find a feasible firing sequence, as long as the maximal number of iteration $i_{\max}$ is not reached. At first $\mathrm{ILP_d}$ (28) is solved to determine a possible shortest firing sequence length. If no feasible solution is found the tracking is not successful. If a feasible solution is found, $d$ is set to $1_{1\times n} q$ and IQCP (27) is solved. For a feasible solution of IQCP (27), the firing sequence $\sigma$ is given as the output. Otherwise the number of iterations $i$ is increased and $d_{\min}$ is set to the current length $d$. Then the next iteration of the while loop is started.

*Remark 4:* A general solution function for IQCP (27) that associate different costs to transitions and places can be found by introducing a transition cost vector $c_T$ as a $(1 \times n)$ vector and place cost vector $c_P$ as a $(1 \times m)$ vector and adapting the objective function for the total cost to $F(x) = \sum_{k=0}^{d-1} (c_T + c_P N^+) q(k)$. Similarly, for $\mathrm{ILP_d}$ (28), the possible length of the cost optimal firing sequence can be found by using the objective function $F(q) = (c_T + c_P N^+) q$.

**Algorithm 1:** Computation of firing sequence $\sigma$

**Input:** $PN,\ M_0,\ M_d,\ C$
**Initialize:** $d_{\min} = 0$, $i_{\max} = n$ , $i = 0$
**while** $i < i_{max}$ **do**
  Solve $\mathrm{ILP_d}$ (28)
  **if** *no feasible solution for ILP$_d$* (28) **then**
    Exit while
  Constraint transformation of constraints in $C$
  Solve IQCP (27) with $d = 1_{1\times n} q$
  **if** *feasible solution for IQCP* (27) **then**
    Exit while
  $i = i + 1$, $d_{\min} = d$
**if** *feasible solution for IQCP* (27) **then**
  Derive the firing sequence $\sigma = t_{q(0)} t_{q(1)} \ldots t_{q(d-1)}$
  **Output:** $\sigma$
**else**
  **Output:** Tracking not successful

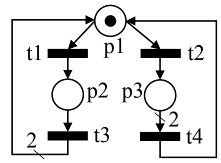


Fig. 1: Petri net for the performance comparison [10]

TABLE I: Computation time comparison for the PN in Fig. 1

| a | Computation time of the incremental approach (s) | Computation time of Algorithm 1 (s) | Sequence Length |
|---|---|---|---|
| 1 | 0.04 | 0.02 | 6 |
| 3 | 0.20 | 0.02 | 22 |
| 5 | 0.42 | 0.04 | 38 |
| 10 | 1.44 | 0.07 | 78 |
| 50 | X | 1.88 | 398 |
| 100 | X | 5.75 | 798 |

*Remark 5:* Most of the parameters used to determine the firing sequence can be changed at runtime. If the desired destination marking changes, Algorithm 1 is executed with the new destination marking $M_d^n$ and the current marking $M_0^c$ as the new initial marking. Changes in the constraints, caused by new requirements, can also be easily implemented by transforming the constraints in the right form as shown in Sections III and V. Structural changes of the PN, such as addtional places, transitions and arcs, are handled by executing Algorithm 1 with the new PN structure (i.e. changed $P, T, N^+$ and $N^-$). In comparison, the approach in [5] requires lengthy recomputation of the basis reachability graph which often can only be handled offline.

## C. Performance improvement of Algorithm 1

To illustrate the improvement of the two-step Algorithm 1 over the incremental approach (incrementally increase $d$), we compare the performance based on the PN shown in Fig. 1. We tested on a PC with a 2.3 Ghz processor and 8 GB RAM. Different destination markings are used, while initial marking is constant ($M_0 = (1\ 0\ 0)^T$). The destination marking is set by an integer $a$, i.e. $M_d = (a\ a\ a)^T$ and the maximal length for the incremental approach to $d_{\max} = 10n = 40$.

The results are given in Table I for 6 different integers $a$. The computation time of the two approaches and the sequence lengths of the resulting firing sequences are shown. Elements marked with "X" represent a computation time higher than 10 minutes. As can be seen from Table I, the computation time of Algorithm 1 is considerably smaller.

Two additional improvements of Algorithm 1 in comparison to the incremental approach can also be identified. For longer firing sequences, the upper bound $d_{\max}$ of the incremental approach is not sufficient, while the proposed Algorithm 1 requires no upper bound. Moreover, cost of places and transitions are difficult to handle with the incremental approach. Longer firing sequences can have lower cost and it is therefore unclear when to stop the computation. In comparison, the proposed Algorithm 1 can handle cost easily by modifying the objective functions, as shown in Remark 4.

## VII. EXAMPLE

In order to illustrate the proposed approach, we consider a grid-based warehouse system with multiple robots collecting products. The robots move on the grid in horizontal and vertical direction and collect products from the grid tiles and transporting them to a conveyor belt that moves the products out of the warehouse. The uncontrolled PN of a grid-based warehouse is shown in Fig. 2. The number of robots is denoted by $R = 2$. At the beginning all robots are idle and place $p_1$ is marked with $R$ tokens. The horizontal and vertical position of each robot is encoded by two places ($p_2,\ p_6$ horizontal, $p_3,\ p_7$ vertical). Firing of the adjacent transitions increases or decreases the number of tokens in these places and models the movement of the robots on the grid. The idle robots enter the grid at horizontal and vertical position 1 by firing of $t_1$ or $t_9$. Places $p_4$ and $p_8$ symbolizes that the robot does not transport a product, while $p_5$ and $p_9$ show that the robot is currently transporting a product. The horizontal and vertical position is denoted by $h/v$, while the maximal horizontal and vertical position is denoted by $H$ and $V$ (here $H = V = 5$ with conveyor belt at position $H/V$).

The control objective is that the robots move through the grid and transporting goods while adhering to constraints.

1) The movement of the robots is limited by the walls of the warehouse. The robots can not occupy positions with a horizontal or vertical position higher than $H$ and $V$ or lower than 1.
2) The warehouse roof is supported by one support column shown at position $3/3$.
3) In order to prevent collisions, only a single robot is allowed at any position in the grid.
4) A defect robot can block a tile of the grid.

For the sake of clarity, we denote the number of tokens in a place by $M(p_i), i = 1, \ldots, m$. For the maximal position of 1) the AND-GMEC $M(p_2) \leq 5 \wedge M(p_3) \leq 5$ is used. The analogue constraint for the minimal position would not allow the robots to leave the warehouse grid. Thus the AND-OR GMEC $(-M(p_2) \leq -1 \wedge -M(p_3) \leq -1) \vee (M(p_2) \leq 0 \wedge M(p_3) \leq 0)$ is used. The second constraints is handled by an OR-NOT GMEC $M(p_2) \neq 3 \vee M(p_3) \neq 3$. These two constraints can be analogously built for the second robot. The third constraint is also handled by a OR-NOT GMEC $M(p_2) \neq M(p_6) \vee M(p_3) \neq M(p_7)$. The last constraint has the same form as the second constraint, with the right side denoting the position of the defect robot. These constraints are transformed into suitable constraints for the ICQP (27) as shown in Sections III and V.

Consider the following process. Robot 1 moves to position $2/3$ and robot 2 to position $4/4$. Robot 2 breaks down at position $4/4$ and robot 1 loads a product and moves it to the conveyor belt. Executing Algorithm 1 results in the firing sequence $\sigma = t_1 t_2 t_4 t_4 t_9 t_{10} t_{10} t_{10} t_{12} t_{12} t_{12} t_6 t_4 t_4 t_2 t_2 t_2 t_7$ with length $d = 18$. The computation time is $t = 0.11$s.

## VIII. CONCLUSIONS

In this paper we have considered the tracking control of PNs with forbidden states. Based on a two-stage optimization approach, a firing sequence is found that steers the PN into a desired destination marking. A new type of constraints called NOT-GMEC has been introduced, wich allows a more

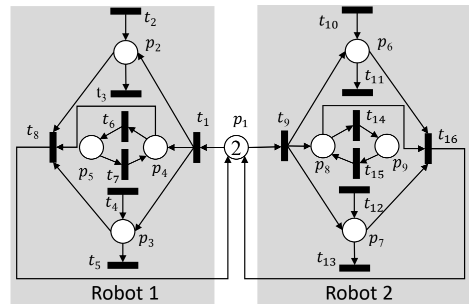

Fig. 2: Petri net of the warehouse with two robots

compact representation of forbidden place-token combinations. The proposed constraint tranformation approaches reduce the number of required constraints, especially for complex forbidden states.

For future research we consider systems containing uncontrollable and unobservable transitions. Partitioning the system into subsystems will be considered in future works, to reduce computational effort for large scale systems. Another question is how to take into account concurrent firing of transitions in different subsystems, while the firing sequence still fulfills the constraints placed on the whole system.

## REFERENCES


[1] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Springer, 2008.

[2] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion contraints on nets with uncontrollable transitions," in *Proc. of the 1992 IEEE Int. Conf. on Syst., Man, Cybern.*, 1992, pp. 974 – 979.

[3] A. Tarek and N. Lopez-Benitez, "Optimal legal firing sequence of petri nets using linear programming," *Optimization and Engineering*, vol. 5, no. 1, pp. 25–43, 2004.

[4] J. Li, M. Zhou, T. Guo, Y. Gan, and X. Dai, "Robust control reconfiguration of resource allocation systems with petri nets and integer programming," *Automatica*, vol. 50, no. 3, pp. 915–923, 2014.

[5] Z. Ma, Y. Tong, Z. Li, and A. Giua, "Basis marking representation of petri net reachability spaces and its application to the reachability problem," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1078–1093, 2017.

[6] D. Y. Lee and F. DiCesare, "Scheduling flexible manufacturing systems using petri nets and heuristic search," *IEEE Trans. Robot. Autom.*, vol. 10, no. 2, pp. 123–132, 1994.

[7] H. Lei, K. Xing, L. Han, F. Xiong, and Z. Ge, "Deadlock-free scheduling for flexible manufacturing systems using petri nets and heuristic search," *Computers & Industrial Engineering*, vol. 72, pp. 297–305, 2014.

[8] K. Xing, L. Han, M. Zhou, and F. Wang, "Deadlock-free genetic scheduling algorithm for automated manufacturing systems based on deadlock control policy," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 603–615, 2012.

[9] Q. Wang and Z. Wang, "Hybrid heuristic search based on petri net for FMS scheduling," *Energy Procedia*, vol. 17, pp. 506–512, 2012.

[10] D. Lefebvre and E. Leclercq, "Control design for trajectory tracking with untimed petri nets," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 1921–1926, 2015.

[11] D. Lefebvre, "Deadlock-free scheduling for flexible manufacturing systems using untimed petri nets and model predictive control," *Proc. of the 8th IFAC Conf. on Manufacturing Modelling, Management and Control*, vol. 49, no. 12, pp. 384–389, 2016.

[12] Z. Ma, Z. Li, and A. Giua, "Design of optimal petri net controllers for disjunctive generalized mutual exclusion constraints," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 1774–1785, 2015.

[13] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541 – 580, 1989.

[14] S. M. Fallat and C. R. Johnson, "Hadamard powers and totally positive matrices," *Linear Algebra and its Applications*, vol. 423, no. 2-3, pp. 420–427, 2007.

[15] H. P. Williams, *Model Building in Mathematical Programming*. Wiley, 2013.