Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research Article

Safety analysis of discrete event systems using a simplified Petri net controller

Meysam Zareiee *, Abbas Dideban, Ali Asghar Orouji

Electrical Engineering Department, Semnan University, Semnan, Iran

ABSTRACT

This paper deals with the problem of forbidden states in discrete event systems based on Petri net models. So, a method is presented to prevent the system from entering these states by constructing a small number of generalized mutual exclusion constraints. This goal is achieved by solving three types of Integer Linear Programming problems. The problems are designed to verify the constraints that some of them are related to verifying authorized states and the others are related to avoiding forbidden states. The obtained constraints can be enforced on the system using a small number of control places. Moreover, the number of arcs related to these places is small, and the controller after connecting them is maximally permissive.

© 2013 ISA. Published by Elsevier Ltd. All rights reserved.

1. Introduction

Discrete event systems (DESs) work based on changing states by occurring events [1]. Supervisory control is a theory which wants to restrict the behavior of the system for obtaining desired function [2,3]. The restriction can be performed by disabling some events in special conditions [4]. DESs can be modeled by Petri net (PN) where its compact structure, modeling power and mathematical properties have made it suitable for modeling this kind of systems [5,6]. Moreover, the PN can also model a large range of systems such as discrete, continuous and hybrid ones [7,8].

In DESs, there are some states which are called forbidden states and the system should be prevented from entering them. The reachable states without forbidden states are called authorized states. In recent years, a lot of researches have been accomplished for avoiding the forbidden states. Specifically, in flexible manufacturing systems (FMS) where deadlocks are major problems, a lot of methods based on PN models have been proposed to deal with deadlocks [9–16]. Some of them generate control places to prevent the system from entering the deadlock states. Particularly, many researchers construct generalized mutual exclusion constraints (GMEC) and enforce them on the system to satisfy a safety specification that specifies which evolutions of the system should not be allowed. However, achieving maximally permissive behavior after this enforcement is important. It means that all the

authorized states should be reachable and all the forbidden states must be avoided. Giua et al., [17] have proposed a method for assigning GMECs to forbidden states in safe PNs which is developed in [18] and [19] for non safe PNs. Also, region theory is a useful method for generation of GMECs [20]. GMECs can be enforced on the system using control places [21]. When the number of GMECs is large, a large number of control places should be added to the system which leads to a complicated model. However, the number of control places can be reduced by considering PN structural properties [22–28]. In all the above methods, the conjunctions of the GMECs are enforced on the system, but, when the set of authorized states is nonconvex, the disjunctions of constraints can be enforced on the system [29].

In this paper, the aim is to develop the method in [25] for obtaining a small number of control places with small number of arcs in smaller time. For this reason, three types of Integer Linear Programming (ILP) problems are solved to classify the forbidden states in small number of sets where for each one of the sets, a GMEC is assigned. The first type problems try to classify the forbidden states in a small number of sets. For each one of these sets, a GMEC can be assigned but the number of arcs related to the control places may be large (in this step the number of control places is only reduced). So, the second type of ILP problems is designed to change the sets of forbidden states and obtain new sets. This leads to reducing the number of arcs of control places. At the end, by solving the third type of ILP problems, a GMEC is assigned to each one of the new sets. Enforcing these GMECs on the system leads to a maximally permissive controller with small numbers of control places and arcs. So, the structural complexity of the controller is reduced. Moreover, the hardware and software costs for implementing the controller

* Corresponding author. Tel.: +98 231 3354123; fax: +98 231 3366997.
  E-mail addresses: mzareiee@semnan.ac.ir, meisamzareiee@gmail.com
(M. Zareiee), adideban@semnan.ac.ir (A. Dideban), aliaorouji@ieee.org
(A. Asghar Orouji).

may be reduced. At the end, to show the advantages of the new method, some examples are introduced.

The rest of this paper is as follows. In Section 2, some important and basic concepts are introduced. The new method is explained in Section 3. In Section 4, experimental results are considered. Finally, conclusions are presented in Section 5.

2. Preliminary presentation

In this section, basic concepts and important definitions are presented which will be used later. It is supposed that the reader is familiar with the PNs basis [30], and the theory of supervisory control [2,3,31].

2.1. Petri nets

A PN is represented by a quadruplet $R=\{P, T, W, M_0\}$ where P is the set of places, T is the set of transitions, W is the incidence matrix and M_0 is the initial marking. Each marking of the PN can be shown by a vector as follows:

$$M^T = [m_1 m_2 m_3 \ldots m_n] \tag{1}$$

where, m_i is the number of tokens in place p_i and n is the number of places. M_R denotes the set of all reachable markings and is divided into two subsets: the set of authorized states M_A and the set of forbidden states M_F. M_F is separated into two groups: (1) the set of reachable states (M'_F) which either do not respect the specifications or are deadlock states. (2) The set of states for which the occurrence of uncontrollable events leads to the states in M'_F. The set of reachable states without forbidden states is the set of authorized states.

2.2. GMECs and enforcing them on the system using control places

GMECs are the constraints that restrict the weight sum of tokens in some places. The constraints can be assigned to forbidden states to prevent the system from entering these states [17–19]. Control places can be connected to the system for enforcing GMECs on the system. In this case, for each GMEC, a control place is added to the system. To explain how it is possible to calculate the control places, suppose that the incidence matrix and the initial marking of the system are W_P and M_{P0} respectively. The set of GMECs is considered as $L \times M_P \leq b$ where M_P is the marking vector, L is a $n_c \times n$ matrix, b is a $n_c \times 1$ vector, n_c is the number of GMECs and n is the number of places. For each GMEC, a row is added to W_P. These rows are considered in matrix W_c and are calculated as follows [21]:

$$W_c = -LW_P \tag{2}$$

So, the incidence matrix of the system after connecting the control places is in the following form:

$$W = \begin{bmatrix} W_P \\ W_c \end{bmatrix} \tag{3}$$

The initial marking of the control places are calculated as follows:

$$M_{c0} = b - LM_{P0} \tag{4}$$

Therefore, the initial marking of the controlled system is in the following form:

$$M_0 = \begin{bmatrix} M_{P0} \\ M_{c0} \end{bmatrix} \tag{5}$$

The set of places in a PN model of an FMS is classified into three groups: Idle, Operation and Resource places, respectively. To calculate the set of GMECs (control places), the markings of operation places

should be only considered [13]. This concept leads to reducing the numbers of states that should be verified or forbidden by the controller [19] which simplifies the computations for constructing the GMECs. The reduced sets of authorized and forbidden states are denoted as M_{C-A} and M_{O-F}, respectively.

When the number of GMECs is large, a large number of control places should be added to the system which complicates the model. In the next section, a method is proposed for obtaining a small number of control places with small number of arcs which is maximally permissive.

3. New approach for obtaining a small number of control places with small number of arcs

In this section, the objective is to obtain a small number of simple GMECs which enforcing them on the system leads to obtaining a small number of control places and small number of related arcs. So, the objective is to modify the method in [25]. To do this, at first step we consider a set of safe constraints (with unknown variables) where each one of these constraints are for verifying an authorized state, and also a set of unsafe constraints (with unknown variables) at which each one of these constraints is for avoiding one of the forbidden states. Verifying all the safe constraints leads to verifying all the authorized states and verifying each one of the unsafe constraints leads to avoiding the related forbidden state. Then, we solve an ILP problem to obtain the unknown variables by verifying all the safe constraints and the largest number of unsafe constraints and we save the answer in a set like W_1. Next, the verified unsafe constraints should be eliminated from the set of unsafe constraints and should be saved in a new set (for example we call this set as R_1). If the set of unsafe constraints is not empty, we repeat this step again for the remaining unsafe constraints and save the answer in a set like W_2 that verify all the safe constraints and the largest number of remaining unsafe constraints. The new verified unsafe constraints should be eliminated from the set of unsafe constraints and must be considered in a new set (we call this set as R_2). Then, we solve another ILP problem which verifies all the safe constraints and all the unsafe constraints in the set R_2 and the largest number of unsafe constraints in R_1 and replace this answer by the answer in W_2 (in this ILP problem a constraint is added that do not permit the right side of the obtained GMEC increase more than before. For example suppose that the obtained GMEC in this step should be in this form: $k_1 + k_2 + \ldots + k_n \leq x$, and the number in the right side of the obtained GMEC in the last step is 5. So, the constraints $x \leq 5$ is added to the ILP problem. This constraint can be lead to reducing the number of arcs and their weighs). The verified unsafe constraints should be eliminated from R_1 and should be added to R_2. If the set of unsafe constraints is not empty, we do these steps for the remaining unsafe states (in this case, if we are in step t, we consider $R_1 \cup R_2 \cup \ldots \cup R_{t-1}$ instead of R_1). When the set of unsafe constraints is empty, for each one of the sets $R_1, R_2, \ldots, R_{t-1}$ (by considering that this is repeated t times), other ILP problems should be solved to verify all the safe constraints and all the constraints in R_e ($e=1, 2, \ldots, t-1$) and replace the answer in W_e ($e=1, 2, \ldots, t-1$). This concept is formalized and generalized in Algorithm 1.

Algorithm 1. Obtaining a small number of control places with small number of arcs

Input: The set of authorized states $M_A=\{[z_{11} z_{12} \ldots z_{1n}], \ldots, [z_{r1} z_{r2} \ldots z_{rn}]\}$ and the set of forbidden states $M_F=\{[B_{11} B_{12} \ldots B_{1n}], \ldots, [B_{t1} B_{t2} \ldots B_{yn}]\}$.

Output: The small number of control places with small number of arcs.

Default: t is a variable and suppose that $t=0$, and $R_t=\varnothing$, $W_t=\varnothing$, $W_t^k=\varnothing$ ($\forall t, k$), and $R_t^k=\varnothing$ ($\forall t, k$) are some sets.

Step 1. Consider a generic constraint as follows:

$$k_1 m_1 + k_2 m_2 + \ldots + k_n m_n \leq x \tag{6}$$

where m_i is the number of tokens in place p_i.

Step 2. Substitute the markings of the authorized states in the constraint (6) and consider the obtained constraints as follows:

$$\sum_{i=1}^{n} z_{j,i} k_i \leq x \quad j=1, 2, \ldots, r \tag{7}$$

which are called safe constraints.

Step 3. Substitute the markings of the forbidden states in constraint (6) and convert the smaller equal sign to greater sign. Consider the obtained constraints as the following form:

$$\sum_{i=1}^{n} B_{l,i} k_i > x \quad l=1, 2, \ldots, y \tag{8}$$

which are called unsafe constraints. The set of unsafe constraints is denoted as H.

Step 4. $t=t+1$

Step 5. Solve the following ILP problem and obtain the constants x and k_i (for $i=1, \ldots, n$) which verify all the safe constraints and the largest number of unsafe constraints in H (this step is described in Remark 3):

$$\min F = \sum_{l \in N_H} f_l \tag{9}$$

Subject to

$$\sum_{i=1}^{n} z_{j,i} k_i - x \leq 0 \quad j=1, 2, \ldots, r \tag{10}$$

$$\sum_{i=1}^{n} B_{l,i} k_i - x > -Q \times f_l \quad \forall l \in N_H \tag{11}$$

$$f_l \in \{0, 1\} \tag{12}$$

where Q is a positive constant that should be considered large enough and N_H denotes

$$\left\{ l \Big| \left(\sum_{i=1}^{n} B_{l,i} k_i > x \right) \in H \right\}$$

Step 6. Save the obtained constants in the set W_t and then remove the verified unsafe constraints from H and substitute them in the set R_t (if $f_l=0$ in the ILP problem in step 5, the unsafe constraint number l is verified, else it is not verified).

Step 7. $x_t = x$ (the obtained x).

Step 8. If $t > 1$,
Solve the following ILP problem and obtain the constants x and k_i (for $i=1, \ldots, n$) which verify all the safe constraints and all the constraints in the set R_t, and the biggest number of unsafe constraints in the set $R_1 \cup R_2 \cup \ldots \cup R_{t-1}$ (this is described in Remark 4):

$$\min F = \sum_{l \in (N_R^1 \cup N_R^2 \cup \ldots \cup N_R^{t-1})} f_l \tag{13}$$

Subject to

$$\sum_{i=1}^{n} z_{j,i} k_i - x \leq 0 \quad j=1, 2, \ldots, r \tag{14}$$

$$\sum_{i=1}^{n} B_{l,i} k_i - x > 0 \quad \forall l \in N_R^t \tag{15}$$

$$\sum_{i=1}^{n} B_{l,i} k_i - x > -Q \times f_l \quad \forall l \in (N_R^1 \cup N_R^2 \cup \ldots \cup N_R^{t-1}) \tag{16}$$

$$x \leq x_t \tag{17}$$

$$f_l \in \{0, 1\} \tag{18}$$

where Q is a positive constants that should be considered large enough and N_R^q denotes

$$\left\{ l \Big| \left(\sum_{i=1}^{n} B_{l,i} k_i > x \right) \in R_q \right\}$$

Replace the obtained answer with the answer in W_t.
Add the verified unsafe constraints to the set R_t.
Remove the verified unsafe constraints from the sets R_1, R_2, … R_{t-1}.

Step 9. If the set of unsafe constraints is not empty, go to step 4.

Step 10. If $t > 1$, Solve the following ILP problem for each one of the sets R_e ($e=1, 2, \ldots, t-1$) and replace the new answers with the answers in the sets W_1, W_2, …, W_{t-1} respectively (this is described in Remark 5).

$$\min X = x \tag{19}$$

Subject to

$$\sum_{i=1}^{n} z_{j,i} k_i - x \leq 0 \quad j=1, 2, \ldots, r \tag{20}$$

$$\sum_{i=1}^{n} B_{l,i} k_i - x > 0 \quad \forall l \in N_R^e \tag{21}$$

where N_R^e denotes

$$\left\{ l \Big| \left(\sum_{i=1}^{n} B_{l,i} k_i > x \right) \in R_e \right\}$$

Step 11. Substitute the answers of the sets W_1, W_2, …, W_t in the constraint (6). These constraints are the small number of GMECs which enforcing them on the system leads to obtaining a maximally permissive controller with small numbers of control places and arcs.

Remark 1. In Algorithm 1, step 2 is considered because if the constraints in this step are verified by the controller, all the authorized states are reachable.

Remark 2. In Algorithm 1, step 3 is considered because verifying each one of the constraints in this step leads to avoiding the related forbidden state.

Remark 3. The ILP problem in step 5 of Algorithm 1 is considered to obtain x and k_i which verify all the safe constraints and the largest number of unsafe constraints in H. In this problem, the relation (10) shows verifying all the safe constraints by x and k_i's. f_l's (for $l=1, 2, \ldots$) represent the relation between lth unsafe constraint and the obtained constants x and k_i's. $f_l=0$ means that lth unsafe constraint in H is verified by the obtained constants and $f_l=1$ means that lth unsafe constraint is not verified by these constants. So, the relation (11) is considered for verifying the largest number of unsafe constraints. Therefore, the objective function is considered as.

$$\min F = \sum_{l \in N_H} f_l$$

Remark 4. An ILP problem similar to step 5 is considered in step 8 of Algorithm 1 for finding the constants x and k_i (for $i=1, \ldots, n$) which verify all the safe constraints and all the constraints in the set R_t, and the biggest number of unsafe constraints in the previous sets (R_1, R_2, … R_{t-1}). In this problem, the relation (15) is considered to verify all the unsafe constraints in R_t. The relation (16) is

considered for verifying the largest number of constraints in the set $R_1 \cup R_2 \cup \dots \cup R_{t-1}$. In this step, the goal is to add the largest number of unsafe constraints in the previous sets (R_1, R_2, … R_{t-1}) to the set R_t. But, adding these constraints may lead to increasing the value of constant x that may lead to increasing the number of arcs related to control places. So, the relation (17) is considered to prevent increasing the value of x. Adding the largest unsafe constraints from previous sets to R_t by the same x reduces the number of unsafe constraints in previous sets which leads to obtaining simpler GMECs for these sets (also some of these sets may get empty). Moreover, the numbers of arcs related to these GMECs may reduce. By fixing x, the number of arcs related to the GMEC for R_t (after adding the unsafe constraints) can be fixed.

Remark 5. In step 10 of Algorithm 1, an ILP problem is considered to find new answers for the new sets R_1, R_2, … R_{t-1}.

Algorithm 1 is a good method for obtaining a small number of control places with small number of related arcs which connecting them to the system leads to obtaining maximally permissive behavior. By using this method, it is possible to generate a small number of GMECs in a lot of kinds of systems. The most important result in this method is the number of arcs related to the control places which is small. Moreover, the small number of arcs can lead to the small implementation costs [32].

4. Experimental results

In this section, some FMS examples are considered to show the experimental results of the proposed method. In all examples, the forbidden states are deadlock states or the states that lead to deadlocks. Moreover, the results are compared with some other methods in Tables 1–3. In these tables, N_{CP}, N_{arc} and N_{RS} are the numbers of control places, arcs and reachable states, respectively.

Consider the PN model of the FMS in Fig. 1 which is taken from [19]. This system consists of 19 places and 14 transitions. The sets of idle, resource and operation places are $P_0 = \{p_1, p_8\}$, $P_R = \{p_{14} -$

$p_{19}\}$ and $P_A = \{p_2 - p_7, p_9 - p_{13}\}$, respectively. It has 282 reachable states where 205 ones are authorized and 77 ones are forbidden states. The numbers of states in the sets M_{C-A} and M_{O-F} are 26 and 8, respectively. To prevent the system from entering the forbidden states, Algorithm 1 is applied to it for generating a small number of GMECs. By applying this method, two GMECs are obtained as follows:

$$m_2 + 2m_3 + m_4 + 2m_5 + 2m_6 + 3m_9 + 3m_{10} \leq 9 \quad (22)$$

$$m_2 + 2m_3 + m_4 + 2m_{11} + 2m_{12} \leq 3 \quad (23)$$

Enforcing these two GMECs on the system prevents it from entering the forbidden states. The incidence matrix and the initial tokens related to these GMECs are respectively as follows:

$$W_c = \begin{bmatrix} -1 & -1 & 0 & -1 & 0 & 0 & 2 & 0 & -3 & 0 & 3 & 0 & 0 & 0 \\ -1 & -1 & 0 & 1 & 2 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 2 & 0 \end{bmatrix},$$

$$M_{c0} = \begin{bmatrix} 9 \\ 3 \end{bmatrix} \quad (24)$$

As it is obvious from the incidence matrix, the numbers of control places and the related arcs are efficient. In this example, two control places with 12 arcs are obtained. So, by using Algorithm 1, it is possible to obtain a small number of control places with small number of arcs. The results are compared with some conventional methods in Table 1.

Now, consider the FMS in Fig. 2 taken from [15]. This system contains 19 places and 14 transitions. It has 168 reachable states at which 96 ones are authorized and 72 ones are forbidden states. The sets M_{C-A} and M_{O-F} have 13 and 11 states, respectively. After applying Algorithm 1, 4 GMECs are obtained in the following forms:

$$3m_3 + 3m_4 + m_6 + m_7 + 5m_{12} \leq 5 \quad (25)$$

$$4m_7 + m_{11} + m_{13} + 2m_{15} + 2m_{16} \leq 5 \quad (26)$$

$$m_3 + m_4 + m_6 + 2m_{11} + m_{12} + 2m_{13} \leq 4 \quad (27)$$

$$m_6 + m_{15} \leq 1 \quad (28)$$

The incidence matrix and initial tokens of these GMECs are computed as follows:

$$W_c = \begin{bmatrix} 0 & 3 & 0 & -2 & 0 & -1 & 0 & -5 & 0 & 5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & -4 & -1 & 1 & 0 & -2 & -1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 & -2 & 1 & 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 0 & 0 \end{bmatrix}$$

Table 1
The results of some methods on the system in Fig. 1.

The methods	[14]	[12]	[19]	[28]	[23]	[24]	[26]	The proposed method
N_{CP}	9	6	8	3	2	2	2	2
N_{arc}	42	32	37	21	12	15	15	12
N_{RS}	205	205	205	205	205	205	205	205

Table 2
The results of some methods on the system in Fig. 2.

The methods	[15]	[19]	[28]	[24]	[26]	The proposed method
N_{CP}	9	11	6	4	4	4
N_{arc}	44	48	43	26	26	23
N_{RS}	96	96	96	96	96	96

Table 3
The results of some methods on the system in Fig. 3.

The method	[11]	[13]	[19]	[28]	[23]	[24]	[26]	The proposed method
N_{CP}	16	19	17	10	5	6	6	6
N_{arc}	88	112	101	89	55	45	45	45
N_{RS}	12656	21562	21581	21581	21581	21581	21581	21581

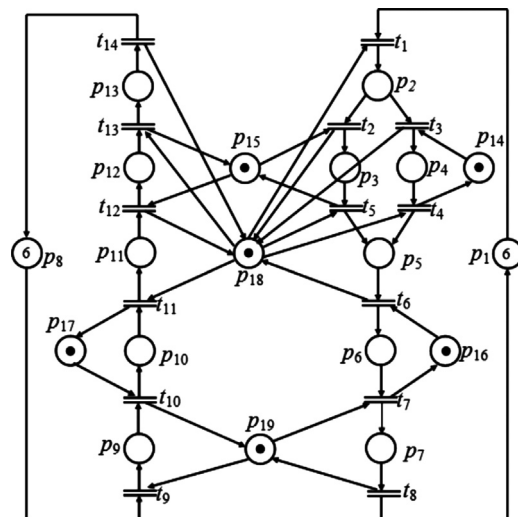**Fig. 1.** The FMS with 282 reachable states.
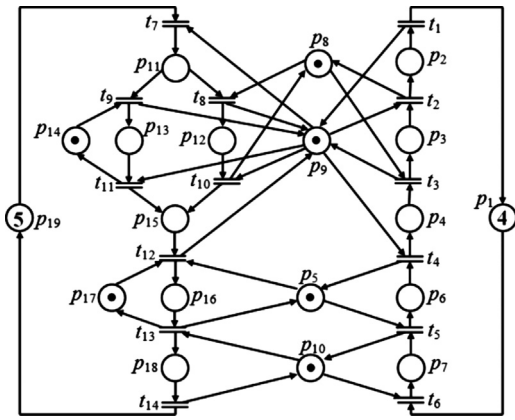
Fig. 2. An FMS with 168 reachable states.

Fig. 3. The FMS with 26750 reachable states.

$$M_{c0} = \begin{bmatrix} 5 \\ 5 \\ 4 \\ 1 \end{bmatrix} \qquad (29)$$

Hence, 4 control places with 23 arcs are obtained. The results are compared with some methods in Table 2.

Finally, consider the PN model of another FMS in Fig. 3, taken from [19]. This model contains 26 places and 20 transitions. There are 26750 reachable states where 21581 ones are authorized states and 5169 ones are forbidden states. The sets M_{C-A} and M_{O-F} have 393 and 34 states, respectively. To prevent the system from entering the forbidden states, Algorithm 1 generates 6 GMECs as follows:

$$15m_2 + 15m_3 + 3m_6 + 3m_7 + 17m_8 + 17m_9$$
$$+ 3m_{11} + 47m_{12} + 50m_{13} + 50m_{15}$$
$$+ 50m_{16} + m_{17} + 2m_{18} \leq 196 \qquad (30)$$

$$m_2 + m_3 + m_8 + m_9 + 27m_{11}$$
$$+ 3m_{12} + 3m_{13} + 6m_{15} + 9m_{16} + 18m_{17} \leq 71 \qquad (31)$$

$$6m_6 + 6m_7 + m_9 + 6m_{11} + m_{12} + m_{15}$$
$$+ m_{16} + 5m_{17} + 6m_{18} \leq 34 \qquad (32)$$

$$m_2 + m_3 + m_8 \leq 2 \qquad (33)$$

$$m_{13} + m_{15} \leq 2 \qquad (34)$$

$$m_{12} + m_{16} \leq 2 \qquad (35)$$

The incidence matrix and initial tokens of these GMECs are respectively calculated as follows:

So, 6 control places with 45 arcs are obtained. The results are compared with some methods in Table 3.

The methods in [24,26] reduce the number of control places and are simple in the case of computational complexity. But, the number of their arcs can be reduced. The method in [23] can generate the small numbers of control places but it has the problem of computational complexity. As an instance, for the system in Fig. 3, it should solve an ILP problem with 15640 constraints and 1700
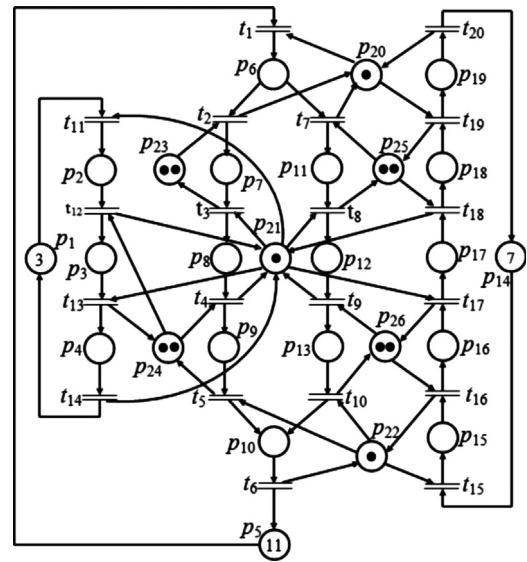
variables which takes long time while the maximum numbers of constraints and variables in the ILP problems of the proposed method for this system are 427 and 51, respectively. For the system in Fig. 3, the proposed method generates 6 control places which is only one greater than the obtained number in [23]. On the other hand, the proposed method obtains 45 arcs which is 10 less than the obtained number in [23]. In this example the proposed method and the methods in [24,26] obtain the same answer. For the systems in Figs. 1 and 2, the proposed method and the methods in [24,26] obtain the same number of control places but the number of arcs in the proposed method is three less than the ones obtained in [24,26]. Reducing the number of arcs may lead to reducing the implementation costs of the controller. The costs include introduction of sensors and actuators, the connection links, software modifications, etc., which are affected by the number of arcs between the control places and the transitions [32]. For the arcs from the transitions to control places and from the control places to transitions, the sensors and actuators for detecting and disabling the transitions should be installed, respectively, if they are required. For both kinds of arcs going from or coming to control places, the connection links and software modifications should be considered [32]. So, when the number of these arcs is reduced, the implementation costs may be reduced. Different costs of sensors, actuators and connection links lead to different implementation costs. So, if we consider the same cost for these parameters in all transitions, it is possible to compare the implementation costs of the methods in Tables 1–3 by considering the number of arcs. In this case, the ratios between the numbers of the arcs indicate the ratios between the implementation costs of the methods in these tables. We should mention that by changing the costs of sensors, actuators, connection links and, etc, the ratios may change.

By using Algorithm 1, it is possible to obtain a small number of control places with small number of arcs. So, the structural

$$W_c = \begin{bmatrix} -3 & 0 & -14 & 0 & 17 & 0 & 0 & -44 & -3 & 50 & -15 & 0 & 15 & 0 & -50 & 0 & 49 & -1 & 2 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 & -27 & 24 & 0 & 3 & -1 & 0 & 1 & 0 & -6 & -3 & -9 & 18 & 0 & 0 \\ -6 & 0 & 6 & -1 & 1 & 0 & 0 & 5 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -4 & -1 & 6 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 \end{bmatrix}, \quad M_{c0} = \begin{bmatrix} 196 \\ 71 \\ 34 \\ 2 \\ 2 \\ 2 \end{bmatrix} \qquad (36)$$

complexity of the PN model is reduced. Another advantage of this method is its computational complexity which is rather simple. We solve a few number of ILP problems and each ILP problem in each step is simpler than the problem in previous step. Because when some unsafe constraints are verified, they are eliminated from the set of unsafe constraints and the number of binary variables (f_l) decreases.

5. Conclusion

This paper presents a method for obtaining a small number of control places with small number of arcs. To achieve this goal, the authorized and forbidden states are given as input. Then, some constraints with unknown variables are assigned to authorized states (which are called safe constraints) and the other constraints with unknown variables are assigned to forbidden states (which are called unsafe constraints). Verifying the safe constraints leads to verifying the authorized states and verifying each one of the unsafe constraints leads to avoiding the related forbidden state. So, three types of ILP problems are solved to obtain a small number of groups (and also the unknown variables) at which each group contains some unsafe constraints that can be verified by the obtained variables (the obtained variables verify all the safe constraints). Then, for each group, a GMEC is assigned. Enforcing the obtained GMECs on the system using control places causes a simple and maximally permissive controller.

References

[1] Cassandras CG, Lafortune S. Introduction to discrete event systems. USA: Springer; 2008.
[2] Ramadge PJ, Wonham WM. Modular feedback logic for discrete event systems. SIAM Journal on Control and Optimization 1987;25:1202–18.
[3] Ramadge PJ, Wonham WM. The control of discrete event systems, dynamics of discrete event systems. Proceedings of the IEEE 1989;77:81–98.
[4] Giua A., Petri nets as discrete event models for supervisory control, Ph.D. thesis, Troy, New York, 1992.
[5] Krogh BH, Holloway LE. Synthesis of feedback control logic for discrete manufacturing systems. Automatica 1991;27:641–51.
[6] Moody JO, Antsaklis P. Petri net supervisor for DES with uncontrollable and unobservable transition. IEEE Transactions on Automatic Control 2000;45: 462–76.
[7] Renganathan K, Bhaskar V. An observer based approach for achieving fault diagnosis and fault tolerant control of systems modeled as hybrid Petri nets. ISA Transactions 2011;50:443–53.
[8] Garg H. Reliability analysis of repairable systems using Petri nets and vague Lambda–Tau methodology. ISA Transactions 2013;52:6–18.
[9] Visvanadham N, Nahari Y, Johnson TL. Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models. IEEE Robotics and Automation 1990;6:713–23.
[10] Abdallah IB, El-Maraghy HA. Deadlock prevention and avoidance in FMS: A Petri net based approach. International Journal of Advanced Manufacturing Technology 1998;14:704–15.
[11] Huang YS, Jeng MD, Xie XL, Chung SL. Deadlock prevention based on Petri nets and siphons. International Journal of Production Research 2001;39:283–305.
[12] Uzam M. An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions. International Journal of Advanced Manufacturing Technology 2002;19:192–208.
[13] Uzam M, Zhou MC. An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems. International Journal of Production Research 2006;44:1987–2030.
[14] Li ZW, Zhou MC, Jeng MD. A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions. IEEE Transactions on Automation Science and Engineering 2008;5:182–8.
[15] Guo J, Li ZW. An improved algorithm for supervisor design for FMS using the theory of regions. In: Proceedings of the IEEE international conference on mechatronics and automation; 2010.
[16] Lefebvre D, Leclercq E. Stochastic Petri net identification for the fault detection and isolation of discrete event systems. IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans 2011;41:213–25.
[17] Giua A, DiCesare FM, Silva M. Generalized mutual exclusion constraints on nets with uncontrollable transitions. In: Proceedings of the IEEE international conference on IEEE transactions on systems, man, and cybernetics; 1992 pp. 974–979.
[18] Zareiee M, Dideban A, Nazemzadeh P. From forbidden states to linear constraints. In: Proceedings of the international conference on modeling and simulation; 2011.
[19] Chen Y, Li ZW, Khalgui M, Mosbahi O. Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems. IEEE Transactions on Automation Science and Engineering 2011;8:374–93.
[20] Ghaffari A, Rezg N, Xie XL. Design of live and maximally permissive Petri net controller using the theory of regions. IEEE Transactions on Robotics and Automation 2003;19:137–42.
[21] Yamalidou K, Moody J, Lemmon M, Antsaklis P. Feedback control of Petri nets based on place invariants. Automatica 1996;32:15–28.
[22] Dideban A, Alla H. Reduction of constraints for controller synthesis based on safe Petri nets. Automatica 2008;44:1697–706.
[23] Chen YF, Li ZW. Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems. Automatica 2011;47:1028–34.
[24] Nazeem A, Reveliotis S, Wang Y, Lafortune S. Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory: the linear case. IEEE Transactions on Automatic Control 2011;56:1818–33.
[25] Zareiee M, Dideban A, Orouji AA, Alla H. A simple petri net controller in discrete event systems. In: Proceedings of information control problems in manufacturing, 2012, vol. 14; pp. 188–93.
[26] Chen YF, Li ZW, Zhou MC. Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems. IEEE Transactions on Systems, Man, and Cybernetics, Part A 2012;42:615–29.
[27] Dideban A, Zareiee M, Alla H. Controller synthesis with highly simplified linear constraints. Asian Journal of Control 2013;15:80–94.
[28] Zareiee M, Dideban A, Orouji AA, Soleymanpour HR. Simplification of a Petri Net controller in industrial systems by using an optimization algorithm. International Journal of Industrial Engineering and Production Research 2013;24:59–70.
[29] Chen Y, Li ZW, Zhou MC. Optimal supervisory control of flexible manufacturing systems by Petri nets: a set classification approach. IEEE Transactions on Automation Science and Engineering (In press).
[30] David R, Alla H. Discrete, continuous, and hybrid Petri nets. Berlin, Germany: Springer; 2005.
[31] Kumar R, Holloway LE. Supervisory control of deterministic Petri nets with regular specification languages. IEEE Transactions on Automatic Control 1996;41:245–9.
[32] Basile F, Chiacchio P, Giua A. An optimization approach to Petri net monitor design. IEEE Transactions on Automatic Control 2007;52:306–11.