



# Simplification of a Petri Net Controller in Industrial Systems by Using an Optimization Algorithm

M. Zareiee, A. Dideban<sup>\*</sup>, A. A. Orouji & H. R. Soleymanpour

*Meysam Zareiee is a Ph.D. student at the Faculty of electrical and computer engineering, Semnan University, Semnan, IRAN*

*Abbas Dideban is an assistant professor at the Faculty of electrical and computer engineering, Semnan University, Semnan, IRAN*

*Ali asghar Orouji is an associate professor at the Faculty of electrical and computer engineering, Semnan University, Semnan, IRAN*

*Hassan rezaei soleymanpour was a Msc student at the Faculty of electrical and computer engineering, Semnan University, Semnan, IRAN*

## KEYWORDS

Discrete Event System (DES),  
Supervisory Control Theory,  
Petri Net (PN),  
Generalized Mutual  
Exclusion constraint (GMEC),  
Flexible Manufacturing  
System (FMS)

## ABSTRACT

*This paper deals with the problem of forbidden states in discrete event systems modeled by Petri Net. To avoid the forbidden states, some constraints which are called Generalized Mutual Exclusion Constraints can be assigned to them. Enforcing these constraints on the system can be performed using control places. However, when the number of these constraints is large, a large number of control places must be connected to the system which complicates the model of controller. In this paper, the objective is to propose a general method for reducing the number of the mentioned constraints and consequently the number of control places. This method is based on combining some constraints for obtaining a constraint verifying all of them which is performed using the optimization algorithms. The obtained controller after reducing the number of the control places is maximally permissive.*

© 2013 IUST Publication, IJIEPR, Vol. 24, No. 1, All Rights Reserved.

## 1. Introduction

Flexible manufacturing systems (FMS) are important among the set of discrete event systems (DES). An FMS consists of such resources like machines, robots and buffers. Crude parts are concurrently manufactured by the system sharing the limited number of resources. Sharing resources may lead to deadlock which is a highly undesirable situation in FMS. In deadlock states, each set of two or more jobs keeps waiting indefinitely for the other jobs in the set to release resources [1], [2]. Therefore, the system must be avoided from entering them. To evaluate FMSs and also DESs, Petri Net (PN) can be used as a suitable tool for modeling them [3]. This tool is composed of places, transitions, arcs and tokens. Each transition is corresponding to an event and the tokens in the places

represent the state of system. Specially, there are a lot of methods for preventing the system from entering the deadlock states based on PN models [1], [4]-[8]. These methods connect some control places to the system for the reversibility guaranty. But, the number of these places may be large and some of them may be redundant. Uzam et al. [2] have proposed a method for removing the redundant control places. But in this method for a system with  $n$  control places, the reachability graph must be calculated  $n$  times. This method is developed in [9] where  $n$  Integer linear programming (ILP) problems must be solved to remove the redundant control places. The advantage of these two methods is their generality for non safe PN, however the computation takes long time and they don't give the least number of control places.

Existence of uncontrollable transitions in the system may cause some other problems beside the deadlock states since these transitions cannot be disabled by the controller and the specification may be violated. So, the forbidden states can be divided into two sets: the states which are related to deadlock and the states that

<sup>\*</sup> Corresponding author: Abbas Dideban

Email: [adideban@semnan.ac.ir](mailto:adideban@semnan.ac.ir)

Paper first received Jan. 28, 2012, and in accepted form Jul. 07, 2012.

violate the specification. Supervisory control theory proposes a strategy for coping with forbidden states in DES [10], [11]. This theory tries to obtain the desired function of the system by restricting its behavior. This restriction can be performed by disabling the transitions in some special conditions [12], [13]. But, disabling the uncontrollable transitions is impossible. In this case before firing these transitions, some controllable transitions should be disabled which leads to reducing the state space of authorized states [14].

In recent years, a lot of methods have been proposed for avoiding forbidden states. In [15] and [16], the authors calculate some conditions to block the controllable transitions in some special situations. But in these methods, the dynamic of controller is not clear. Ghaffari et al. [17] have proposed a method for calculating the control places using regions theory. In this method some constraints are generated which some of them verify the authorized states and the others violate the forbidden states. Then, by solving some ILP problems, the control places are generated. But in this method the computational time for solving the ILP problems is large and also a large number of control places are generated generally. The problem of the large number of control places may be solved using the method in [18]. However, this method does not give the least number of control places.

Generalized Mutual Exclusion Constraints (GMECs) are the ones that verifying them may lead to complying with the specification and avoiding the forbidden states. These constraints can be enforced on the system using control places [19]. Giua et al. [20] have proposed a method for assigning a GMEC to each forbidden state in safe PNs. This method has been developed in our previous work [21] where the GMECs can be assigned to forbidden states in non safe PNs. Enforcing the GMECs related to forbidden states on the system prevents it from entering these states. But in these methods when the number of forbidden states and consequently the number of GMECs is large, a large number of control places are connected to the system. So, some methods have been proposed for reducing the number of GMECs.

In [22], the authors proposed a method in safe and conservative PNs for reducing the number of GMECs using invariant property. The conservative limitation is removed in the methods in [23] and [24] which use the over-state concept for the similar reduction. However, these methods are only applicable on safe PNs and do not give the least number of control places. In our previous work [25], we proposed a method in non safe PN for obtaining a control place to avoid all the forbidden states. This control place is obtained by solving an ILP problem. But this method is a special case and can generate an answer in the systems with small number of forbidden states.

In this paper the objective is to propose a general method for reducing the number of control places in safe and non safe PNs. To perform this reduction, some

constraints should be generated which some of them verify the authorized states and the others violate the forbidden states. Then, the constraints must be classified in some groups. Each group consists of all the constraints related to authorized states plus some of the constraints related to forbidden states. At the end, by solving an ILP problem for each group, the small number of GMECs will be obtained. Enforcing the obtained GMECs on the system leads to obtaining a maximally permissive behavior. Classifying the constraints must generate the small number of groups which can be performed using Modified Adaptive Particle Swarm Optimization (MAPSO) [26]. At the end of this paper, the new method is applied on some practical examples and the advantages of the new method comparing with the conventional methods are shown.

The rest of this paper is as follows. In section 2, the important concept for introducing the new method is presented. Our previous work which is the base of the new method is presented in section 3. The new method is proposed in section 4. Comparing the new method with the conventional methods is discussed in section 5. Finally in section 6 conclusions are presented.

## 2. Basic Concepts

In this section we introduce the basic concepts which are necessary for introducing the new method. We suppose that the reader is familiar with the PN basis [27].

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. Places are represented by circles and transitions are represented by bars. Places and transitions are connected together by arcs. Incidence matrix shows the relation between places, transitions and arcs. Places can be marked by tokens. PNs can be divided into two types: safe PNs and non safe PNs. In safe PNs, the number of tokens in each place is one or zero but in non safe PNs this number can be more than one. The marking of a PN at a given moment is a column vector whose  $i^{\text{th}}$  component is the marking of place  $P_i$  at this moment. For the sake of simplicity we write the markings in the transposed form as follows:

$$M^T = [m_1 \dots m_i \dots m_n] \quad (1)$$

where  $m_i$  is the number of tokens in place  $P_i$  and  $n$  is the number of places.

In a system,  $M_R$  denotes the set of all the reachable markings and can be divided into two subsets: the set of authorized states  $M_A$ , and the set of forbidden states  $M_F$ . The set of forbidden states is corresponding to two groups: a) the set of reachable states ( $M_F$ ) which violates the specification or are deadlock states, b) The set of reachable states at which the occurrence of uncontrollable events leads to entering the system into

the states in  $M_F$ . The set of reachable states without forbidden states is the set of authorized states.

In the set of forbidden states, there is a very important subset which is called the set of border forbidden states [14] and is defined as follows:

**Definition 1.**  $M_B$  is the set of border forbidden states and is defined as follows:

$$M_B = \{M_i \in M_F \mid \text{if } \exists M_j \in M_A, M_j \xrightarrow{t_k} M_i \Rightarrow t_k \in T_c\} \quad (2)$$

where  $T_c$  is the set of controllable transitions.  $\square$

The border forbidden states are the ones that forbidding them leads to forbidding all the forbidden states. So, for preventing the system from entering the forbidden states, it is sufficient to forbid the border forbidden states.

### 2.1. From GMECs to Control Places

GMECs are the constraints which restrict the weight sum of tokens in some places. Enforcing them on the system complies the specification and can avoid the forbidden states. For example suppose that places  $P_1$ ,  $P_2$  and  $P_3$  are corresponding to working states of machines 1, 2 and 3, respectively and the specification says that these three machines must not work at the same time. So the following GMEC complies the specification:

$$m_1 + m_2 + m_3 \leq 2$$

Enforcing GMECs on the system can be performed using control places [19]. In this case, instead of a GMEC, a control place should be added to the PN model of the system.

Calculating the control places can be performed as follows [19]:

Suppose that the set of GMECs is as the following form:

$$L.M_p \leq b \quad (3)$$

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. In this method for each GMEC, a place should be connected to the model. So, for each GMEC, a row should be added to the incidence matrix of the model. These rows are shown by  $W_c$  and can be calculated as follows:

$$W_c = -L.W_p \quad (4)$$

where  $W_p$  is the incidence matrix of the system before connecting the control places. This matrix ( $W_c$ ) should be added to  $W_p$ . So, the incidence matrix of the controlled model is as follows:

$$W = \begin{bmatrix} W_p \\ W_c \end{bmatrix} \quad (5)$$

The initial marking of the control places can be calculated as the following form:

$$M_{c_0} = b - L.M_{p_0} \quad (6)$$

where  $M_{p_0}$  is the initial marking of the model before connecting the control places. So, the initial marking of the controlled model is:

$$M_0 = \begin{bmatrix} M_{p_0} \\ M_{c_0} \end{bmatrix} \quad (7)$$

Using these calculations, the control places related to the GMECs can be enforced on the system.

When the number of GMECs is large, a large number of control places should be added to the PN model of the system. This concept complicates the controlled model. But, the number of GMECs can be reduced [20]. In the next section, our previous work [25] for obtaining a GMEC corresponding to all forbidden states is recalled. This is the base of our new method in this paper.

### 3. One GMEC for Avoiding all Forbidden States

In this section the goal is to recall our previous work [25] for solving the problem of forbidden states by obtaining one GMEC for preventing all forbidden states. This method is a special case and can be applicable on the system having small number of forbidden states. To recall this method, consider a generic constraint as follows:

$$k_1 m_1 + k_2 m_2 + \dots + k_n m_n \leq x \quad (8)$$

where  $x$  and  $k_i$  for  $i=1, \dots, n$  are integer constants,  $n$  is the number of places in PN model and  $m_i$  is the number of tokens in place  $P_i$ . If we can obtain  $x$  and  $k_i$  (for  $i=1, \dots, n$ ) so that all the authorized states verify the constraint (8) and all the border forbidden states violate it, then the resultant constraint can be considered as a GMEC for preventing the system from entering all the forbidden states.

Verifying this constraint (constraint (8)) by the authorized states is obtained when the term  $(k_1 m_1 + k_2 m_2 + \dots + k_n m_n)$  is smaller than or equal to  $x$  for all the authorized states and violating it by the border forbidden states is obtained when the term  $(k_1 m_1 + k_2 m_2 + \dots + k_n m_n)$  is greater than  $x$  for all the border forbidden states. So, for verifying this constraint by the authorized states, we substitute the markings of all the authorized states in the constraint (8) and obtain a set of inequalities. Then for violating the constraint (8) by the border forbidden states, we substitute the markings of all the border forbidden states in the constraint (8) and convert the smaller equal sign to greater sign to obtain another set of inequalities. Now, we consider the two sets of obtained inequalities and

solve them to obtain an answer (the answers are  $x$  and  $k_i$  for  $i=1, \dots, n$ ). This is an ILP problem where the objective function is:

Minimum  $(k_1+k_2+\dots+k_n+x)$  in which  $x>0$  and  $k_i \geq 0$  for  $i=1, \dots, n$ .

If this ILP problem has an answer, we should substitute the obtained  $x$  and  $k_i$  in the constraint (8). So, the obtained constraint will be a GMEC for all the forbidden states. This concept is introduced in Algorithm 1.

**Algorithm 1 [25].** One GMEC for preventing all the forbidden states.

**Input:** The set of authorized states  $M_A$  and the set of border forbidden states  $M_B$ .

**Output:** One GMEC for preventing all the forbidden states.

Suppose that  $M_A=\{[z_{11} z_{12} \dots z_{1n}], \dots, [z_{r1} z_{r2} \dots z_{rn}]\}$  is the set of authorized states and  $M_B=\{[B_{11} B_{12} \dots B_{1n}], \dots, [B_{t1} B_{t2} \dots B_{tn}]\}$  is the set of border forbidden states. Follow these steps to obtain a GMEC related to all the forbidden states:

**Step 1.** Consider a generic constraint as follows:

$$k_1m_1+k_2m_2+\dots+k_nm_n \leq x \tag{9}$$

where  $n$  is the number of places,  $m_i$  is the number of tokens in place  $P_i$ , and  $x$  and  $k_i$  for  $i=1,2,\dots,n$  are integer constants.

**Step 2.** Substitute the markings of all the authorized states in the constraint (9) and construct the inequalities as follows:

$$[z_{11} z_{12} \dots z_{1n}] \rightarrow k_1z_{11}+k_2z_{12}+\dots+k_nz_{1n} \leq x \tag{10-1}$$

$$\vdots$$

$$[z_{r1} z_{r2} \dots z_{rn}] \rightarrow k_1z_{r1} + k_2z_{r2} + \dots + k_nz_{rn} \leq x \tag{10-r}$$

where  $r$  is the number of authorized states.

**Step 3.** Substitute the markings of all the border forbidden states in the constraint (9) and convert the smaller equal sign to greater sign as follows:

$$[B_{11} B_{12} \dots B_{1n}] \rightarrow k_1B_{11} + k_2B_{12} + \dots + k_nB_{1n} > x \tag{11-1}$$

$$\vdots$$

$$[B_{t1} B_{t2} \dots B_{tn}] \rightarrow k_1B_{t1} + k_2B_{t2} + \dots + k_nB_{tn} > x \tag{11-t}$$

where  $t$  is the number of border forbidden states.

**Step 4.** Solve the set of relations (10-1) to (10- r)

and (11-1) to (11-t) which is an ILP problem and obtain the minimum values of  $x$  and  $k_i$  for  $i=1,2,\dots,n$ . (in this problem the objective function is: minimum  $(k_1+k_2+\dots+k_n+x)$  where  $x>0$  and  $k_i \geq 0$  for  $i=1,2,\dots,n$ )

**Step 5.** If step 4 has an answer, then substitute  $x$  and  $k_i$  for  $i=1,2,\dots,n$  in the constraint (9). The resultant constraint is a GMEC for all the forbidden states that enforcing it on the system leads to obtaining a maximally permissive behavior.

□ Now, an example is considered to show the impact of this method.

**Example 1 [25].** Consider the PN model in Fig. 1. This figure models a system composed of two machines  $M_1$  and  $M_2$  and two robots. The start command of machine  $M_1$  ( $M_2$ ) is accomplished by firing of the controllable transition  $t_1$  ( $t_2$ ) and the end command of this machine is accomplished by firing of the uncontrollable transition  $t_3$  ( $t_4$ ). When each machine completes its task, one of the robots must convey the constructed piece. Each firing of the uncontrollable transition  $t_5$  release one of the robots. In this model  $\langle P_1t_1P_2t_3P_1 \rangle$  and  $\langle P_3t_2P_4t_4P_3 \rangle$  are corresponding to the process model and  $\langle P_5t_3P_6t_5P_5 \rangle$  and  $\langle P_5t_4P_6t_5P_5 \rangle$  are corresponding to the specification model.

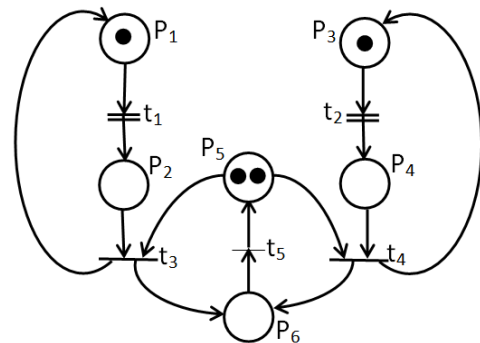


Fig. 1. The PN model of the system in example 1.

In the system with uncontrollable transitions there may be a problem when the model of process and the model of specification are synchronized in an uncontrollable transition. In this case the closed loop model may not complete its task correctly. The problem exists when the input places of the uncontrollable transition related to process are marked and the input places of this uncontrollable transition related to specification are unmarked.

Due to the uncontrollability of the transition, the closed loop model cannot respect the firing rules of PN model. For instance, in this example consider the uncontrollable transition  $t_4$  in the closed loop model.

The input places of this transition are  $P_4$  and  $P_5$  at which  $P_4$  is related to the process and  $P_5$  is related to the specification. Suppose that the system is in the state  $[1\ 0\ 0\ 1\ 0\ 2]$ .

In this state according to the closed loop model, the transition  $t_4$  cannot fire but since in the process model, this transition is uncontrollable, the controller cannot disable the transition to verify the specification. Therefore, this state is a forbidden state. So, the controller must disable the controllable transitions in a special condition (before entering this state) to prevent the system from entering this state. This concept is the same for the states  $[0\ 1\ 0\ 1\ 1\ 1]$ ,  $[0\ 1\ 1\ 0\ 0\ 2]$ , and  $[0\ 1\ 0\ 1\ 0\ 2]$ .

So, these states are forbidden states. In the closed loop model when the system is in these states, the states after firing the uncontrollable transitions are unknown and are shown by  $\otimes$ .

The reachability graph of this system is illustrated in Fig 2. In this figure, the forbidden states are depicted in grey color.

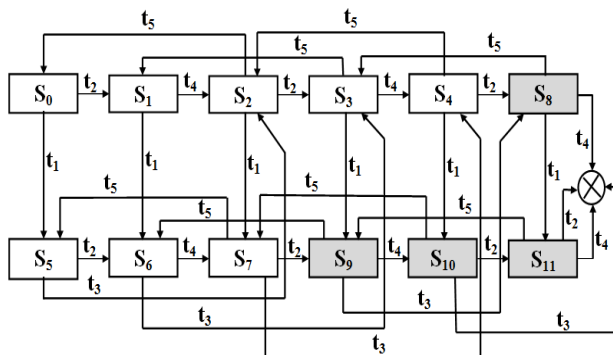


Fig. 2. The reachability graph of the system in example 1.

The detailed characterization of the markings corresponding to the various nodes of the graph of Fig. 2 can be found in Table 1 related to the authorized states and in Table 2 related to the forbidden states (in Table 2, the states  $S_8$ ,  $S_9$  and  $S_{10}$  are border forbidden states).

Tab. 1. The authorized states of the system in example 1.

AUTHORIZED STATE	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$
$S_0$	1	0	1	0	2	0
$S_1$	1	0	0	1	2	0
$S_2$	1	0	1	0	1	1
$S_3$	1	0	0	1	1	1
$S_4$	1	0	1	0	0	2
$S_5$	0	1	1	0	2	0
$S_6$	0	1	0	1	2	0
$S_7$	0	1	1	0	1	1

Tab. 2. The forbidden states of the system in example 1.

FORBIDDEN STATE	$P_1$	$P_2$	$P_3$	$P_4$	$P_5$	$P_6$
$S_8$	1	0	0	1	0	2
$S_9$	0	1	0	1	1	1
$S_{10}$	0	1	1	0	0	2
$S_{11}$	0	1	0	1	0	2

In this example, the forbidden states are the ones that preventing them leads to verifying the specification. So, to prevent the system from entering these states, Algorithm 1 can be used to generate a GMEC. Therefore, according to Algorithm 1, a constraint should be considered as follows (step 1 of Algorithm 1):

$$k_1m_1+k_2m_2+\dots+k_6m_6 \leq x \tag{12}$$

Substituting the markings of the authorized states in the constraint (12) leads to the set of inequalities as follows (step 2 of Algorithm 1):

$$A_1 = \{k_1+k_3+2k_5 \leq x, \quad k_1+k_3+k_5+k_6 \leq x, \quad k_1+k_3+2k_6 \leq x, \\ k_1+k_4+2k_5 \leq x, \quad k_1+k_4+k_5+k_6 \leq x, \quad k_1+k_4+2k_6 \leq x, \\ k_2+k_3+2k_5 \leq x, \quad k_2+k_3+k_5+k_6 \leq x, \quad k_2+k_3+2k_6 \leq x, \\ k_2+k_4+2k_5 \leq x, \quad k_2+k_4+2k_6 \leq x\}$$

And substituting the border forbidden states ( $S_8$ ,  $S_9$  and  $S_{10}$ ) in the constraint (12) and converting the smaller equal sign to greater sign leads to the set of inequalities as follows (step 3 of Algorithm 1):

$$A_2 = \{k_1+k_4+2k_6 > x, \quad k_2+k_3+2k_6 > x, \quad k_2+k_4+k_5+k_6 > x\}$$

Now, the ILP problem composed of the inequalities in  $A_1$  and  $A_2$  and  $x > 0$  and  $k_i \geq 0$  for  $i=1, \dots, 6$  where the objective function is minimum ( $k_1+k_2+\dots+k_6+x$ ), must be solved. The answer for this ILP problem is as follows:

$$k_1=0, \quad k_2=1, \quad k_3=1, \quad k_4=2, \quad k_5=0, \quad k_6=1, \quad x=3$$

So, the constraint (12) is obtained as the following form:

$$m_2+m_3+2m_4+m_6 \leq 3 \tag{13}$$

Enforcing this constraint on the system prevents it from entering the forbidden states. This constraint can be considered as a GMEC for all the forbidden states. The incidence matrix and the initial marking related to this control place are as follows:

$$W_c = [-1 \ -1 \ 0 \ 0 \ 1] \quad M_{c0} = m_{Pc} = [2] \tag{14}$$

The controlled model of this system after enforcing the GMEC is depicted in Fig. 3.

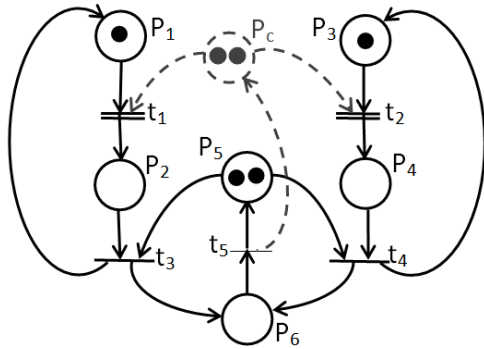


Fig. 3. The controlled model of the system in example 1.

In this figure, the control place and the related arcs are shown in gray color and dashed lines.

But Algorithm 1 has two disadvantages as follows:

- When the number of forbidden states is large, Algorithm 1 may not generate an answer.
- When the number of places is large, the number of variables is large and solving the ILP problem may be difficult.

To solve the first problem, we can classify the border forbidden states in some groups and consider each one of these groups in step 3 of algorithm 1, separately. Classification of the border forbidden states in some groups should be in the form that Algorithm 1 must generate an answer for each group and the number of groups must be the smallest one. In this case, Algorithm 1 can generate a GMEC for each group. But classification of the border forbidden states is a problem.

In the next section a powerful method will be proposed which using the optimization algorithms classifies the border forbidden states in some groups and generates a GMEC for each group. Also, using this method, it is not necessary to solve the ILP problem and the constants ( $k_i$  and  $x$ ) will be obtained during the classification of the border forbidden states. So, impact of the second problem in above decreases.

#### 4. Obtaining Small Number of GMECs Using an Iterative Based Algorithm

In this section the objective is to develop Algorithm 1 for proposing an efficient method to obtain a small number of GMECs related to forbidden states. The main idea of the proposed method is classification of the inequalities related to border forbidden states (step 3 of Algorithm 1) in some different groups. Also during this classification, by considering the inequalities related to the authorized states (step 2 of Algorithm 1), the constants ( $x$  and  $k_i$ ) can be obtained for each group without solving any ILP problem. This will be performed using an iterative based algorithm which tries to find the biggest group of the inequalities related to the forbidden states that can be satisfied by the constants ( $x$  and  $k_i$ ).

The proposed method is described in Algorithm 2.

**Algorithm 2.** Obtaining a small number of GMECs related to border forbidden states in bounded PNs.

**Input:** The set of authorized states  $M_A$ , and the set of border forbidden states  $M_B$ .

**Output:** The small number of GMECs for preventing the system from entering the forbidden states.

Suppose that  $M_A = \{[z_{11} z_{12} \dots z_{1n}], \dots, [z_{r1} z_{r2} \dots z_{rn}]\}$  is the set of authorized states and  $M_B = \{[B_{11} B_{12} \dots B_{1n}], \dots, [B_{l1} B_{l2} \dots B_{ln}]\}$  is the set of border forbidden states.

**Step 1.** Consider a generic constraint as follows:

$$k_1 m_1 + k_2 m_2 + \dots + k_n m_n \leq x \tag{15}$$

where  $n$  is the number of places and  $x$  and  $k_i$  (for  $i=1, \dots, n$ ) are positive integers.

**Step 2.** Substitute the markings of the authorized states in the constraint (15) and consider the obtained constraints as follows:

$$\sum_{i=1}^n z_{j,i} \cdot k_i \leq x \quad j = 1, 2, \dots, r \tag{16}$$

These constraints are called SE constraints where  $r$  represents the number of the constraints (or the number of authorized states).

**Step 3.** Substitute the markings of the border forbidden states in the constraint (15) and convert the smaller equal sign to greater sign and consider the obtained constraints as the following form:

$$\sum_{i=1}^n B_{l,i} \cdot k_i > x \quad l = 1, 2, \dots, t \tag{17}$$

These constraints are called G constraints where  $t$  represents the number of the constraints.

**Step 4.** Try to find a set of integer constants  $x$  and  $k_i$  (for  $i=1, \dots, n$ ) which: a) satisfy all of the SE constraints and b) satisfy the maximum number of the G constraints. After finding the set of constants  $x$  and  $k_i$ , consider the satisfied G constraints as the biggest group among the given set of G constraints and substitute  $x$  and  $k_i$  in the constraint (15). The obtained constraint is a GMEC for this group (the detailed process of this step for finding  $x$  and  $k_i$  is discussed in section 4.1).

**Step 5.** Save the biggest group found in step 4 with its GMEC, and then remove all the G constraints related to the biggest group from the set of G constraints.

**Step 6.** After removing the biggest group, if there is no ungrouped G constraint, the algorithm becomes terminated, otherwise continue the algorithm from step 4 considering the residual ungrouped G constraints as the new given G constraints. After termination of the process, there is an archive of the grouped G constraints with their respective GMECs. The obtained GMECs are the ones that enforcing them on the system leads to obtaining maximally permissive behavior.

It should be noted that the number of G constraints related to group  $y$  ( $y$  is the number of iteration) is always greater than or equal to the number of these constraints related to group  $y+1$ .  $\square$

**4.1. Finding  $x$  and  $k_i$  ( $i=1, \dots, n$ ) for Satisfying all the SE Constraints and the Biggest Group of G Constraints**

To find the set of integer constants  $x$  and  $k_i$  for satisfying all the SE constraints and the maximum number of the G constraints, the following optimization problem is considered:

$$\text{Minimize } f(K) = x(K) + \lambda \times UI(K) \tag{18}$$

$$x(K) = \text{round} \left( \begin{matrix} \max \left\{ \sum_{i=1}^n z_{1,i} \cdot k_i, \sum_{i=1}^n z_{2,i} \cdot k_i, \dots, \right. \\ \left. \sum_{i=1}^n z_{j,i} \cdot k_i, \dots, \sum_{i=1}^n z_{r,i} \cdot k_i \right\} \end{matrix} \right) \tag{19}$$

where  $K = [k_1, k_2, \dots, k_i, \dots, k_n]$  is the set of decision variables of the optimization problem.  $f(K)$  is value of the objective function related to the given set  $K$ .  $UI(K)$  represents the number of unsatisfied G constraints considering the given set  $K$ .  $\lambda$  ( $\lambda \geq 1$ ) is a proportional factor and is used to increase the importance of minimizing  $UI(K)$  with respect to minimizing  $x(K)$  in the objective function. This means that the main aim of this optimization problem is to minimize the number of

unsatisfied G constraints in order to satisfy the biggest group of G constraints (this leads to obtaining a small number of control places).  $x(K)$  is considered in the objective function since the small value for  $x$  is suitable.

For satisfying all the SE constraints, we consider  $x$  as a dependent variable which can be found from (19) for the given set  $K$ . The operand “ $\text{round}(a)$ ” rounds the value  $a$  toward the nearest positive integer value, and the operand “ $\max(a, b, \dots, h, \dots, l)$ ” select the maximum value from the values  $a, b, \dots, h, \dots, l$ . By choosing the exact value for  $\lambda$  and solving the above problem using an optimization solver, both the biggest group of G constraints and also the minimum value for  $x$  can be satisfied. For solving the proposed optimization problem, the Modified Adaptive Particle Swarm Optimization (MAPSO) algorithm presented in [26] is used (it is supposed that the reader is familiar with MAPSO algorithm). The MAPSO is based on the real value of particle swarm optimization (PSO) algorithm, and so, we round the proposed solutions of the algorithm during the optimization process to the nearest positive integer values. Solving the mentioned problem using this algorithm leads to an optimally better solution comparing with the conventional methods [2], [9].

To see the effectiveness of the new idea presented in this section, we apply it to some examples and compare the results with the conventional works.

**Example 2:** A PN model similar to Example 1 is considered in Fig 4. In this figure, the transitions  $t_1$  and  $t_2$  are controllable and the transitions  $t_3, t_4, t_5$  and  $t_6$  are uncontrollable. The paths  $\langle P_1 t_1 P_2 t_3 P_1 \rangle$  and  $\langle P_3 t_2 P_4 t_4 P_3 \rangle$  are corresponding to process and the paths  $\langle P_5 t_3 P_6 t_5 P_5 \rangle$ ,  $\langle P_5 t_4 P_6 t_5 P_5 \rangle$ ,  $\langle P_7 t_3 P_8 t_6 P_7 \rangle$  and  $\langle P_7 t_4 P_8 t_6 P_7 \rangle$  are corresponding to specifications. In this example the set of markings of the authorized states and the border forbidden states can be found in Table 3 and 4 respectively.

**Tab. 3. The marking of the authorized states related to the system in example 2**

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	P <sub>8</sub>		P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	P <sub>8</sub>
S <sub>0</sub>	1	0	1	0	2	0	2	0	S <sub>9</sub>	1	0	0	1	2	0	1	1
S <sub>1</sub>	1	0	1	0	2	0	1	1	S <sub>10</sub>	1	0	0	1	2	0	2	0
S <sub>2</sub>	1	0	1	0	2	0	0	2	S <sub>11</sub>	1	0	0	1	1	1	1	1
S <sub>3</sub>	1	0	1	0	1	1	0	2	S <sub>12</sub>	1	0	0	1	1	1	2	0
S <sub>4</sub>	1	0	1	0	1	1	1	1	S <sub>13</sub>	0	1	1	0	2	0	1	1
S <sub>5</sub>	1	0	1	0	1	1	2	0	S <sub>14</sub>	0	1	1	0	2	0	2	0
S <sub>6</sub>	1	0	1	0	0	2	0	2	S <sub>15</sub>	0	1	1	0	1	1	1	1
S <sub>7</sub>	1	0	1	0	0	2	1	1	S <sub>16</sub>	0	1	1	0	1	1	2	0
S <sub>8</sub>	1	0	1	0	0	2	2	0	S <sub>17</sub>	0	1	0	1	2	0	1	0

**Tab. 4. The marking of the border forbidden states related to the system in example 2**

	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	P <sub>8</sub>		P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>	P <sub>6</sub>	P <sub>7</sub>	P <sub>8</sub>
S <sub>18</sub>	0	1	1	0	0	2	0	2	S <sub>25</sub>	1	0	0	1	1	1	0	2
S <sub>19</sub>	0	1	0	1	1	1	1	1	S <sub>26</sub>	0	1	1	0	0	2	1	1
S <sub>20</sub>	1	0	0	1	0	2	1	1	S <sub>27</sub>	0	1	1	0	2	0	0	2
S <sub>21</sub>	0	1	1	0	1	1	0	2	S <sub>28</sub>	0	1	0	1	1	1	2	0
S <sub>22</sub>	1	0	0	1	2	0	0	2	S <sub>29</sub>	0	1	1	0	0	2	2	0
S <sub>23</sub>	0	1	0	1	2	0	1	1	S <sub>30</sub>	1	0	0	1	0	2	2	0
S <sub>24</sub>	1	0	0	1	0	2	0	2	-	-	-	-	-	-	-	-	-

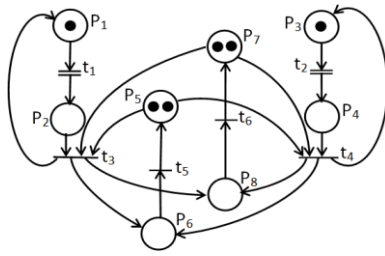


Fig. 4. The PN model in example 2.

Algorithm 1 cannot generate any answer for this example. So, for preventing the system from entering the forbidden states, according to Algorithm 2, we consider a generic constraint as follows:

$$k_1m_1+k_2m_2+\dots+k_8m_8 \leq x \tag{20}$$

Substituting the markings of the authorized states in the constraint (20) leads to the set of SE constraints as follows:

$$S_1 = \{k_1+k_3+2k_5+2k_8 \leq x, k_1+k_3+2k_5+k_7+k_8 \leq x, k_1+k_3+2k_5+2k_7 \leq x, k_1+k_3+k_5+k_6+2k_8 \leq x, k_1+k_3+k_5+k_6+k_7+k_8 \leq x, k_1+k_3+k_5+k_6+2k_7 \leq x, k_1+k_3+2k_6+2k_8 \leq x, k_1+k_3+2k_6+k_7+k_8 \leq x, k_1+k_3+2k_6+2k_7 \leq x, k_1+k_4+2k_5+k_7+k_8 \leq x, k_1+k_4+2k_5+2k_7 \leq x, k_1+k_4+k_5+k_6+k_7+k_8 \leq x, k_1+k_4+k_5+k_6+2k_7 \leq x, k_2+k_3+2k_5+k_7+k_8 \leq x, k_2+k_3+2k_5+2k_7 \leq x, k_2+k_3+k_5+k_6+k_7+k_8 \leq x, k_2+k_3+k_5+k_6+2k_7 \leq x, k_2+k_4+2k_5+2k_7 \leq x\}.$$

Substituting the markings of the border forbidden states in the constraint (20) and converting the smaller equal sign to greater sign leads to the set of G constraints as the following form:

$$S_2 = \{k_2+k_3+2k_6+2k_8 > x, k_2+k_4+k_5+k_6+k_7+k_8 > x, k_1+k_4+2k_6+k_7+k_8 > x, k_2+k_3+k_5+k_6+2k_8 > x, k_1+k_4+2k_5+2k_8 > x, k_2+k_4+2k_5+k_7+k_8 > x, k_1+k_4+2k_6+2k_8 > x, k_1+k_4+k_5+k_6+2k_8 > x, k_2+k_3+2k_6+k_7+k_8 > x, k_2+k_3+2k_5+2k_8 > x, k_2+k_4+k_5+k_6+2k_7 > x, k_2+k_3+2k_6+2k_7 > x, k_1+k_4+2k_6+2k_7 > x\}.$$

The objective function is:

$$f(K) = x(K) + \lambda \times UI(K) \tag{21}$$

And

$$x(K) = \text{round} \{ \max[(k_1+k_3+2k_5+2k_8), (k_1+k_3+2k_5+k_7+k_8), (k_1+k_3+2k_5+2k_7), (k_1+k_3+k_5+k_6+2k_8), (k_1+k_3+k_5+k_6+k_7+k_8), (k_1+k_3+k_5+k_6+2k_7), (k_1+k_3+2k_6+2k_8), (k_1+k_3+2k_6+k_7+k_8), (k_1+k_3+2k_6+2k_7), (k_1+k_4+2k_5+k_7+k_8), (k_1+k_4+2k_5+2k_7), (k_1+k_4+k_5+k_6+k_7+k_8), (k_1+k_4+k_5+k_6+2k_7), (k_2+k_3+2k_5+k_7+k_8), (k_2+k_3+2k_5+2k_7), (k_2+k_3+k_5+k_6+k_7+k_8), (k_2+k_3+k_5+k_6+2k_7), (k_2+k_4+2k_5+2k_7)] \},$$

where  $K = [k_1, k_2, k_3, k_4, k_5, k_6, k_7, k_8]$ . By solving the above problem using MAPSO [26] and considering  $\lambda=2$ , the G constraints can be classified into two groups. So the first answer related to the first group is as follows:

$$k_1 = 0, k_2 = 2, k_3 = 0, k_4 = 2, k_5 = 1, k_6 = 2, k_7 = 1, k_8 = 1$$

and  $x = 7$ ,

By substituting this answer in the constraint (20), the first GMEC is:

$$2m_2+2m_4+m_5+2m_6+m_7+m_8 \leq 7 \tag{22}$$

The second answer related to the second group is:

$$k_1 = 0, k_2 = 2, k_3 = 0, k_4 = 2, k_5 = 1, k_6 = 1, k_7 = 0, k_8 = 2 \text{ and } x = 6,$$

By substituting this answer in the constraint (20), the second GMEC is obtained as the following form:

$$2m_2+2m_4+m_5+m_6+2m_8 \leq 6 \tag{23}$$

The incidence matrix related to these two GMECs is as follows:

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

The initial marking of the two control places is:

$$M_{c,0} = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

Enforcing the GMECs (22) and (23) on the system leads to preventing the system from entering all the forbidden states. This example shows the capability of the new method for reducing the number of GMECs.

Now in the following, we introduce other practical examples which are also used in the previous works, to highlight the differences between our work and the previous works.

**Example 3.** Consider the resource allocation System in Fig. 5 which was presented in [5] and was used in [2]. In this system there are two processing part types. Initially it is assumed that there are no parts in the system. Places  $P_1, P_2$  and  $P_3$  illustrate using the resources  $R_1, R_2$  and  $R_3$  respectively by the first part type. Places  $P_{21}, P_{22}$  and  $P_{23}$  represent using the resources  $R_3, R_2$  and  $R_1$  respectively by the second part type. The number of concurrent activities that can take place for the two part types is 4 ( $m_{10}=m_{20}=4$ ). Places  $P_{31}, P_{32}$  and  $P_{33}$  represent the shared resources  $R_1, R_2$  and  $R_3$  respectively. In this model, all the transitions are controllable.

In this example, the objective is to prevent the system from entering the deadlock states. The number of reachable states is 47 at which 42 ones are authorized and 5 ones are forbidden. The marking of the authorized states and the forbidden states (which are also border forbidden states) can be found in Table 5 and 6 respectively.



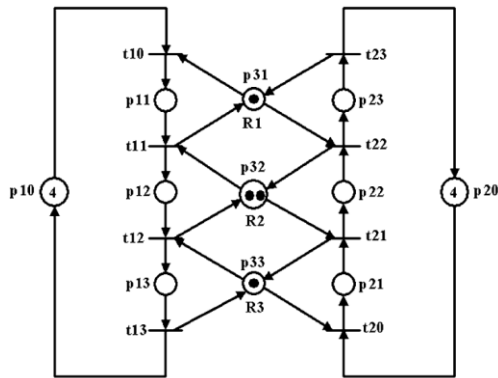


Fig. 5. The resource allocation system in example 3.

Using Algorithm 2, the G constraints related to the forbidden states can be classified into two groups. The first answer related to the first group is as follows:

$$k_1 = 0, k_2 = 2, k_3 = 1, k_4 = 0, k_5 = 0, k_6 = 1, k_7 = 2, k_8 = 0, k_9 = 0, k_{10} = 0, k_{11} = 0 \text{ and } x = 5.$$

So, the first GMEC is:

$$2m_2 + m_3 + m_6 + 2m_7 \leq 5 \tag{24}$$

The second answer related to the second group is obtained as follows:

$$k_1 = 0, k_2 = 1, k_3 = 2, k_4 = 0, k_5 = 0, k_6 = 1, k_7 = 1, k_8 = 1, k_9 = 1, k_{10} = 1, k_{11} = 0 \text{ and } x = 5,$$

Then the second GMEC is as the following form:

$$m_2 + 2m_3 + m_6 + m_7 + m_8 + m_9 + m_{10} \leq 5 \tag{25}$$

The incidence matrix related to these two GMECs is:

$$W_c = \begin{bmatrix} -2 & 1 & 1 & 0 & -1 & -1 & 2 & 0 \\ 0 & -1 & 1 & 0 & -1 & 1 & 0 & 0 \end{bmatrix} \tag{26}$$

The initial marking related to the control places is:

$$M_{c0} = \begin{bmatrix} 5 \\ 2 \end{bmatrix} \tag{27}$$

Tab. 5. The marking of the authorized states related to the system in example 3

	P <sub>10</sub>	P <sub>11</sub>	P <sub>12</sub>	P <sub>13</sub>	P <sub>20</sub>	P <sub>21</sub>	P <sub>22</sub>	P <sub>23</sub>	P <sub>31</sub>	P <sub>32</sub>	P <sub>33</sub>
S <sub>0</sub>	4	0	0	0	4	0	0	0	1	2	1
S <sub>1</sub>	3	1	0	0	4	0	0	0	0	2	1
S <sub>2</sub>	4	0	0	0	3	1	0	0	1	2	0
S <sub>3</sub>	3	0	1	0	4	0	0	0	1	1	1
S <sub>4</sub>	3	1	0	0	3	1	0	0	0	2	0
S <sub>5</sub>	4	0	0	0	3	0	1	0	1	1	1
S <sub>6</sub>	2	1	1	0	4	0	0	0	0	1	1
S <sub>7</sub>	3	0	0	1	4	0	0	0	1	2	0
S <sub>8</sub>	3	0	1	0	3	1	0	0	1	1	0
S <sub>9</sub>	3	1	0	0	3	0	1	0	0	1	1
S <sub>10</sub>	4	0	0	0	2	1	1	0	1	1	0
S <sub>11</sub>	4	0	0	0	3	0	0	1	0	2	1
S <sub>12</sub>	2	0	2	0	4	0	0	0	1	0	1
S <sub>13</sub>	2	1	0	1	4	0	0	0	0	2	0
S <sub>14</sub>	2	1	1	0	3	1	0	0	0	1	0
S <sub>15</sub>	3	0	1	0	3	0	1	0	1	0	1
S <sub>16</sub>	3	1	0	0	2	1	1	0	0	1	0
S <sub>17</sub>	4	0	0	0	2	0	2	0	1	0	1
S <sub>18</sub>	4	0	0	0	2	1	0	1	0	2	0
S <sub>19</sub>	1	1	2	0	4	0	0	0	0	0	1
S <sub>20</sub>	2	0	1	1	4	0	0	0	1	1	0
S <sub>21</sub>	2	1	1	0	3	0	1	0	0	0	1
S <sub>22</sub>	3	0	0	1	3	0	1	0	1	1	0
S <sub>23</sub>	3	0	1	0	2	1	1	0	1	0	0
S <sub>24</sub>	3	0	1	0	3	0	0	1	0	1	1
S <sub>25</sub>	4	0	0	0	1	1	2	0	1	0	0
S <sub>26</sub>	4	0	0	0	2	0	1	1	0	1	1
S <sub>27</sub>	1	1	1	1	4	0	0	0	0	1	0
S <sub>28</sub>	2	1	0	1	3	0	1	0	0	1	0
S <sub>29</sub>	3	0	0	1	3	0	0	1	0	2	0
S <sub>30</sub>	3	0	1	0	2	1	0	1	0	1	0
S <sub>31</sub>	4	0	0	0	1	1	1	1	0	1	0
S <sub>32</sub>	1	0	2	1	4	0	0	0	1	0	0
S <sub>33</sub>	2	0	1	1	3	0	1	0	1	0	0
S <sub>34</sub>	3	0	1	0	2	0	1	1	0	0	1
S <sub>35</sub>	4	0	0	0	1	0	2	1	0	0	1
S <sub>36</sub>	0	1	2	1	4	0	0	0	0	0	0
S <sub>37</sub>	1	1	1	1	3	0	1	0	0	0	0
S <sub>38</sub>	2	0	1	1	3	0	0	1	0	1	0
S <sub>39</sub>	3	0	0	1	2	0	1	1	0	1	0
S <sub>40</sub>	3	0	1	0	1	1	1	1	0	0	0
S <sub>41</sub>	4	0	0	0	0	1	2	1	0	0	0

Tab. 6. The marking of the forbidden states related to the system in example 3

	P <sub>10</sub>	P <sub>11</sub>	P <sub>12</sub>	P <sub>13</sub>	P <sub>20</sub>	P <sub>21</sub>	P <sub>22</sub>	P <sub>23</sub>	P <sub>31</sub>	P <sub>32</sub>	P <sub>33</sub>
S <sub>42</sub>	1	1	2	0	3	1	0	0	0	0	0
S <sub>43</sub>	2	1	1	0	2	1	1	0	0	0	0
S <sub>44</sub>	3	1	0	0	1	1	2	0	0	0	0
S <sub>45</sub>	2	0	2	0	3	1	0	0	1	0	0
S <sub>46</sub>	3	1	0	0	2	0	2	0	0	0	1

By adding the two control places related to the GMECs (24) and (25), the system is prevented from entering the deadlock states and the 42 authorized states are obtained. Also, as it is obvious from (26) the number of arcs related to these two control places is 12 (this number is obtained by considering the weights associating with arcs).

**Example 4.** A Similar example similar to Example 3 is considered in Fig. 6 which was presented in [28] and was used in [2].

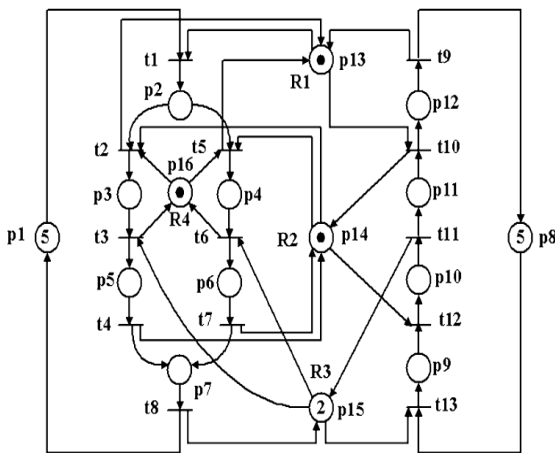


Fig. 6. The resource allocation system in example 4.

In this example the objective is avoiding the deadlock states. The number of reachable states is 77 at which 64 ones are authorized and 13 ones are forbidden.

By using Algorithm 2 for preventing the system from entering the forbidden states, the G constraints are classified into two groups. So, the answer related to the first group is as follows:

$$k_1 = 1, k_2 = 2, k_3 = 2, k_4 = 2, k_5 = 2, k_6 = 1, k_7 = 1, k_8 = 0, k_9 = 2, k_{10} = 2, k_{11} = 0, k_{12} = 0, k_{13} = 0, k_{14} = 0, k_{15} = 0, k_{16} = 0, \text{ and } x = 9.$$

Therefore, the first GMEC is:

$$m_1 + 2m_2 + 2m_3 + 2m_4 + 2m_5 + m_6 + m_7 + 2m_9 + 2m_{10} \leq 9 \quad (28)$$

The answer related to the second group is as the

following form:

$$k_1 = 0, k_2 = 2, k_3 = 1, k_4 = 1, k_5 = 1, k_6 = 1, k_7 = 1, k_8 = 0, k_9 = 1, k_{10} = 2, k_{11} = 1, k_{12} = 1, k_{13} = 1, k_{14} = 0, k_{15} = 1, k_{16} = 1, \text{ and } x = 5.$$

So, the second GMEC is:

$$2m_2 + m_3 + m_4 + m_5 + m_6 + m_7 + m_9 + 2m_{10} + m_{11} + m_{12} + m_{13} + m_{15} + m_{16} \leq 5 \quad (29)$$

The incidence matrix related to these two GMECs is as follows:

$$W_c = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & -2 \\ -1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \end{bmatrix} \quad (30)$$

The initial marking of these two control places is:

$$M_{c0} = \begin{bmatrix} 4 \\ 1 \end{bmatrix} \quad (31)$$

Enforcing the two GMECs (28) and (29) on the system using control places, prevents the system from entering all the forbidden states. Moreover, all the 64 authorized states will be reachable. In this controlled system the number of arcs related to the control places is 12.

### 5. Discussion

In this section we want to compare our method with the previous methods [2] and [9]. In this case, we compare the results after applying our method to examples 3 and 4 (which are used in a lot of papers) with the results after using the previous methods. So, Table 7 is constructed.

In this Table,  $n_{bf}$  is the number of border forbidden states,  $n_{cp}$  is the number of control places,  $n_{arc}$  is the number of arcs related to the control places (this number is computed by considering the weights associating with arcs),  $n_r$  is the number of reachability graph which must be computed and  $n_{ILP}$  is the number of ILP problems which must be solved.

Tab. 7. Comparing our method with the previous methods

	The proposed method					The method in [2]					The method in [9]				
	$n_{bf}$	$N_{cp}$	$n_{arc}$	$n_r$	$n_{ILP}$	$n_{bf}$	$N_{cp}$	$n_{arc}$	$n_r$	$n_{ILP}$	$n_{bf}$	$N_{cp}$	$n_{arc}$	$n_r$	$n_{ILP}$
Example 3	5	2	12	1	0	5	3	12	5	0	5	3	12	1	4
Example 4	13	2	12	1	0	13	3	16	6	0	13	3	16	1	5
Sum	18	4	24	2	0	18	6	28	11	0	18	6	28	2	9

In example 3, using the method in [28], 4 control places are generated for preventing the system from entering the deadlock states at which the number of their related arcs ( $n_{arc}$ ) is 16. Using the methods in [2] and [9], one of these control places is recognized redundant and can be eliminated. So, by three residual control places a maximally permissive behavior is obtained at which the number of their related arcs ( $n_{arc}$ ) is 12. However using the new method in this paper, two control places are directly obtained where the number of their related arcs is 12 and a maximally permissive behavior is obtained. Moreover, using our method, we compute the reachability graph one time and solving the ILP problem is not necessary.

In example 4, by using the method in [28], 5 control places are computed for avoiding the deadlock states where the number of their related arcs is 31. By using the methods in [2] and [9], two control places are recognized redundant and can be eliminated. So, three remained control places generate maximally permissive behavior at which the number of their related arcs is 16.

But using our method, by computing the one reachability graph and without solving any ILP problem, two control places are directly obtained that generate the maximally permissive behavior. Also, the number of related arcs is 12.

As it is obvious from Table 7, after applying our method to all the examples, the number of control places and the number of arcs are smaller than or equal to these numbers after using the previous methods. Moreover, using our method the small number of control places are directly computed in one step but using the previous methods, the large number of control places must be calculated using some methods [17], [28] and other methods [2], [9] must eliminate the redundant control places among these places. So, using the previous methods, a simple controller is obtained in two steps. These concepts show that our method is simpler than the previous methods and generates better answers compared with them. However, the drawback of the new method is generating the reachability graph which is an exponential problem (Also the previous methods [2], [9] have this problem too).

## 6. Conclusion

In this paper a general method for solving the problem of the large number of control places has been proposed. This method tries to combine some constraints to obtain a constraint verifying all of them. So, using the optimization algorithms, the constraints which can generate an answer with each other constitute a group. Then, for each group a control place can be generated. The objective is to constitute the small number of groups for obtaining the small number of control places. Connecting the obtained control places to the system leads to a maximally permissive controller.

## References

- [1] Abdallah, I.B., ElMaraghy, H.A., "Deadlock Prevention and Avoidance in FMS: a Petri Net Based Approach", Int. J. Adv. Manuf. Tech, 14, 1998, pp. 704-715.
- [2] Uzam, M., Li, Z., Zhou, M., *Identification and Elimination of Redundant Control Places in Petri net Based Liveness Enforcing Supervisors of FMS*", Int. J. Adv. Manuf. Tech, 35, 2007, pp. 150-168.
- [3] Krogh, B.H., Holloway, L.E., "Synthesis of Feedback Control Logic for Discrete Manufacturing Systems", Automatica, 27, 1991, pp.641-651.
- [4] Visvanadham, N., Nahari, Y., Johnson, T.L., "Deadlock Prevention and Deadlock Avoidance in Flexible Manufacturing Systems using Petri Net Models", IEEE. Trans. Robot. Automat., 6(6), 1990, pp.713-723.
- [5] Park, J., Reveliotis, S. A., "Algebraic Synthesis of Efficient Deadlock Avoidance Policies for Sequential Resource Allocation Systems", IEEE. Trans. Robot. Automat, 16, 2000, pp. 190-195.
- [6] Piroddi, L., Cossalter, M., Ferrarini, L., "A Resource Decoupling Approach for Deadlock Prevention in FMS", Int. J. Adv. Manuf. Tech, 40, 2009, pp. 157-170.
- [7] Piroddi, L., Cordone, R., Fumagalli, I., "Selective Siphon Control for Deadlock Prevention in Petri Nets", IEEE Transactions on Systems, Man, and Cybernetics, Part A, 38, 2008, pp. 1337-1348.
- [8] Piroddi, L., Cordone, R., Fumagalli, I., "Combined Siphon and Marking Generation for Deadlock Prevention in Petri Nets", IEEE Transactions on Systems, Man and Cybernetics, Part A, 39, 2009, pp. 650-661.
- [9] Li, Z., Hu, H., "On Systematic Methods to Remove Redundant Monitors from Liveness-Enforcing Net Supervisors", Computer & Industrial Engineering, 56, 2009, pp. 53-62.
- [10] Ramadge, P.J., Wonham, W.M., "Modular Feedback Logic for Discrete Event Systems", SIAM Journal of Control and Optimization, 25(5), 1987, pp. 1202-1218.
- [11] Ramadge, P.J., Wonham, W., "The Control of Discrete Event Systems", Dynamics of discrete event systems [Special issue]. Proceedings of the IEEE, 77(1), 1989, pp. 81-98.
- [12] Cassandras, G.G., Lafontaine, S., Introduction to discrete event systems, Springer, 2008.
- [13] Giua, A., Petri Nets as Discrete Event Models for Supervisory Control. Ph.D. Thesis, 1992.
- [14] Kumar, R., Holloway, L.E., "Supervisory Control of Deterministic Petri Nets with Regular Specification Languages", IEEE. Trans. Automatic Control, 41(2): 1996, pp. 245-249.

- [15] Holloway, L.E., Guan, X., Zhang, L., "A Generalisation of State Avoidance Policies for Controlled Petri Nets", IEEE. Trans. Autom. Control, AC-41, 6, 1996, pp. 804-816.
- [16] Dideban, A., Alla, H., "Feedback Control Logic Synthesis for Non Safe Petri Nets", The 13th IFAC symposium on information control problems in manufacturing, Moscow, Russia. 2009.
- [17] Ghaffari, A., Rezg, N., Xie, X.L., "Design of Live and Maximally Permissive Petri Net Controller Using the Theory of Regions", IEEE Transactions on Robotics and Automation, 19(1), 2003.
- [18] Reveliotis, S.A., Choi, J.Y., "Designing Reversibility-Enforcing Supervisors of Polynomial Complexity for Bounded Petri Nets Through the Theory of Regions", LNCS, 4024, 2006, pp. 322-341.
- [19] Yamalidou, K., Moody, J., Lemmon, M., Antsaklis, P., "Feedback Control of Petri Nets Based on Place Invariants", Automatica, 32(1), 1996, pp. 15-28.
- [20] Giua, A., DiCesare, F.M., Silva, M., "Generalized Mutual Exclusion Constraints on Nets with Uncontrollable Transitions", In Proc. IEEE int. conf. on systems, man, and cybernetics, 1992, pp. 974-799.
- [21] Zareiee, M., Dideban, A., Nazemzadeh, P., "From Forbidden States to Linear Constraints", World Academy of Science, Engineering and Technology, 2011, pp. 358-364.
- [22] Dideban, A., Alla, H., "From Forbidden State to Linear Constraints for the Optimal Supervisory Control", Control Engineering and applied Informatics (CEAI), 7(3), 2005, pp. 48-55.
- [23] Dideban, A., Alla, H., "Reduction of Constraints for Controller Synthesis based on Safe Petri Nets", Automatica, 44, 2008, pp. 1697-1706.
- [24] Dideban, A., Zareiee, M., Alla, H., "Controller Synthesis with Very Simplified Linear Constraints in PN Model", The 2nd IFAC workshop on Dependable Control of Discrete Systems, Bari, Italy, June 10-12, 2009.
- [25] Zareiee, M., Dideban, A., "Reducing the Number of Constraints in Non Safe Petri Net". World Academy of Science, Engineering and Technology, 2011, pp. 354-357.
- [26] Amjady, N., Soleymanpour, H.R., "Daily Hydrothermal Generation Scheduling by a New Modified Adaptive Particle Swarm Optimization Technique", Electric power systems research, 2010, pp. 723-732.
- [27] David, R., Alla, H., "Discrete, continuous, and Hybrid Petri Nets", Springer, 2005.
- [28] Huang, Y.S., Jeng, M.D., Xie, X., Chung, D.H., "Siphon-Based Deadlock Prevention Policy for Flexible Manufacturing Systems", IEEE Trans Syst Man Cybern Part-A, 2006.