IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 6 January 2024, accepted 23 January 2024, date of publication 2 February 2024, date of current version 9 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3361811

RESEARCH ARTICLE

Optimality Test for Control Places of Petri Net Based Liveness Enforcing Supervisors of FMSs

MURAT UZAM [1], **ZHIWU LI** [2], **(Fellow, IEEE), MOHAMMED A. EL-MELIGY** [3], **MOHAMED A. SHARAF** [3], **AND QIWEI TANG** [4]

[1]Mühendislik-Mimarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü, Yozgat Bozok Üniversitesi, 66100 Yozgat, Turkey
[2]Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau, China
[3]Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[4]Hitachi Building Technology (Guangzhou) Company Ltd., Guangzhou 510670, China

Corresponding authors: Murat Uzam (murat.uzam@bozok.edu.tr) and Zhiwu Li (zwli@must.edu.mo)

This work was supported in part by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Program under Grant RSPD2024R704; in part by the Guangzhou Innovation and Entrepreneurship Leading Team Project Funding under Grant 202009020008; and in part by the Science and Technology Development Fund, Macao Special Administrative Region (MSAR), under Grant 0029/2023/RIA1.

ABSTRACT In the past three decades, a lot of Petri net-based methods have been proposed for deadlock prevention/liveness enforcing in flexible manufacturing systems (FMSs). Firstly, a plant Petri net model of an FMS is obtained and then the liveness enforcing supervisor (LES) or the controller is computed as a Petri net. An LES contains of a set of control places (CPs). The plant Petri net model and the LES are merged to obtain the controlled model. Once the Petri net model of an FMS is live, deadlocks never occur. When all legal markings of a Petri net model are reachable by the live system, the controlled model is called maximally permissive or optimal. If the controlled model is optimal, then all CPs are also optimal. However, when the controlled model is suboptimal, some CPs are optimal while the others are not. In order to improve behavioral permissiveness and/or to reduce the structural complexity of the CPs, it is crucial to identify the set of suboptimal CPs. This important issue has not been tackled before. To-date, when dealing with suboptimal controlled models no attention has been paid to identify both sets of optimal and suboptimal CPs. An optimality test for an LES of an FMS is proposed in this paper to address this problem. The optimality test takes an LPN model, controlled by a set of CPs, as input and in the case of suboptimal controlled models it produces both sets of optimal and suboptimal CPs. The optimality test proposed is applicable to any LPN that contains a Petri net model (PNM), controlled by means of a set of CPs. The applicability of this method is shown by considering several examples from the literature.

INDEX TERMS Flexible manufacturing system, deadlock, deadlock prevention, petri net (PN), liveness enforcing supervisor, optimality test.

I. INTRODUCTION

A flexible manufacturing system (FMS) contains of a set of shared resources such as robots, machines, storage devices and conveyors. These resources are used concurrently to process raw or intermediate parts through pre-defined manufacturing routes. Deadlock in an FMS is an unacceptable situation in which shared resources have been allocated parts so that further part movement is impossible. to To gain high productivity in an FMS, the effective handling of deadlocks is treated as a primary premise for a system's normal operation [1]. It is crucial for an efficient FMS control policy to ensure that deadlocks never occur. The past three decades have witnessed very fruitful investigations on deadlock resolution in FMSs [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43],

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

[44], [45], [46], [47], [49], [51], [52], [53], [54], [55], [56]. The following topics are three major methodologies to deal with the deadlock problems in FMSs [16]: *deadlock avoidance* [2, 3, 6, 9, 10-13, 15-19, 22, 23), *deadlock detection and recovery* [4], [5], [8], and *deadlock prevention* [2], [7], [14], [16], [19], [20], [22], [24], [26], [28], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [52], [53], [54], [55], [56]. Graph-based techniques [8], [11], [13], finite state machine-based models [15] and Petri net models [2], [3], [6], [7], [9], [16], [17], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [51], [52], [53], [54], [55], [56] have been utilized for deadlock analysis and control. Owing to their ability to detect desirable behavioral properties of a system such as boundedness and deadlock-freeness, Petri nets (PNs) are extensively employed as an important vehicle to characterize FMSs [16]. The reader is referred to [33] and [45] for the of Petri net basics. In this paper, we deal with the PN based *deadlock prevention*, where a suitable design is obtained before the control implementation of a system.

Behavioral permissiveness, structural complexity, and computational complexity are considered to be three major indicators to design liveness-enforcing supervisors. The legal markings of a Petri net represent the behavioral permissiveness. If all legal markings are reachable, then a supervisor is said to be maximally permissive or optimal. The structural complexity is generally evaluated by the number of control places (CPs) in a supervisor. The fact that whether or not a CP contains weighted input/output arcs can also be considered to be related with the structural complexity. The efficiency of a proposed algorithm to design a supervisor is studied under the computational complexity.

In general, techniques used in the synthesis of deadlock prevention policies can be considered as structural analysis [27], [34], [35], [36] and reachability graph (RG) analysis [37], [38], [39], [40], [41], [42]. However, usually it leads to the fact that the resulting net model is not optimally controlled [43], [44], since the computed supervisor is usually rather conservative, implying that a great deal of legal markings of the system are prevented from being reached. On the other hand, in contrary to structural analysis techniques, the RG analysis based deadlock prevention policies can lead to optimal or near-optimal supervisors for generalized Petri net models. The disadvantage of these methods is that they require a full reachable marking enumeration of a system [37], [45], [46], [47].

The controlled (closed-loop) model is obtained by merging the plant Petri net model and the LES. If a controlled model is optimal, then all CPs are also optimal. However, when the controlled model is suboptimal, some CPs are optimal while the others are not. To improve the permissiveness of a controlled system and/or to reduce the structural complexity of the CPs, it is crucial to identify the set of suboptimal CPs. When dealing with suboptimal controlled models, no attention has been paid to identify both sets of optimal and suboptimal CPs. To our best knowledge to-date this important issue has not been studied in the literature as far as we know.

In this paper, an optimality test is proposed for a liveness enforcing supervisor of an FMS. The method takes an LPN model of an FMS, controlled by a set of CPs, as input and in the case of suboptimal controlled models it produces both sets of optimal CPs and suboptimal CPs. The optimality test is applicable to any LPN consisting of a Petri net model (PNM), controlled by a family of CPs. The feasibility of this method is demonstrated by considering a number of existing FMS examples.

In Section II, Petri net concepts, regarding the analysis of an RG are considered. Section III proposes an optimality test for LES of FMS. In Section IV, a number of examples are considered to show the applicability of the proposed optimality test. A discussion about how the set of suboptimal CPs can be used to improve the behavioral permissiveness or to reduce the structural complexity of an LES is provided in Section V. Conclusions are reached in Section VI.

II. ANALYSIS OF A REACHABILITY GRAPH

The markings (states) in an RG can be classified into four categories: *deadlock*, *bad*, *dangerous* and *good* ones [45]. This section briefly analyzes an RG and shows these four markings as well as the deadlockzone (DZ) and the livezone (LZ) of an RG through an example. Let $R(N, M_0)$ be the RG of a bounded Petri net. For optimal control purposes, both good and dangerous markings in $R(N, M_0)$ must be kept in the controlled system, which serve as the legal markings whose set is denoted by $\mathcal{M}_L$. For a system (N, M_0), the set of its legal markings $\mathcal{M}_L$ is defined as:

$$\mathcal{M}_L=\{M|M \in R(N, M_0)M_0 \wedge \ M_0 \in R(N, M)\}$$

The set $\mathcal{M}_L$ is the maximal set of reachable markings such that it is possible to reach the initial marking M_0 from any legal marking without leaving $\mathcal{M}_L$. When the system is controlled optimally, the legal set of markings M_L, i.e., all dangerous and good markings in RG, constitutes the maximum legal (maximally permissive or optimal) behavior. As shown in [33], DZ contains deadlock and bad markings and the LZ contains all legal markings. To explain these concepts clearly, consider an example as depicted in Fig. 1 [51], with the following production sequences:

$$P1 : M1 \rightarrow Robot \rightarrow M2$$
$$P2 : M1 \leftarrow Robot \leftarrow M2$$

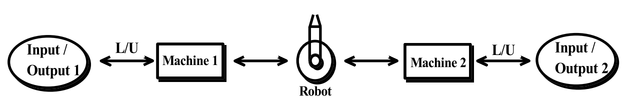

FIGURE 1. An example FMS.

Fig. 2 shows the Petri net model (PNM) of the FMS for these production sequences. In this model there are eleven places, $P = \{p1, p2, \ldots, p11\}$ and eight transitions, $T = \{t1,$

$t2, \ldots, t8\}$. Detailed explanation of this model can be seen from [51].

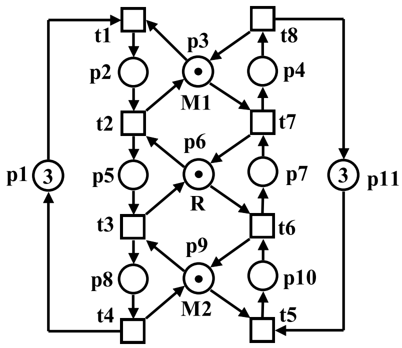

FIGURE 2. Petri net model (N, M_0) of the FMS for the two production sequences.

The RG of the PNM, shown in Fig. 3, contains 20 markings. Table 1 depicts the markings (states) of the RG. M_{13} and M_{14} are deadlock, M_4, M_8, and M_9 are bad, M_1, M_2, M_3, M_5, M_6, and M_{11} are dangerous, and the others are good markings. Illegal markings M_4, M_8, M_9, M_{13} and M_{14} together with illegal transitions $M_4 \xrightarrow{t2}$, $M_4 \xrightarrow{t6}$, $M_8 \xrightarrow{t1}$ and $M_9 \xrightarrow{t5}$ constitute the DZ. Illegal transitions $M_1 \xrightarrow{t5}$, $M_2 \xrightarrow{t1}$, $M_3 \xrightarrow{t5}$, $M_5 \xrightarrow{t1}$, $M6 \xrightarrow{t5}$ and $M_{11} \xrightarrow{t1}$ show the firing of critical system transitions which take the system from LZ to DZ. An optimally controlled system must stop these critical transitions from firing, while allowing all other states to be reachable and all other transitions to be firable. Thus, the LZ represents the optimally controlled (maximally permissive) system behavior. It can be seen from Fig. 3 that for this example, the LZ contains 15 legal markings (states) and 24 legal transitions. Therefore, a live and optimally controlled PNM must be able to reach all legal markings.

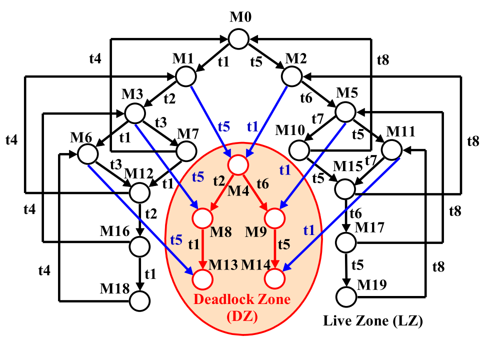

FIGURE 3. The RG of the PNM in Fig. 2.

III. OPTIMALITY TEST FOR LIVENESS ENFORCING SUPERVISORS OF FMS

We assume that a liveness enforcing supervisor (LES), consists of n control places (CPs) $CP = \{C_1, C_2, \ldots, C_n\}$. When an uncontrolled plant Petri net model and the related LES are merged, the controlled (closed-loop) model is obtained. The number of states within the RG of a live controlled model is considered as a quality measure. It represents the possible legal system behavior of the FMS under the supervision of LES. The maximum number of states within the RG of a controlled model can be achieved when the control action of an LES is optimal. Then, in this case the LES is said to be maximally permissive. It can easily be seen that when an LES is maximally permissive, all the CPs in $\{C_1, C_2, \ldots, C_n\}$ of the LES are also optimal. However, when an LES is not maximally permissive, i.e., it enforces liveness in a suboptimal way, it can be concluded that at least one of the CPs in $\{C_1, C_2, \ldots, C_n\}$ of the LES is suboptimal. Although for a maximally permissive LES, it is not necessary to make an optimality test for each CP, since all of them are optimal, an optimality test is very crucial for a suboptimal LES to find out sets of both optimal and suboptimal CPs. The set of suboptimal CPs can be used to improve the LES in terms of the behavioral permissiveness and/or to reduce the structural complexity of the LES as explained in Section V. Algorithm 1 is proposed to carry out an optimality test for an LES.

TABLE 1. Markings (states) of the RG, shown in Fig. 3.

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11
M0:	3	0	1	0	0	1	0	0	1	0	3
M1:	2	1	0	0	0	1	0	0	1	0	3
M2:	3	0	1	0	0	1	0	0	0	1	2
M3:	2	0	1	0	1	0	0	0	1	0	3
M4:	2	1	0	0	0	1	0	0	0	1	2
M5:	3	0	1	0	0	0	1	0	1	0	2
M6:	1	1	0	0	1	0	0	0	1	0	3
M7:	2	0	1	0	0	1	0	1	0	0	3
M8:	2	0	1	0	1	0	0	0	0	1	2
M9:	2	1	0	0	0	0	1	0	1	0	2
M10:	3	0	1	0	0	0	1	0	0	1	1
M11:	3	0	0	1	0	1	0	0	1	0	2
M12:	1	1	0	0	0	1	0	1	0	0	3
M13:	1	1	0	0	1	0	0	0	0	1	2
M14:	2	1	0	0	0	0	1	0	0	1	1
M15:	3	0	0	1	0	1	0	0	0	1	1
M16:	1	0	1	0	1	0	0	1	0	0	3
M17:	3	0	0	1	0	0	1	0	1	0	1
M18:	0	1	0	0	1	0	0	1	0	0	3
M19:	3	0	0	1	0	0	1	0	0	1	0

In this work, a CP is said to be optimal, if it forbids the markings only from DZ but not from LZ. A CP is said to be suboptimal, if it can forbid markings not only from DZ but also from LZ. In Algorithm 1, it is assumed that a live Petri net model (LPN), consisting of a PNM of an FMS, denoted by a net system (N_0, M_0), prone to deadlock, and n control places $CP = \{C_1, C_2, \ldots, C_n\}$ are given as input. Then the set of optimal control places of LPN, O_{CP}, and the set of suboptimal control places of LPN, S_{CP}, are provided as output. First, the reachability graph (RG_0) of the uncontrolled plant Petri net model (N_0, M_0) is computed and both the number of states in LZ_0 and in DZ_0 of RG_0 are also calculated. The number of states in DZ_0 plays an important role in the decision whether or not a CP is optimal. Secondly, the RG of the controlled model consisting of (N_0, M_0) and $CP = \{C_1,$

Algorithm 1 Optimality Test for LES of FMSs

Input: A Live Net Model Consisting of a PNM of an FMS, Denoted by a Net System (N_0, M_0), Prone to Deadlock, and n Control Places $CP = \{C_1, C_2, \ldots, C_n\}$.

Output: The Set of Optimal Control Places of LPN and the Set of Suboptimal Control Places of LPN.

1) [Define] O_{CP}: The Set of Optimal Control Places of LPN,
   [Define] S_{CP}: The Set of Suboptimal Control Places of LPN.
2) [Initialize] $O_{CP} = \{\}$; $S_{CP} = \{\}$;
3) Check the Liveness Property of (N_0, M_0). Compute the Reachability Graph (RG_0) of (N_0, M_0). Calculate Both the Number of States in LZ_0 and in DZ_0 of RG_0.
4) Check the Liveness Property of the Controlled Model Consisting of (N_0, M_0) and $CP = \{C_1, C_2, \ldots, C_n\}$. Compute the RG of the Controlled Model and Calculate the Number of States in the LZ_{cm}.
5) *If* the Number of States in LZ_{cm} = the Number of States in LZ_0
   then All Control Places $CP = \{C_1, C_2, \ldots, C_n\}$ Are Optimal,
   Therefore $O_{CP} = \{C_1, C_2, \ldots, C_n\}$; $S_{CP} = \{\}$;
   Goto Step 7.
   else Carry on With the Next Step,
6) for ($i = 1, i \leq n, i++$)
   6.i.1) Add C_i to (N_0, M_0). Denote the Resulting Net System by (N_i, M_i).
   6.i.2) Check the Liveness Property of (N_i, M_i). Compute the Reachability Graph (RG_i) of (N_i, M_i). Calculate Both the Number of States in LZ_i and in DZ_i of RG_i.
   If the Number of States in LZ_i = the Number of States in $LZ_.$
   then the Considered C_i Is an Optimal CP, Therefore Put It in O_{CP},
   else the Considered C_i Is a Suboptimal CP, Therefore Put It in S_{CP},
   endif
   End for
   endif
7) Output O_{CP} and S_{CP}
8) End.

$C_2, \ldots, C_n\}$ is computed and the number of states in the LZ_{cm} is calculated. It can then be seen whether or not the controlled model is maximally permissive. If the number of states in the LZ_{cm} is equal to the number of states in the LZ_0, then the controlled model is maximally permissive (optimal). As a result, we can conclude that all CPsin set $\{C_1, C_2, \ldots, C_n\}$ are optimal. If the number of states in the LZ_{cm} is less than the number of states in the LZ_0, then the controlled model is not maximally permissive, i.e., it is suboptimal. When the controlled model is not maximally permissive, at least one of the control places is suboptimal. To find out sets of optimal and suboptimal CPs, for each control place C_i following procedure is conducted: A partially controlled net (N_i, M_i) is obtained by adding C_i to (N_0, M_0). Then, the reachability graph (RG_i) of (N_i, M_i) is computed and both the number of states in LZ_i and in DZ_i of RG_i are calculated. At this point it is important to understand the meaning of the number of states in LZ_i of the partially controlled net (N_i, M_i). This number shows the total number of legal states that are survived under the supervision of C_i. When C_i is optimal, no legal states of RG_0 will be prevented from being reached. On the other hand, when C_i is suboptimal, the supervision of C_i is too restrictive and therefore some legal states of RG_0 will not be reachable. Next, the decision about whether or not C_i is optimal is made by comparing the number of states in LZ_i and the number of states in LZ_0. If the number of states in LZ_i is equal to the number of states in LZ_0, then this indicates that all legal states of RG_0 can be reached under the supervision of C_i. Therefore, C_i is optimal. However, if the number of states in LZ_i is less than the number of states in LZ_0, then this indicates that some legal states of RG_0 cannot be reached under the supervision of C_i. Therefore, C_i is suboptimal. Finally, based on this reasoning the set of optimal control places O_{CP} and the set of suboptimal control places S_{CP} of LPN are obtained. Technically, INA [48] can be used for the computation of RGs. Then, by using the computed RGs obtained from INA, a recently proposed method [49] can be utilized for the computation of LZ and DZ on the RGs. In this work, a much faster tool called TINA (TIme petri Net Analyzer) [50] is used to obtain RGs and LZ and DZ of RGs.

IV. EXAMPLES

In this section let us first consider an illustrative example to show the details of the optimality test. Then, a number of FMS deadlock prevention examples are considered to show its' applicability. Note that normally Step 6 of Algorithm 1 is skipped, when the controlled Petri net model is maximally permissive, However, in this section, whenever applicable all steps are provided for all LESs to make some discussions in Section V based on the results obtained.

A. ILLUSTRATIVE EXAMPLE

As a first example, the Petri net model (PNM) shown in Fig. 2 is considered. As explained in Section II it suffers from deadlocks. Its' LZ and DZ contain 15 and 5 states respectively. In this section two different sets of LESs will be considered for this example PNM. The first LES has three CPs as depicted in Table 2. When we add the LES, i.e., the three CPs depicted in Table 2, to the PNM, shown in Fig. 2, an LPN model of the FMS is obtained as shown in Fig. 4. The PN model shown in Fig. 4 is live with 15 legal states. The RG of LPN model is depicted in Fig. 5. It represents the optimally controlled system behavior. Table 3 shows the markings of this RG. Since the system controlled optimally, all three CPs are optimal. Next, the proposed optimality test is applied to this LES.

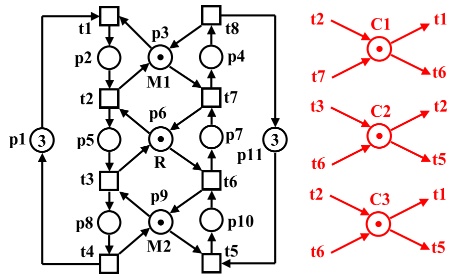

FIGURE 4. An LPN model of the FMS, consisting of the uncontrolled PNM (N_0,M_0) and three CPs depicted in Table 2.

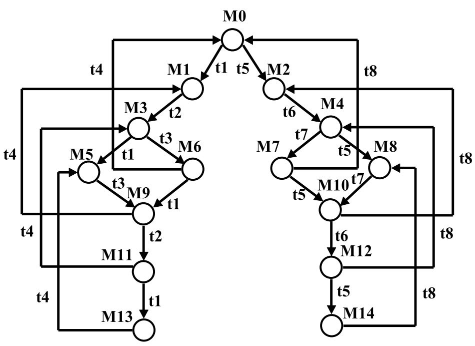

FIGURE 5. The RG of LPN model, shown in Fig. 4, representing the optimally controlled system behavior.

TABLE 2. Three *CP*s obtained for the uncontrolled *PNM*, shown in Fig. 2.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	t2, t7	t1, t6	1
2	t3, t6	t2, t5	1
3	t2, t6	t1, t5	1

TABLE 3. Markings of the *RG*, shown in Fig. 5.

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	C1	C2	C3
M0:	3	0	1	0	0	1	0	0	1	0	3	1	1	1
M1:	2	1	0	0	0	1	0	0	1	0	3	0	1	0
M2:	3	0	1	0	0	1	0	0	0	1	2	1	0	0
M3:	2	0	1	0	1	0	0	0	1	0	3	1	0	1
M4:	3	0	1	0	0	0	1	0	1	0	2	0	1	1
M5:	1	1	0	0	1	0	0	0	1	0	3	0	0	0
M6:	2	0	1	0	0	1	0	1	0	0	3	1	1	1
M7:	3	0	0	1	0	1	0	0	1	0	2	1	1	1
M8:	3	0	1	0	0	0	1	0	0	1	1	0	0	0
M9:	1	1	0	0	0	1	0	1	0	0	3	0	1	0
M10:	3	0	0	1	0	1	0	0	0	1	1	1	0	0
M11:	1	0	1	0	1	0	0	1	0	0	3	1	0	1
M12:	3	0	0	1	0	0	1	0	1	0	1	0	1	1
M13:	0	1	0	0	1	0	0	1	0	0	3	0	0	0
M14:	3	0	0	1	0	0	1	0	0	1	0	0	0	0

Algorithm 1 Optimality Test Applied to the First LES

Input: A live Petri net model, LPN shown in Fig. 4, consisting of an uncontrolled PNM (N_0, M_0), prone to deadlock, and $n = 3$ control places $CP = \{C_1, C_2, C_3\}$.

1) Let O_{CP} *be the set of optimal control places of LPN, and* S_{CP} *be the set of suboptimal control places of LPN.*

2) Initially both sets are empty: $O_{CP} = \{\}$, $S_{CP} = \{\}$.

3) *Check* the liveness property of (N_0, M_0): The uncontrolled PNM (N_0, M_0), is not live. *Compute* the reachability graph (RG_0) of (N_0, M_0): RG_0, shown in Fig. 3 contains 20 markings. *Compute* both the number of states in LZ_0 and in DZ_0 of RG_0: The number of states in LZ_0 and DZ_0 are 15 and 5, respectively.

4) *Check* the liveness property of the controlled model consisting of (N_0, M_0) and $CP = \{C_1, C_2, C_3\}$. The controlled model is live with 15 legal states, i.e., the number of states in $LZ_{cm} = 15$.

5) The number of states in LZ_{cm} = the number of states in LZ_0, all CPs are optimal.

The number of states in LZ_{cm} = the number of states in LZ_0.

Therefore, all *CP*s are optimal: $O_{CP} = \{C_1, C_2, C_3\}$, $S_{CP} = \{\}$.

Goto Step 7

6) for ($i = 1, i \leq n, i++$)

6.1 ________ 1st iteration ___________________________

($i = 1, i \leq n$)

6.1.1) C_1 is added to (N_0, M_0), and the resultant net system is denoted by (N_1, M_1).

6.1.2) (N_1, M_1) is not live. RG1 contains 18 states (markings).

The number of states in LZ1 and DZ1 are 15 and 3, respectively.

Since the number of states in LZ1 = the number of states in LZ0

The considered C1 is an optimal control place, therefore $O_{CP} = \{C_1\}$.

$i = i + 1 = 2$

6.2 ________ 2nd iteration ___________________________

($i = 2$, i $\leq n$)

6.2.1) C_2 is added to (N_0, M_0), and the resultant net system is denoted by (N_2, M_2).

6.2.2)(N_2, M_2) is not live. RG_2 contains 18 states (markings).

The number of states in LZ_2 and DZ_2 are 15 and 3, respectively.

Since the number of states in LZ_2 = the number of states in LZ_0, the considered C_2 is an optimal control place; therefore $O_{CP} = \{C_1, C_2\}$.

$i = i + 1 = 3$

6.3 ________ 3rd iteration ___________________________

($i = 3, i \leq n$)

6.3.1) C_3 is added to (N_0, M_0), and the resultant net system is denoted by (N_3, M_3).

5.3.2)(N_3, M_3) is not live. RG_3 contains 17 states.

The number of states in LZ_3 and DZ_3 are 15 and 2, respectively.

Since the number of states in LZ_3 = the number of states in LZ_0 the considered C_3 is an optimal control place, therefore $O_{CP} = \{C_1, C_2, C_3\}$.

$i = i + 1 = 4$

____

($i = 4, i > n$)

end for

7) Output: $O_{CP} = \{C_1, C_2, C_3\}$, $S_{CP} = \{\}$.

8) End.

The optimality test performed above is summarized in Table 4. In this example as all control places are optimal, the controlled model is also optimal, i.e., maximally permissive.

Let us now consider the 2nd set of control places, depicted in Table 5. When two CPs, i.e. the LES, depicted in Table 5, are added to the PNM, shown in Fig. 2, a live Petri net (LPN) model of the FMS is obtained as shown in Fig. 6. The LPN model depicted in Fig. 4 is live with 13 legal markings. The RG of LPN model is depicted in Fig. 7. Table 6 shows the markings of this RG. Since the live controlled system can reach only 13 legal markings it can be seen that at least one of the CPs is suboptimal. When we compare the RG shown in Fig. 5 which represents the optimally controlled system

TABLE 4. Optimality test conducted for the three CPs depicted in Table 2.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in			Is the considered *CP* optimal?
					RG	*LZ*	*DZ*	
		-	-	NO	*20*	15	5	
IN	*PNM* shown in Fig. 2.	*C1, C2, C3*	-	*YES*	15	15	-	-
1		*C1*	*C1*	*NO*	*18*	*15*	*3*	*YES*
2		*C2*	*C2*	*NO*	*18*	*15*	*3*	*YES*
3		*C3*	*C3*	*NO*	*17*	*13*	*2*	*YES*
OUT	*Optimal CP: C1, C2 C3*			*Suboptimal CPs: -*				

behavior and the RG shown in Fig. 7, it can be realized that two good markings of the optimally controlled system behavior cannot be reached in the RG shown in Fig. 7.

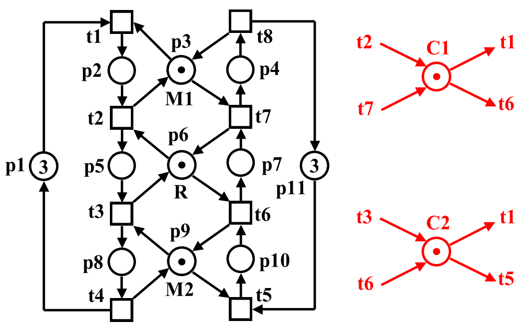

FIGURE 6. A live Petri net (LPN) model of the FMS, consisting of the uncontrolled PNM (N_0,M_0) and two CPs depicted in Table 2.

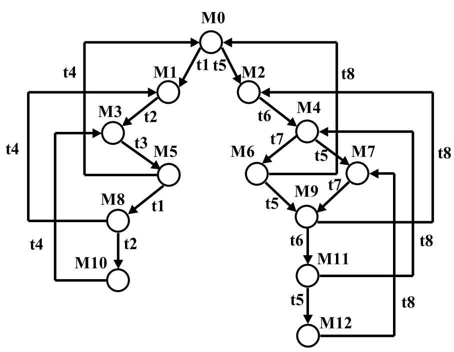

FIGURE 7. The RG of LPN model, shown in Fig. 6.

TABLE 5. Two CPs obtained for the uncontrolled PNM, shown in Fig. 2.

C_i	$^{\bullet}C_i$	$C_i^{\bullet}$	$\mu_0(C_i)$
1	t2, t7	t1, t6	1
2	t3, t6	t1, t5	1

Now the proposed optimality test is applied to the second LES:

The optimality test performed above is summarized in Table 7. In this example it can be seen that C_1 is an optimal CP, while C_2 is a suboptimal one. As one of the CPs is suboptimal, the controlled model is also suboptimal.

Algorithm 1 Optimality Test Applied to the Second LES

Input: A live Petri net model, LPN shown in Fig. 4, consisting of an uncontrolled PNM (N_0, M_0), prone to deadlock, and n = *2* control places $CP = \{C_1, C_2\}$.

1) Let O_{CP} *be the set of optimal control places of LPN, and* S_{CP} *be the set of suboptimal control places of LPN.*

2) Initially both sets are empty: $O_{CP} = \{\}$, $S_{CP} = \{\}$.

3) *Check* the liveness property of (N_0, M_0): The uncontrolled PNM (N_0, M_0), is not live. *Compute* the reachability graph (RG_0) of (N_0, M_0): RG_0, shown in Fig. 3 contains 20 markings. *Compute* both the number of states in LZ_0 and in DZ_0 of RG_0: The number of states in LZ_0 and DZ_0 are 15 and 5, respectively.

4) Check the liveness property of the controlled model consisting of (N_0, M_0) and $CP = \{C_1, C_2\}$. The controlled model is live with 13 legal states, i.e., the number of states in LZ_{cm} = 13.

5) Since the number of states in LZ_{cm} < the number of states in LZ_0, at least one CP is suboptimal. Carry on with the next step.

6) for ($i = 1$, $i \leq n$, i++)

6.1 ________ 1st iteration ____________________________

($i = 1$, i $\leq n$)

6.1.1) C_1 is added to (N_0, M_0), and the resultant net system is denoted by (N_1, M_1).

6.1.2)(N_1, M_1) is not live. RG_1 contains 18 states (markings).

The number of states in LZ_1 and DZ_1 are 15 and 3, respectively.

Since the number of states in LZ_1 = the number of states in LZ_0,

the considered C_1 is an optimal control place, therefore O_{CP} = *{*C_1*}*.

$i = i + 1 = 2$

6.2 ________ 2nd iteration ____________________________

($i = 2$, i $\leq n$)

6.2.1) C_2 is added to (N_0, M_0), and the resultant net system is denoted by (N_2, M_2).

6.2.2)(N_2, M_2) is not live. RG_2 contains 14 states (markings).

The number of states in LZ_2 and DZ_2 are 13 and 1, respectively.

Since the number of states in $LZ_2 \neq$ the number of states in LZ_0,

the considered C_2 is a suboptimal control place, therefore S_{CP} =$\{C_2\}$.

$i = i + 1 = 3$

($i = 3$, $i > n$)

end for

7) Output: O_{CP} = *{*C_1*}*, S_{CP} = *{*C_2*}*.

8) End.

TABLE 6. Markings of the RG, shown in Fig. 7.

	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10	p11	C1	C2
M0:	3	0	1	0	0	1	0	0	1	0	3	1	1
M1:	2	1	0	0	0	1	0	0	1	0	3	0	0
M2:	3	0	1	0	0	1	0	0	0	1	2	1	0
M3:	2	0	1	0	1	0	0	0	1	0	3	1	0
M4:	3	0	1	0	0	0	1	0	1	0	2	0	1
M5:	2	0	1	0	0	1	0	1	0	0	3	1	1
M6:	3	0	0	1	0	1	0	0	1	0	2	1	1
M7:	3	0	1	0	0	0	1	0	0	1	1	0	0
M8:	1	1	0	0	0	1	0	1	0	0	3	0	0
M9:	3	0	0	1	0	1	0	0	0	1	1	1	0
M10:	1	0	1	0	1	0	0	1	0	0	3	1	0
M11:	3	0	0	1	0	0	1	0	1	0	1	0	1
M12:	3	0	0	1	0	0	1	0	0	1	0	0	0

TABLE 7. Optimality test conducted for the two CPs depicted in Table 2.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in			Is the considered *CP* optimal?
					RG	*LZ*	*DZ*	
	PNM	-	-	NO	*20*	15	5	
IN	shown	*C1, C2*	-	*YES*	13	13	-	-
1	in Fig.	***C1***	***C1***	***NO***	***18***	***15***	***3***	***YES***
2	2.	*C2*	*C2*	*NO*	*14*	*13*	*1*	*NO*
OUT	*Optimal CP: C1*			*Suboptimal CP: C2*				

B. S^3PR NETS

For a class of Petri net models of FMSs, called an S^3PR [7], deadlock prevention policies were proposed in [7], [21], [27], and [28]. A conceptual FMS depicted in Fig. 8(a), is taken from [7]. The production cycles are as shown in Fig. 8(b).

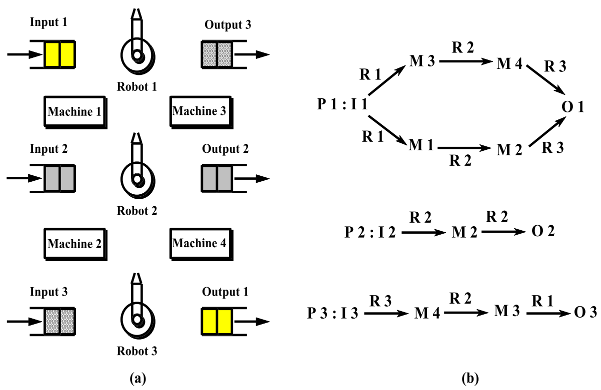

FIGURE 8. (a) An FMS layout and (b) Production cycles for FMS.

Fig. 9 shows the S^3PR model of the system [7]. For the detailed explanation of this model the reader is referred to [7]. The uncontrolled S^3PR model shown in Fig. 9 suffers from deadlock. The RG of the uncontrolled S^3PR model has 26750 states, whose LZ and DZ contain 21581 and 5169 states respectively.

We will consider five different sets of LESs for this problem. The first LES is from [7], where eighteen CPs, depicted in Table 8, are obtained to prevent deadlocks in the S^3PR model. The controlled S^3PR model is live with 6287 good states. By using the redundancy test proposed in [51], it can be shown that, only six control places, namely *C1, C2, C3, C4, C7, C16*, are necessary.

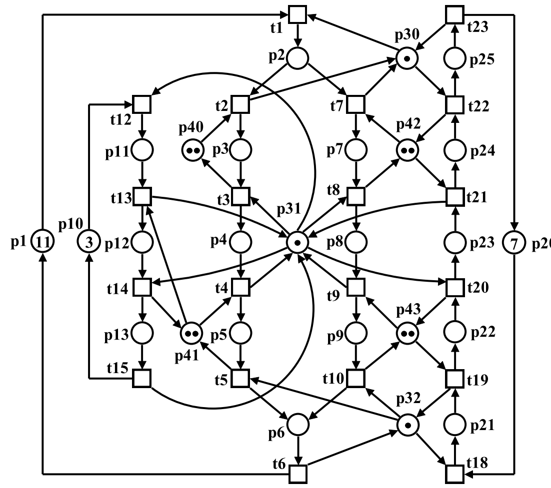

FIGURE 9. The S^3PR model of the FMS taken from [7].

TABLE 8. 18 CPs, obtained in [7] for the S^3PR model.

C	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	**t2, t10, t19**	**t1, t18**	**2**
2	**t2, t8, t21**	**t1, t18**	**2**
3	**t3, t8, t22**	**t1, t18**	**5**
4	**t4, t7, t14**	**t1, t12**	**2**
5	t4, t8, t14, t21	t1, t12, t18	4
6	t4, t8, t14, t22	t1, t12, t18	7
7	**t2, t9, t20**	**t1, t18**	**2**
8	t2, t9, t21	t1, t18	4
9	t2, t9, t22	t1, t18	7
10	t4, t9, t14, t20	t1, t12, t18	4
11	t4, t9, t14, t21	t1, t12, t18	6
12	t4, t9, t14, t22	t1, t12, t18	9
13	t2, t10, t20	t1, t18	3
14	t2, t10, t21	t1, t18	5
15	t2, t10, t22	t1, t18	8
16	**t5, t10, t14, t20**	**t1, t12, t18**	**5**
17	t5, t10, t14, t21	t1, t12, t18	7
18	t5, t10, t14, t22	t1, t12, t18	10

Table 9 shows the results of the proposed optimality test conducted for the six necessary CPs depicted in Table 8. It can be seen that all six necessary CPs are suboptimal.

The second LES is from [30], where sixteen CPs, depicted in Table 10, are obtained to prevent deadlocks in the S^3PR model. The controlled S^3PR model is live with 12656 good states. By using the redundancy test proposed in [51], it can be shown that, only seven control places, namely *C1, C4, C12, C13, C14, C15, C16*, are necessary. Table 11 shows the results of the proposed optimality test conducted for the seven necessary CPs depicted in Table 10. It can be seen that *C16* is suboptimal, while the rest of necessary CPs, i.e., *C1, C4, C12, C13, C14, C15*, are optimal.

The third LES is from [32] and [33], where 19 CPs, depicted in Table 12, are obtained to prevent deadlocks in the S^3PR model. The controlled S^3PR model is live with 21562 good states. By using the redundancy test proposed

TABLE 9. Optimality test conducted for the six necessary CPs depicted in Table 8.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in *RG*	*LZ*	*DZ*	Is the considered *CP* optimal?
	S^3PR model shown in Fig. 9.	-	-	*NO*	*26750*	*21581*	5169	
IN		*C1-4, C7, C16*	-	*YES*	6287	6287	-	-
1		*C1*	*C1*	*NO*	17578	*15435*	*2143*	*NO*
2		*C2*	*C2*	*NO*	16506	*14418*	*2088*	*NO*
3		*C3*	*C3*	*NO*	22968	*19169*	*3799*	*NO*
4		*C4*	*C4*	*NO*	14568	*12733*	*1835*	*NO*
5		*C7*	*C7*	*NO*	16587	*14640*	*1947*	*NO*
6		*C16*	*C16*	*NO*	13976	*12994*	*982*	*NO*
OUT	*Optimal CPs:* -				*Suboptimal CPs: C1-4, C7, C16*			

TABLE 10. 16 CPs obtained in [30] for the S^3PR model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	**t5, t10, t14, t20**	**t3, t8, t12, t18**	**5**
2	t3, t10, t22	t1, t18	8
3	t4, t9, t14, t20	t3, t8, t12, t19	4
4	**t3, t8, t22**	**t1, t20**	**5**
5	t3, t9, t22	t1, t19	7
6	t10, t21	t7, t18	5
7	t10, t20	t8, t18	3
8	t4, t8, t14, t21	t3, t7, t12, t20	4
9	t4, t8, t14, t22	t1, t12, t20	7
10	t9, t21	t7, t19	4
11	t4, t9, t14, t21	t3, t7, t12, t19	6
12	**t9, t20**	**t8, t19**	**2**
13	**t8, t21**	**t7, t20**	**2**
14	**t4, t14**	**t3, t12**	**2**
15	**t10, t19**	**t9, t18**	**2**
16	**t2, t5, 2t10, 2t20**	**2t1, 2t18**	**7**

TABLE 11. Optimality test conducted for the seven necessary CPs depicted in Table 10.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in *RG*	*LZ*	*DZ*	Is the considered *CP* optimal?
	PNM in Fig. 8.	-	-	*NO*	*26750*	*21581*	5169	
IN		*C1, C4, C12-16*	-	*YES*	12656	12656	-	-
1		*C1*	*C1*	*NO*	25577	*21581*	*3996*	*YES*
2		*C4*	*C4*	*NO*	26498	*21581*	*4917*	*YES*
3		*C12*	*C12*	*NO*	26049	*21581*	*4468*	*YES*
4		*C13*	*C13*	*NO*	26246	*21581*	*4665*	*YES*
5		*C14*	*C14*	*NO*	25019	*21581*	*3438*	*YES*
6		*C15*	*C15*	*NO*	25721	*21581*	*4140*	*YES*
7		***C16***	***C16***	***NO***	**14016**	***12656***	***1360***	***NO***
OUT	*Optimal CPs: C1, C4, C12-15*				*Suboptimal CP: C16*			

in [52], it can be shown that, *C10* and *C12* are redundant and the remaining 17 CPs are necessary. Table 13 shows the results of the proposed optimality test conducted for the

TABLE 12. 19 CPs obtained in [32], [33] for the S^3PR model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	t14	t12	2
2	t4, t14	t3, t13	2
3	t8, t21	t7, t20	2
4	t9, t20	t8, t19	2
5	t8, t20	t7, t19	3
6	t10, t19	t9, t18	2
7	t9, t20	t7, t18	4
8	t10, t20	t8, t18	3
9	t8, t10, t20	t7, t9, t18	4
10	**t5, t14, t20**	**t4, t12, t18**	**5**
11	t3, t8, t22	t1, t20	5
12	**t5, t14, t20**	**t3, t13, t18**	**5**
13	t5, t8, t14, t20	t4, t7, t12, t18	6
14	t5, t10, t14, t20	t4, t9, t12, t18	5
15	t5, t8, t14, t20	t3, t7, t13, t18	6
16	t5, t10, t14, t20	t3, t9, t13, t18	5
17	t5, t8, t10, t20, t22	t1, t9, t18, t21	9
18	t3, t5, t8, t10, t21	t1, t4, t9, t18	9
19	t3, t5, t9, t20, t22	t1, t4, t18, t21	9

TABLE 13. 19 CPs obtained in [32], [33] for the S^3PR model.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in *RG*	*LZ*	*DZ*	Is the considered *CP* optimal?
	PNM shown in Fig. 8.	-	-	*NO*	*26750*	*21581*	5169	
IN		*C1-9, C11, C13-19*	-	*YES*	21562	21562	-	-
1		*C1*	*C1*	*NO*	25868	*21581*	*4287*	*YES*
2		*C2*	*C2*	*NO*	25901	*21581*	*4320*	*YES*
3		*C3*	*C3*	*NO*	26246	*21581*	*4665*	*YES*
4		*C4*	*C4*	*NO*	26049	*21581*	*4468*	*YES*
5		*C5*	*C5*	*NO*	26174	*21581*	*4593*	*YES*
6		*C6*	*C6*	*NO*	25721	*21581*	*4140*	*YES*
7		*C7*	*C7*	*NO*	26318	*21581*	*4737*	*YES*
8		*C8*	*C8*	*NO*	26066	*21581*	*4485*	*YES*
9		*C9*	*C9*	*NO*	26192	*21581*	*4611*	*YES*
10		*C11*	*C11*	*NO*	26498	*21581*	*4917*	*YES*
11		*C13*	*C13*	*NO*	26522	*21581*	*4941*	*YES*
12		*C14*	*C14*	*NO*	26327	*21581*	*4746*	*YES*
13		*C15*	*C15*	*NO*	26525	*21581*	*4944*	*YES*
14		*C16*	*C16*	*NO*	26342	*21581*	*4761*	*YES*
15		***C17***	***C17***	***NO***	**26654**	***21562***	***5092***	***NO***
16		*C18*	*C18*	*NO*	26720	*21581*	*5139*	*YES*
17		*C19*	*C19*	*NO*	26718	*21581*	*5137*	*YES*
OUT	*Optimal CPs: C1-9, C11, C13-16, C18, C19*						*Suboptimal CP: C17*	

seventeen necessary CPs depicted in Table 12. It can be seen that *C17* is suboptimal, while the rest of necessary CPs, i.e., *C1-9, C11, C13-16, C18, C19*, are optimal.

The fourth LES is from [52], where thirteen CPs, depicted in Table 14, are obtained to prevent deadlocks in the S^3PR model. The controlled S^3PR model is live and can reach 21581 legal states. This means that that all thirteen CPs are optimal. Table 15 shows the results of the proposed optimality test conducted for the 13 CPs depicted in Table 14. It can be seen that all CPs are optimal.

TABLE 14. 13 CPs obtained in [52] for the S^{3}PR model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	t10, t19	t9, t18	2
2	t5, t10, t14, t20	t3, t8, t12, t18	5
3	t4, t14	t3, t12	2
4	t9, t20	t8, t19	2
5	t8, t21	t7, t20	2
6	t3, t8, t22	t1, t20	5
7	t10, t20	t8, t18	3
8	2t8, t21	2t7, t19	5
11	t3, t5, t8, 2t10, t20, 2t22	2t1, t9, 2t18, t21	17
12	3t8, t10, 2t21	4t7, 2t18	12
15	t3, 2t5, t8, 3t10, t14, t21, 2t22	3t1, t9, t12, 3t18	27
16	t3, 2t5, t9, 2t10, t14, t20, 3t22	3t1, t12, 3t18, t21	27
17	t5, 2t8, t14, t21	t3, 2t7, t12, t18	8

Note: Some original transition numbers of [52] are renamed in this table to comply with the transition numbers of the uncontrolled S^{3}PR model shown in Fig. 9.

TABLE 15. Optimality test conducted for the 13 CPs depicted in Table 14.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in			Is the considered *CP* optimal?
					RG	*LZ*	*DZ*	
	S^{3}PR model shown in Fig. 9.	-	-	*NO*	*26750*	*21581*	5169	
IN		*C1-8, C11, C12, C15, C16, C17*	-	*YES*	*21581*	*21581*	-	-
1		*C1*	*C1*	*NO*	*25721*	*21581*	*4140*	*YES*
2		*C2*	*C2*	*NO*	*25577*	*21581*	*3996*	*YES*
3		*C3*	*C3*	*NO*	*25019*	*21581*	*3438*	*YES*
4		*C4*	*C4*	*NO*	*26049*	*21581*	*4468*	*YES*
5		*C5*	*C5*	*NO*	*26246*	*21581*	*4665*	*YES*
6		*C6*	*C6*	*NO*	*26498*	*21581*	*4917*	*YES*
7		*C7*	*C7*	*NO*	*26066*	*21581*	*4485*	*YES*
8		*C8*	*C8*	*NO*	*25976*	*21581*	*4395*	*YES*
9		*C11*	*C11*	*NO*	*26674*	*21581*	*5093*	*YES*
10		*C12*	*C12*	*NO*	*26030*	*21581*	*4449*	*YES*
11		*C15*	*C15*	*NO*	*26668*	*21581*	*5087*	*YES*
12		*C16*	*C16*	*NO*	*26666*	*21581*	*5085*	*YES*
13		*C17*	*C17*	*NO*	*26426*	*21581*	*4845*	*YES*
OUT	*Optimal CPs: C1-8, C11, C12, C15, C16, C17*				*Suboptimal CP: -*			

Note: Some original transition numbers of [52] are renamed in this table to comply with the transition numbers of the uncontrolled S^{3}PR model shown in Fig. 9.

The fifth and last LES is from [45], where 17 CPs, depicted in Table 16, are obtained to prevent deadlocks in the S^{3}PR model. The controlled S^{3}PR model is live and can reach 21581 legal states. This means that all 17 CPs are optimal. Table 17 shows the results of the proposed optimality test conducted for the 17 CPs depicted in Table 16. It can be seen that all CPs are optimal.

Note: Some original transition numbers of [45] are renamed in this table to comply with the transition numbers of the uncontrolled S^{3}PR model shown in Fig. 9.

C. S^{4}PR NETS

Fig. 10 shows an S^{4}PR model of an FMS, taken from [20]. The RG of the uncontrolled S^{4}PR model has 54869 states,

TABLE 16. CPs obtained in [45] for the S^{3}PR model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	t14	t12	2
2	t4, t14	t3, t13	2
3	t8, t21	t7, t20	2
4	t9, t20	t8, t19	2
5	t10, t19	t9, t18	2
6	t10, t20	t8, t18	3
7	t8, t20	t7, t19	3
8	t9, t20	t7, t18	4
9	t8, t10, t20	t7, t9, t18	4
10	t5, t10, t14, t20	t4, t9, t12, t18	5
11	t5, t10, t14, t20	t3, t9, t13, t18	5
12	t3, t8, t22	t1, t20	5
13	t5, t8, t14, t20	t4, t7, t12, t18	6
14	t5, t8, t14, t20	t3, t7, t13, t18	6
15	t5, t8, 2t10, 2t20, t22	t1, 2t9, 2t18, t21	12
16	t3, t5, t8, t10, t21	t1, t4, t9, t18	9
17	t3, t5, t9, t20, t22	t1, t4, t18, t21	9

Note: Some original transition numbers of [45] are renamed in this table to comply with the transition numbers of the uncontrolled S^{3}PR model shown in Fig. 9.

TABLE 17. Optimality test conducted for the 17 CPs depicted in Table 16.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in			Is the considered *CP* optimal?
					RG	*LZ*	*DZ*	
	S^{3}PR model shown in Fig. 9.	-	-	*NO*	*26750*	*21581*	5169	
IN		*C1-17*	-	*YES*	*21581*	*21581*	-	-
1		*C1*	*C1*	*NO*	25868	*21581*	*4287*	*YES*
2		*C2*	*C2*	*NO*	25901	*21581*	*4320*	*YES*
3		*C3*	*C3*	*NO*	26246	*21581*	*4665*	*YES*
4		*C4*	*C4*	*NO*	26049	*21581*	*4468*	*YES*
5		*C5*	*C5*	*NO*	25721	*21581*	*4140*	*YES*
6		*C6*	*C6*	*NO*	26066	*21581*	*4485*	*YES*
7		*C7*	*C7*	*NO*	26174	*21581*	*4593*	*YES*
8		*C8*	*C8*	*NO*	26318	*21581*	*4737*	*YES*
9		*C9*	*C9*	*NO*	26192	*21581*	*4611*	*YES*
10		*C10*	*C10*	*NO*	26327	*21581*	*4746*	*YES*
11		*C11*	*C11*	*NO*	26342	*21581*	*4761*	*YES*
12		*C12*	*C12*	*NO*	26498	*21581*	*4917*	*YES*
13		*C13*	*C13*	*NO*	26522	*21581*	*4941*	*YES*
14		*C14*	*C14*	*NO*	26525	*21581*	*4944*	*YES*
15		*C15*	*C15*	*NO*	26676	*21562*	*5095*	*YES*
16		*C16*	*C16*	*NO*	26720	*21581*	*5139*	*YES*
17		*C17*	*C17*	*NO*	26718	*21581*	*5137*	*YES*
OUT	*Optimal CPs: C1, C2, ..., C17*				*Suboptimal CP: -*			

whose LZ and DZ contain 51506 and 3363 states, respectively.

We will consider four different sets of LESs for this problem. The first LES is from [20], where five CPs, depicted in Table 18, are obtained to prevent deadlocks in the S^{4}PR model. The controlled S^{4}PR model is live with 51386 good states. Table 19 shows the results of the proposed optimality test conducted for the five CPs depicted in Table 18. It can be seen that *C1*, *C2* and *C4* are optimal CPs, while *C3* and *C5* are suboptimal.

The second LES is from [33], where eight CPs, depicted in Table 20, are obtained to prevent deadlocks in the S^{4}PR

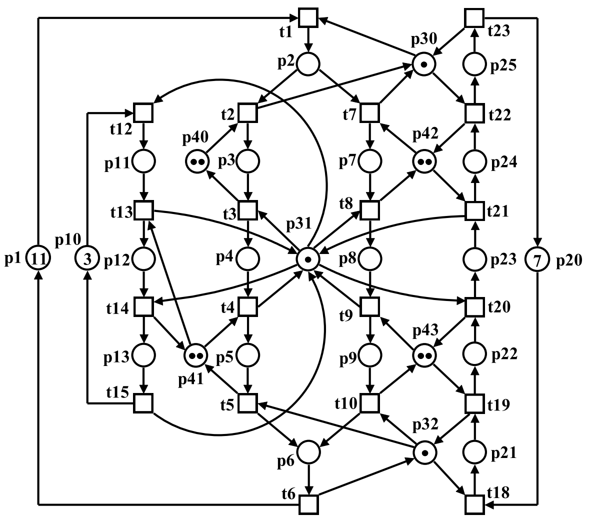

FIGURE 10. The S^3PR model of the FMS taken from [7].

TABLE 18. 5 CPs obtained in [20] for the S^4PR model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	2t2	2t1	3
2	5t4, 3t9, 4t11	2t1, 3t2, 3t8, t10	23
3	2t4, 3t5, 3t9, 4t11	2t1, 3t2, t7, 2t8, t10	26
4	3t5, t8	3t4, t7	23
5	6t5, 3t8	2t1, 3t2, t4, 3t7	49

TABLE 19. Optimality test conducted for the 5 CPs depicted in Table 18.

i	net	included *CP*	considered *CP*	Is the net LIVE ?	The number of states in			Is the considered *CP* optimal?
					RG	*LZ*	*DZ*	
	S^4PR model shown in Fig. 10.	-	-	*NO*	*54869*	*51506*	*3363*	
IN		*C1-5*	-	*YES*	*51386*	*51386*	-	-
1		*C1*	*C1*	*NO*	*51666*	*51506*	*160*	*YES*
2		*C2*	*C2*	*NO*	*54634*	*51506*	*3128*	*YES*
3		***C3***	***C3***	***NO***	***54553***	***51388***	***3165***	***NO***
4		*C4*	*C4*	*NO*	*54864*	*51506*	*3358*	*YES*
5		***C5***	***C5***	***NO***	***54846***	***51492***	***3354***	***NO***
OUT	*Optimal CPs:*	*C1, C2, C4*			*Suboptimal CPs: C3, C5*			

TABLE 20. CPs obtained in [33] for the S^4PR model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	t2	t1	1
2	t3, t9, t11	t2, t8	5
3	t4, t9, t11	t3, t8	5
4	t3, t5, t8, t11	t2, t4, t7, t10	8
5	t3, t5, t9, t11	t2, t4, t7	9
6	t5, t9, t10	t3, t7	9
7	t5, t8, t11	t3, t7, t10	8
8	t5, t9, t11	t3, t7	9

model. The controlled S^4PR model is live with 48752 good states.

Table 21 shows the results of the proposed optimality test conducted for the eight CPs depicted in Table 20. It can be seen that *C1* is an optimal CP, while the rest of the CPs, i.e., *C2, …, C8*, are suboptimal.

The third LES is from [53], where eight CPs, depicted in Table 22, are obtained to prevent deadlocks in

TABLE 21. Optimality test conducted for the 8 CPs depicted in Table 20.

i	net	included *CP*	considered *CP*	Is the net LIVE ?	The number of states in			Is the considered *CP* optimal ?
					RG	*LZ*	*DZ*	
	S^4PR model shown in Fig. 10.	-	-	*NO*	*54869*	*51506*	*3363*	
IN		*C1, …, C8*	-	*YES*	*48752*	*48752*	-	-
1		*C1*	*C1*	*NO*	*51666*	*51506*	*160*	*YES*
2		***C2***	***C2***	***NO***	***52269***	***49203***	***3066***	***NO***
3		***C3***	***C3***	***NO***	***52266***	***49116***	***3150***	***NO***
4		***C4***	***C4***	***NO***	***54628***	***51302***	***3326***	***NO***
5		***C5***	***C5***	***NO***	***54574***	***51267***	***3307***	***NO***
6		***C6***	***C6***	***NO***	***54589***	***51267***	***3322***	***NO***
7		***C7***	***C7***	***NO***	***54628***	***51302***	***3326***	***NO***
8		***C8***	***C8***	***NO***	***54574***	***51267***	***3307***	***NO***
OUT	*Optimal CP:*	*C1*			*Suboptimal CPs: C2, …, C8*			

the S^4PR model. The controlled S^4PR model is live with 51418 good states.

TABLE 22. 8 CPs obtained in [53] for the S^4PR model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	3t2	3t1	5
2	6t3, 4t9, 5t11	6t2, 4t8, t10	29
3	6t4, 4t9, 5t11	6t3, 4t8, t10	29
4	2t4, 4t5, 2t8, 5t11	6t3, 2t7, 5t10	35
5	6t3, 4t5, 2t8, 5t11	6t2, 4t4, 2t7, 5t10	35
6	6t3, 4t5, 4t9, 5t11	6t2, 4t4, 2t7, 2t8, t10	36
7	2t4, 4t5, 4t9, 5t11	6t3, 2t7, 2t8, t10	36
8	4t5, 2t8	4t4, 2t7	33

Table 23 shows the results of the proposed optimality test conducted for the eight CPs depicted in Table 22. It can be seen that *C1* and *C8* are optimal CPs, while the rest of the CPs, i.e., *C2, …, C7*, are suboptimal.

The last LES is from [54], where five CPs, depicted in Table 24, are obtained to prevent deadlocks in the S^4PR model. The controlled S^4PR model is live with 51506 legal states. All CPs are optimal, since 51506 good states represent optimally permissive system behavior.

Table 25 shows the results of the proposed optimality test conducted for the eight CPs depicted in Table 24. It can be seen that all CPs are optimal.

D. G-SYSTEM

A *G-System* Petri net model from [55] is depicted in Fig. 11.

In order to simplify the G-System Petri net depicted in Fig. 11, first of all series places are merged as follows: p18 (I1) and p19 (O1) are merged as p18 (I1/O1); p11 (I3) and p20 (O3) are merged as p11 (I3/O3); p17 (I2) and p21 (O2) are merged as p18 (I2/O2). Second, as the input/output arcs of p1 and p5 are the same, p5 is removed from the model. The simplified G-System net is obtained as shown in Fig. 12. This model suffers from deadlocks. The RG of this G-System net contains 68531 states, whose LZ and DZ contain 66400 and 2131 states, respectively.

TABLE 23. Optimality test conducted for the 8 CPs depicted in Table 22.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in *RG*	*LZ*	*DZ*	Is the considered *CP* optimal?
	S^4PR model shown in Fig. 10.	-	-	*NO*	*54869*	*51506*	*3363*	
IN		*C1, C2, ..., C8*	-	*YES*	*51418*	*51418*	-	-
1		*C1*	*C1*	*NO*	*51666*	*51506*	*160*	*YES*
2		***C2***	***C2***	***NO***	***54789***	***51486***	***3303***	***NO***
3		***C3***	***C3***	***NO***	***54786***	***51466***	***3320***	***NO***
4		***C4***	***C4***	***NO***	***54852***	***51496***	***3356***	***NO***
5		***C5***	***C5***	***NO***	***54852***	***51496***	***3356***	***NO***
6		***C6***	***C6***	***NO***	***54821***	***51486***	***3335***	***NO***
7		***C7***	***C7***	***NO***	***54821***	***51485***	***3336***	***NO***
8		*C8*	*C8*	*NO*	*54864*	*51506*	*3358*	*YES*
OUT	*Optimal CPs: C1, C8*				*Suboptimal CPs: C2, ..., C7*			

TABLE 24. 8 CPs obtained in [54] for the S^4PR model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	t2	t1	1
2	2t4, t9, 2t11	2t2, t8, t10	10
3	3t4, 3t9, 4t11	3t2, 3t8, t10	21
4	t4, t5, t8, 2t11	2t2, t7, 2t10	13
5	16t5, t8, 15t9, 21t11	15t2, t4, 16t7, 6t10	159

TABLE 25. Optimality test conducted for the 5 CPs depicted in Table 24.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in *RG*	*LZ*	*DZ*	Is the considered *CP* optimal?
	S^4PR model shown in Fig. 10.	-	-	*NO*	*54869*	*51506*	*3363*	
IN		*C1, C2, ..., C5*	-	*YES*	*51506*	*51506*	-	-
1		*C1*	*C1*	*NO*	*51666*	*51506*	*160*	*YES*
2		C2	C2	*NO*	54809	*51506*	3303	*YES*
3		C3	C3	*NO*	54751	*51506*	3245	*YES*
4		C4	C4	*NO*	54859	*51506*	3353	*YES*
5		C5	C5	*NO*	54816	*51506*	3310	*YES*
OUT	*Optimal CPs: C1, C2, ..., C5*				*Suboptimal CPs: -*			

In this example the optimality tests are applied to four different sets of LESs obtained for the simplified G-System net model shown in Fig. 12.

The first LES is from [46], where eleven CPs, depicted in Table 26 (for economy of space, the tables in the subsection are presented in the Appendix), are obtained to prevent deadlocks in the simplified G-System net model. The controlled simplified G-System net model is live with 62682 good states. Since optimally controlled live G-System net model must reach 66400 good states, it can be seen that the controlled model is not optimal. Therefore, at least one of the CPs is suboptimal.

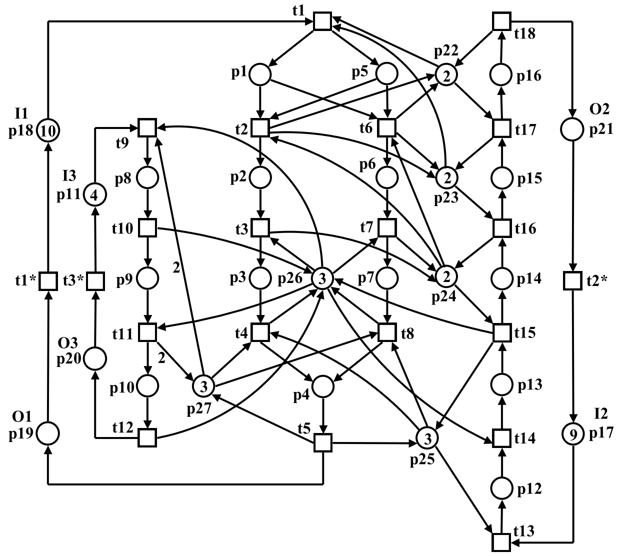

FIGURE 11. A G-system petri net from [55].

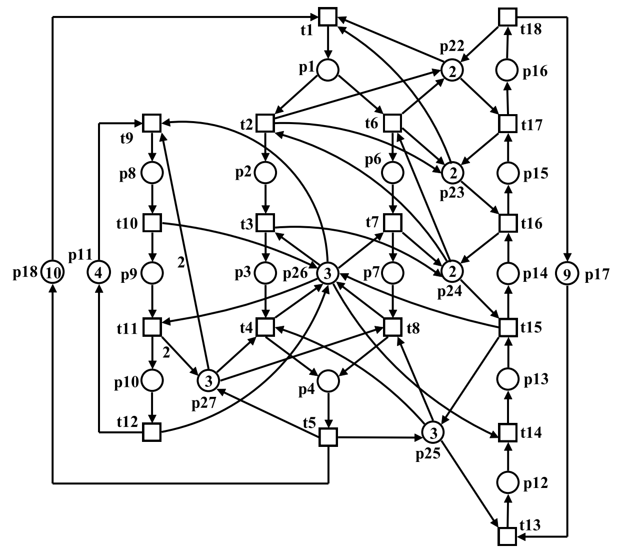

FIGURE 12. Simplified G-system model.

TABLE 26. 11 CPs obtained in [46] for the G-System model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	t2, t6, t16	t1, t15	3
2	t3, t7, t15	t2, t6, t14	4
3	t4, t7, t15	t2, t6, t13	5
4	t3, t8, t15	t2, t6, t13	5
5	t4, t8, t14	t3, t7, t13	5
6	t3, t6, t16	t1, t14	6
7	t2, t7, t16	t1, t14	6
8	t4, t6, t16	t1, t13	7
9	t2, t8, t16	t1, t13	7
10	t2, t4, t7, t16	t1, t3, t13	7
11	t3, t6, t8, t16	t1, t7, t13	7

Table 27 shows the results of the proposed optimality test conducted for the eleven CPs depicted in Table 26. It can be seen that C1, C2, C5, C6, C7 are optimal CPs, while C3, C4, C8, ..., C11 are suboptimal.

TABLE 27. Optimality test conducted for the 11 CPs depicted in Table 24.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in *RG*	*LZ*	*DZ*	Is the considered *CP* optimal?
	Simplified *G-System* model shown in Fig. 12.	-	-	*NO*	*68531*	*66400*	2131	
IN		*C1-11*	-	*YES*	*62682*	*62682*	-	-
1		***C1***	***C1***	***NO***	***68096***	***66400***	***1696***	***YES***
2		***C2***	***C2***	***NO***	***68471***	***66400***	***2071***	***YES***
3		*C3*	*C3*	*NO*	*65865*	*64628*	*1237*	*NO*
4		*C4*	*C4*	*NO*	*65865*	*64628*	*1237*	*NO*
5		***C5***	***C5***	***NO***	***67699***	***66400***	***1299***	***YES***
6		***C6***	***C6***	***NO***	***68525***	***66400***	***2125***	***YES***
7		***C7***	***C7***	***NO***	***68525***	***66400***	***2125***	***YES***
8		*C8*	*C8*	*NO*	*68107*	*66206*	*1901*	*NO*
9		*C9*	*C9*	*NO*	*68107*	*66206*	*1901*	*NO*
10		*C10*	*C10*	*NO*	*68107*	*66193*	*1914*	*NO*
11		*C11*	*C11*	*NO*	*68107*	*66193*	*1914*	*NO*
OUT	*Optimal CPs: C1, C2, C5, C6, C7*				*Suboptimal CP: C3, C4, C8, ..., C11*			

The second LES is from [53], where seventeen CPs, depicted in Table 28, are obtained to prevent deadlocks in the simplified *G-System* net mode. The controlled simplified *G-System* net model is live with 65888 good states. Since optimally controlled live *G-System* net model must reach 66400 good states, it can be seen that the controlled model is not optimal. Therefore, at least one of the CPs is suboptimal.

TABLE 28. 17 CPs obtained in [53] for the G-System model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	3t2, 3t6, 2t16	3t1, 2t15	9
2	2t3, 2t7, 3t15	2t2, 2t6, 3t14	12
3	2t4, 2t7, 3t15	2t2, 2t6, 2t13, t14	13
4	2t3, 2t8, 3t15	2t2, 2t6, 2t13, t14	13
5	2t4, 2t8, 2t14	2t3, 2t7, 2t13	11
6	2t4, 2t8, 2t14	2t2, 2t6, 2t13	13
7	2t4, 2t8, 3t15	2t2, 2t6, 2t13, t14	14
8	t2, 2t3, 3t6, t15, 2t16	3t1, 3t14	18
9	3t2, t6, 2t7, t15, 2t16	3t1, 3t14	18
10	t2, 2t3, 3t6, 2t8, t15, 2t16	3t1, 2t7, 2t13, t14	19
11	3t2, 2t4, t6, 2t7, t15, 2t16	3t1, 2t3, 2t13, t14	19
12	t2, 2t4, 3t6, t15, 2t16	3t1, 2t13, t14	19
13	3t2, t6, 2t8, t15, 2t16	3t1, 2t13, t14	19
14	t2, 2t4, 3t6, 2t8, t15, 2t16	3t1, 2t7, 2t13, t14	20
15	t2, 2t4, 3t6, 2t8, 2t14, 2t16	3t1, 2t7, 2t13, 2t15	19
16	3t2, 2t4, t6, 2t8, t15, 2t16	3t1, 2t3, 2t13, t14	20
17	3t2, 2t4, t6, 2t8, 2t14, 2t16	3t1, 2t3, 2t13, 2t15	19

Table 29 shows the results of the proposed optimality test conducted for the seventeen CPs depicted in Table 28. It can be seen that *C1, C2, C7, C8, C9, C14, C16* are optimal CPs, while *C3, C4, C6, C10, ..., C13, C15, C17* are suboptimal.

The third LES is from [56], where eleven CPs, depicted in Table 30, are obtained to prevent deadlocks in the simplified *G-System* net mode. The controlled simplified *G-System* net model is live with 66400 good states. This represents optimally controlled live system behavior. Therefore, all CPs are optimal.

TABLE 29. Optimality test conducted for the 17 CPs depicted in Table 26.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in *RG*	*LZ*	*DZ*	Is the considered *CP* optimal?
	Simplified *G-System* model shown in Fig. 12.	-	-	*NO*	*68531*	*66400*	2131	
IN		*C1-17*	-	*YES*	*65888*	*65888*	-	-
1		***C1***	***C1***	***NO***	***68096***	***66400***	***1696***	***YES***
2		***C2***	***C2***	***NO***	***68471***	***66400***	***2071***	***YES***
3		*C3*	*C3*	*NO*	*67929*	*66280*	*1649*	*NO*
4		*C4*	*C4*	*NO*	*67929*	*66280*	*1649*	*NO*
5		***C5***	***C5***	***NO***	***67699***	***66400***	***1299***	***YES***
6		*C6*	*C6*	*NO*	*66993*	*65920*	*1073*	*NO*
7		***C7***	***C7***	***NO***	***68111***	***66400***	***1711***	***YES***
8		***C8***	***C8***	***NO***	***68525***	***66400***	***2125***	***YES***
9		***C9***	***C9***	***NO***	***68525***	***66400***	***2125***	***YES***
10		*C10*	*C10*	*NO*	*68481*	*66392*	*2089*	*NO*
11		*C11*	*C11*	*NO*	*68481*	*66392*	*2089*	*NO*
12		*C12*	*C12*	*NO*	*68481*	*66392*	*2089*	*NO*
13		*C13*	*C13*	*NO*	*68481*	*66392*	*2089*	*NO*
14		***C14***	***C14***	***NO***	***68497***	***66400***	***2097***	***YES***
15		*C15*	*C15*	*NO*	*68409*	*66368*	*2041*	*NO*
16		***C16***	***C16***	***NO***	***68497***	***66400***	***2097***	***YES***
17		*C17*	*C17*	*NO*	*68409*	*66368*	*2041*	*NO*
OUT	*Optimal CPs:C1, C2, C7, C8, C9, C14, C16*				*Suboptimal CP: C3, C4, C6, C10, ..., C13, C15, C17*			

TABLE 30. 11 CPs obtained by using [56] for the G-System model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	t2, t6, t16	t1, t15	3
2	t3, t7, t15	t2, t6, t14	4
3	t4, t7, 2t15	t2, t6, t13, t14	7
4	t3, t8, 2t15	t2, t6, t13, t14	7
5	t4, t8, t14	t3, t7, t13	5
6	t4, t8, 2t14	t3, t6, 2t13	9
7	t4, t8, 2t15	t2, t6, t13, t14	7
8	t3, t6, t16	t1, t14	6
9	t2, t7, t16	t1, t14	6
10	t3, t6, t8, t15, t16	t1, t7, t13, t14	9
11	t2, t4, t7, t15, t16	t1, t3, t13, t14	9
12	t4, t6, t15, t16	t1, t13, t14	9
13	t2, t8, t15, t16	t1, t13, t14	9
14	t4, t6, t8, t15, t16	t1, t7, t13, t14	9
15	t4, t6, t8, 2t14, t16	t1, t7, 2t13, t15	11
16	t2, t4, t8, t15, t16	t1, t3, t13, t14	9
17	t2, t4, t8, 2t14, t16	t1, t3, 2t13, t15	11

Table 31 shows the results of the proposed optimality test conducted for the eleven CPs depicted in Table 30. It can be seen that all CPs, i.e., *C1, C2, ..., C17*, are optimal.

The last LES is from [54], where eight CPs, depicted in Table 32, are obtained to prevent deadlocks in the simplified *G-System* net mode. The controlled simplified *G-System* net model is live with 66400 good states. This represents optimally controlled live system behavior. Therefore, all CPs are optimal.

Table 33 shows the results of the proposed optimality test conducted for the eight CPs depicted in Table 32. It can be seen that all CPs, i.e., *C1, C2, ..., C8*, are optimal.

TABLE 31. Optimality test conducted for the 17 CPs depicted in Table 28.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in *RG*	*LZ*	*DZ*	Is the considered *CP* optimal ?
	Simplified G-System model shown in Fig. 12.	-	-	*NO*	*68531*	*66400*	2131	
IN		*C1-17*	-	*YES*	*66400*	*66400*	-	-
1		*C1*	*C1*	*NO*	*68096*	*66400*	*1696*	*YES*
2		*C2*	*C2*	*NO*	*68471*	*66400*	*2071*	*YES*
3		*C3*	*C3*	*NO*	*68291*	*66400*	*1891*	*YES*
4		*C4*	*C4*	*NO*	*68291*	*66400*	*1891*	*YES*
5		*C5*	*C5*	*NO*	*67699*	*66400*	*1299*	*YES*
6		*C6*	*C6*	*NO*	*67473*	*66400*	*1073*	*YES*
7		*C7*	*C7*	*NO*	*67931*	*66400*	*1531*	*YES*
8		*C8*	*C8*	*NO*	*68525*	*66400*	*2125*	*YES*
9		*C9*	*C9*	*NO*	*68525*	*66400*	*2125*	*YES*
10		*C10*	*C10*	*NO*	*68509*	*66400*	*2109*	*YES*
11		*C11*	*C11*	*NO*	*68509*	*66400*	*2109*	*YES*
12		*C12*	*C12*	*NO*	*68509*	*66400*	*2109*	*YES*
13		*C13*	*C13*	*NO*	*68509*	*66400*	*2109*	*YES*
14		*C14*	*C14*	*NO*	*68409*	*66400*	*2079*	*YES*
15		*C15*	*C15*	*NO*	*68377*	*66400*	*1977*	*YES*
16		*C16*	*C16*	*NO*	*68479*	*66400*	*2079*	*YES*
17		*C17*	*C17*	*NO*	*68377*	*66400*	*1977*	*YES*
OUT	*Optimal CPs: C1, C2, ..., C17*				*Suboptimal CP: -*			

TABLE 32. Optimality test conducted for the 8 CPs depicted in Table 30.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	t2, t6, t16	t1, t15	3
2	t3, t7, t15	t2, t6, t14	4
3	t4, t8, 2t15	t2, t6, t13, t14	7
4	t4, t8, t14	t3, t7, t13	5
5	t4, t8, 2t14	t2, t6, 2t13	9
6	t3, t7, t16	t1, t14	6
7	t4, t8, t15, t16	t1, t13, t14	9
8	t4, t8, 2t14, t16	t1, 2t13, t15	11

TABLE 33. Optimality test conducted for the 8 CPs depicted in Table30.

i	net	included *CP*	considered *CP*	Is the net LIVE?	The number of states in *RG*	*LZ*	*DZ*	Is the considered *CP* optimal?
	Simplified G-System model shown in Fig. 12.	-	-	*NO*	*68531*	*66400*	2131	
IN		*C1-17*	-	*YES*	*66400*	*66400*	-	-
1		*C1*	*C1*	*NO*	68096	*66400*	*1696*	*YES*
2		*C2*	*C2*	*NO*	68471	*66400*	*2071*	*YES*
3		*C3*	*C3*	*NO*	67931	*66400*	*1531*	*YES*
4		*C4*	*C4*	*NO*	67699	*66400*	*1299*	*YES*
5		*C5*	*C5*	*NO*	67473	*66400*	*1073*	*YES*
6		*C6*	*C6*	*NO*	68519	*66400*	*2119*	*YES*
7		*C7*	*C7*	*NO*	68419	*66400*	*2019*	*YES*
8		*C8*	*C8*	*NO*	68220	*66400*	*1820*	*YES*
OUT	*Optimal CPs: C1, C2, ..., C8*				*Suboptimal CP: -*			

V. DISCUSSION

In this section we will discuss and show how the proposed optimality test may be used to improve behavioral permissiveness and/or to reduce the structural complexity of the CPs. Once we have a set of suboptimal CPs obtained from the proposed optimality test, the method used in the synthesis of CPs may be improved by concentrating on the suboptimal CPs. For example, let us take a look at different LESs obtained for S^3PR model shown in Fig. 9. First, let us consider seventeen necessary CPs out of 19 depicted in Table 12. It can be seen from Table 13 that the only suboptimal CP is *C17* out of seventeen necessary CPs. For this example, we can find out the cause of the reason why the method proposed in [32] and [33] is not able to provide a maximally permissive LES, is due to the suboptimal CP C17. This means that the method proposed in [32] and [33] fails to provide an optimal CP due the computation of *C17*. As a result, the behavioral permissiveness of method proposed in [32] and [33] may be improved by concentrating on the computation of this CP. Second, let us compare seventeen necessary CPs out of 19 depicted in Table 12 with seventeen optimal CPs depicted in Table 16. It can be seen that sixteen optimal CPs of Table 12 are the same as sixteen optimal CPs of Table 16, as follows (First CP is from Table 12, the second one is from Table 16): $C1 = C1$, $C2 = C2$, $C3 = C3$, $C4 = C4$, $C5 = C7$, $C6 = C5$, $C7 = C8$, $C8 = C6$, $C9 = C9$, $C11 = C12$, $C13 = C13$, $C14 = C10$, $C15 = C14$, $C16 = C11$, $C18 = C16$, $C19 = C17$. Then it can be concluded that *C15* of Table 16 is the improved version of *C17* of Table 12. As a result, this can be considered as an example to show how to improve the behavioral permissiveness of the method proposed in [32] and [33].

As a general rule, it is always desirable to obtain ordinary CPs without weighted input/output arcs whenever possible while keeping or improving the behavioral permissiveness of the controlled net. Now let us show how to reduce the structural complexity of an LES by using the proposed optimality test. Table 28 shows 17 CPs obtained in [53] for the uncontrolled *G-System* model depicted in Fig. 12. With the addition of these CPs the controlled *G-System* model is live and can reach 65888 legal states, with 99.23% permissiveness. All these 17 CPs are generalized since their input/output arcs are weighted. On the other hand, Table 26 shows 11 CPs obtained in [46] for the uncontrolled *G-System* model. With the addition of these CPs the controlled *G-System* model is live and can reach 62682 legal states, with 94.40% permissiveness. All these 11 CPs are ordinary since the weight of all input/output arcs are 1. Now the question is "can we reduce the structural complexity of some of the CPs depicted in Table 28 by replacing them with ordinary CPs from Table 26 while keeping the same with 99.23 % permissiveness?". The answer is yes. Let us see how it is possible by using the previously obtained optimality results. The effect of including a single CP to the uncontrolled net and then analyzing the RG of this partially controlled net is provided by the number of states in RG, LZ and DZ in the optimality tests provided in the tables provided before.

In this regard, from Table 27, it is seen that when *C1* of Table 26 is included in the simplified *G-System* model shown in Fig. 12, the number of states of this partially controlled net in RG, LZ and DZ are 68096, 66400, 1696 respectively. Likewise, from Table 29, it is seen that when *C1* of Table 28 is included in the simplified *G-System* model shown in Fig. 12, the number of states of this partially controlled net in RG, LZ and DZ are 68096, 66400, 1696 respectively. This means that *C1* of Table 26 and *C1* of Table 28 have the same control action on the simplified *G-System* model shown in Fig. 12 and therefore each one of these two CPs can be used in the others place. This means that we can replace *C1* of Table 28 with *C1* of Table 26 and obtain the same permissive behavior. By further applying the same reasoning to the other CPs of Table 26 and Table 28, we can obtain total matches of CPs as follows (first from Table 26 and the second from Table 28): *C1* = *C1*, *C2* = *C2*, *C5* = *C5*, *C6* = *C8*, *C7* = *C9*. In other words, *CPs C1*, *C2*, *C5*, *C6*, *C7* from Table 26 can replace *CPs C1*, *C2*, *C5*, *C8*, *C9* of Table 28 as depicted in Table 34. It can be verified that when the CPs depicted in Table 34 are added to the uncontrolled *G-System* model depicted in Fig. 12, the resulting controlled net is live and can reach 65888 legal states, with 99.23% permissiveness. This shows that we can use the proposed optimality test to reduce the structural complexity of an LES with the same behavioral permissiveness.

TABLE 34. 17 CPs obtained for the G-System model.

C_i	$^\bullet C_i$	$C_i^\bullet$	$\mu_0(C_i)$
1	**t2, t6, t16**	**t1, t15**	**3**
2	**t3, t7, t15**	**t2, t6, t14**	**4**
3	2t4, 2t7, 3t15	2t2, 2t6, 2t13, t14	13
4	2t3, 2t8, 3t15	2t2, 2t6, 2t13, t14	13
5	**t4, t8, t14**	**t3, t7, t13**	**5**
6	2t4, 2t8, 2t14	2t2, 2t6, 2t13	13
7	2t4, 2t8, 3t15	2t2, 2t6, 2t13, t14	14
8	**t3, t6, t16**	**t1, t14**	**6**
9	**t2, t7, t16**	**t1, t14**	**6**
10	t2, 2t3, 3t6, 2t8, t15, 2t16	3t1, 2t7, 2t13, t14	19
11	3t2, 2t4, t6, 2t7, t15, 2t16	3t1, 2t3, 2t13, t14	19
12	t2, 2t4, 3t6, t15, 2t16	3t1, 2t13, t14	19
13	3t2, t6, 2t8, t15, 2t16	3t1, 2t13, t14	19
14	t2, 2t4, 3t6, 2t8, t15, 2t16	3t1, 2t7, 2t13, t14	20
15	t2, 2t4, 3t6, 2t8, 2t14, 2t16	3t1, 2t7, 2t13, 2t15	19
16	3t2, 2t4, t6, 2t8, t15, 2t16	3t1, 2t3, 2t13, t14	20
17	3t2, 2t4, t6, 2t8, 2t14, 2t16	3t1, 2t3, 2t13, 2t15	19

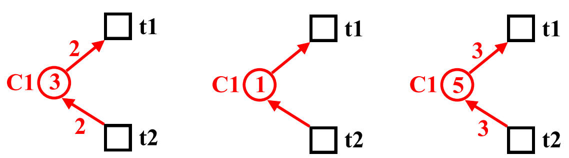

FIGURE 13. Three CPs with the same control action.

Last but not least, let us consider three CPs shown in Fig. 13. *C1* shown in Fig. 13(a) (b, c, respectively) is the CP from Table 18 (20, 22, respectively). From Table 19, it is seen that when *C1* of Table 18 is included in the S^4PR model shown in Fig. 10, the number of states of this partially controlled net in RG, LZ and DZ are 51666, 51506, 160 respectively. Likewise, from Table 21 (23, respectively), it is seen that when *C1* of Table 20 is included in the S^4PR model shown in Fig. 10, the number of states of this partially controlled net in RG, LZ and DZ are 51666, 51506, 160 respectively. As a result, we can also observe from this fact that these three CPs have the same control action on the S^4PR model shown in Fig. 10.

VI. CONCLUSION

To-date no study has been reported in the literature in order to identify both optimal and suboptimal sets of CPs of a given suboptimally controlled live Petri net model of an FMS consisting of an uncontrolled plant Petri net model and an LES that contains a set of CPs. This paper presents a straight forward and easy to apply method to identify both of these sets. The proposed approach takes an LPN model, controlled by n CPs, as input and in the case of suboptimal controlled models it produces both sets of optimal CPs and suboptimal CPs. The applicability of the proposed method is demonstrated by considering a number of examples from the literature. As shown in this paper, the set of suboptimal CPs can be used to improve behavioral permissiveness and/or to reduce the structural complexity of the CPs. Further studies are necessary to make use of the proposed optimality test in the currently available suboptimal techniques for the synthesis of deadlock prevention policies. In addition, it is interesting to consider the deadlock control problem under opacity requirements via optimal control places in manufacturing systems [57].

APPENDIX

See Tables 26–34.

ACKNOWLEDGMENT

The authors would like to thank Dr. Bernard Berthomieu, the developer of TINA, for his friendly help in the computation of LZ and DZ of RGs of petri nets by using TINA.

REFERENCES

[1] M. P. Fanti and M. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 1, pp. 5–22, Jan. 2004, doi: 10.1109/TSMCA.2003.820590.

[2] N. Viswanadham, Y. Narahari, and T. L. Johnson, "Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models," *IEEE Trans. Robot. Autom.*, vol. 6, no. 6, pp. 713–723, Dec. 1990, doi: 10.1109/70.63257.

[3] Z. A. Banaszak and B. H. Krogh, "Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows," *IEEE Trans. Robot. Autom.*, vol. 6, no. 6, pp. 724–734, Dec. 1990, doi: 10.1109/70.63273.

[4] Y. Tat Leung and G.-J. Sheen, "Resolving deadlocks in flexible manufacturing cells," *J. Manuf. Syst.*, vol. 12, no. 4, pp. 291–304, Jan. 1993, doi: 10.1016/0278-6125(93)90320-s.

[5] R. A. Wysk, N. S. Yang, and S. Joshi, "Resolution of deadlocks in flexible manufacturing systems: Avoidance and recovery approaches," *J. Manuf. Syst.*, vol. 13, no. 2, pp. 128–138, 1994.

[6] F.-S. Hsieh and S.-C. Chang, "Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 10, no. 2, pp. 196–209, Apr. 1994, doi: 10.1109/70.282544.

[7] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995, doi: 10.1109/70.370500.

[8] H. Cho, T. K. Kumaran, and R. A. Wysk, "Graph-theoretic deadlock detection and resolution for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 3, pp. 413–421, Jun. 1995, doi: 10.1109/70.388784.

[9] K. Y. Xing, B. S. Hu, and H. X. Chen, "Deadlock avoidance policy for flexible manufacturing systems," in *Petri Nets in Flexible and Agile Automation*, vol. 310. Boston, MA, USA: Springer, 1995, pp. 239–263, doi: 10.1007/978-1-4615-2231-7_9.

[10] K.-Y. Xing, B.-S. Hu, and H.-X. Chen, "Deadlock avoidance policy for Petri-net modeling of flexible manufacturing systems with shared resources," *IEEE Trans. Autom. Control*, vol. 41, no. 2, pp. 289–295, Feb. 1996, doi: 10.1109/9.481550.

[11] M. P. Fanti, B. Maione, S. Mascolo, and A. Turchiano, "Event-based feedback control for deadlock avoidance in flexible production systems," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 347–363, Jun. 1997, doi: 10.1109/70.585898.

[12] M. Lawley, S. Reveliotis, and M. Lawley, "Design guidelines for deadlock-handling strategies in flexible manufacturing systems," *Int. J. Flexible Manuf. Syst.*, vol. 9, no. 1, pp. 5–30, Jan. 1997, doi: 10.1023/a:1007937925728.

[13] D.-S. Yim, J.-I. Kim, and H.-S. Woo, "Avoidance of deadlocks in flexible manufacturing systems using a capacity-designated directed graph," *Int. J. Prod. Res.*, vol. 35, no. 9, pp. 2459–2476, Sep. 1997, doi: 10.1080/002075497194606.

[14] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpelata, "A structural approach to the problem of deadlock prevention in processes with resources," in *Proc. 4th Workshop Discrete Event Syst.*, Aug. 1998, pp. 273–278.

[15] M. A. Lawley, S. A. Reveliotis, and P. M. Ferreira, "A correct and scalable deadlock avoidance policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 14, no. 5, pp. 796–809, Oct. 1998, doi: 10.1109/70.720355.

[16] I. B. Abdallah and H. A. ElMaraghy, "Deadlock prevention and avoidance in FMS: A Petri net based approach," *Int. J. Adv. Manuf. Technol.*, vol. 14, no. 10, pp. 704–715, Oct. 1998, doi: 10.1007/bf01438223.

[17] L. Ferrarini and M. Maroni, "Deadlock avoidance control for manufacturing systems with multiple capacity resources," *Int. J. Adv. Manuf. Technol.*, vol. 14, no. 10, pp. 729–736, Oct. 1998, doi: 10.1007/bf01438225.

[18] M. A. Lawley, "Deadlock avoidance for production systems with flexible routing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 3, pp. 497–509, Jun. 1999, doi: 10.1109/70.768182.

[19] J. Park and S. P. Reveliotis, "Algebraic synthesis of efficient deadlock avoidance policies for sequential resource allocation systems," *IEEE Trans. Robot. Autom.*, vol. 16, no. 2, pp. 190–195, Apr. 2000, doi: 10.1109/70.843175.

[20] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpeleta, "An iterative method for deadlock prevention in FMS," in *Discrete Event Systems*. Boston, MA, USA: Springer, 2000, pp. 139–148, doi: 10.1007/978-1-4615-4493-7_14.

[21] Y. Huang, M. Jeng, X. Xie, and S. Chung, "Deadlock prevention policy based on Petri nets and siphons," *Int. J. Prod. Res.*, vol. 39, no. 2, pp. 283–305, Jan. 2001, doi: 10.1080/00207540010002405.

[22] J. Park and S. A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Trans. Autom. Control*, vol. 46, no. 10, pp. 1572–1583, Oct. 2001, doi: 10.1109/9.956052.

[23] J. Ezpeleta, F. Tricas, F. Garcia-Valles, and J. M. Colom, "A Banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 621–625, Aug. 2002, doi: 10.1109/TRA.2002.801048.

[24] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions," *Int. J. Adv. Manuf. Technol.*, vol. 19, no. 3, pp. 192–208, Feb. 2002, doi: 10.1007/s001700200014.

[25] M. V. Iordache, J. Moody, and P. J. Antsaklis, "Synthesis of deadlock prevention supervisors using Petri nets," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 59–68, Feb. 2002, doi: 10.1109/70.988975.

[26] M. Uzam, "The use of the Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 23, nos. 3–4, pp. 204–219, Feb. 2004, doi: 10.1007/s00170-002-1526-5.

[27] Z. Li and M. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 1, pp. 38–51, Jan. 2004, doi: 10.1109/TSMCA.2003.820576.

[28] Z. Li and M. Zhou, "An effective FMS deadlock prevention policy based on elementary siphons," in *Proc. IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, USA, Apr. 2004, pp. 3143–3148, doi: 10.1109/ROBOT.2004.1307546.

[29] M. Jeng, X. Xie, and S.-L. Chung, "ERCN merged nets for modeling degraded behavior and parallel processes in semiconductor manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 1, pp. 102–112, Jan. 2004, doi: 10.1109/TSMCA.2003.820579.

[30] Z. Li, M. Uzam, and M. Zhou, "Comments on 'Deadlock prevention policy based on Petri nets and siphons,'" *Int. J. Prod. Res.*, vol. 42, no. 24, pp. 5253–5254, Dec. 2004, doi: 10.1080/00207540412331330822.

[31] Y.-S. Huang, M. Jeng, X. Xie, and D.-H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 36, no. 6, pp. 1248–1256, Nov. 2006, doi: 10.1109/TSMCA.2006.878953.

[32] M. Uzam and M. Zhou, "An iterative synthesis approach to Petri net-based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 37, no. 3, pp. 362–371, May 2007, doi: 10.1109/TSMCA.2007.893484.

[33] M. Uzam and M. C. Zhou, "An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 44, no. 10, pp. 1987–2030, May 2006, doi: 10.1080/00207540500431321.

[34] H. Hu, Y. Liu, and M. Zhou, "Maximally permissive distributed control of large scale automated manufacturing systems modeled with Petri nets," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 5, pp. 2026–2034, Sep. 2015, doi: 10.1109/TCST.2015.2391014.

[35] S. Wang, C. Wang, and M. Zhou, "Controllability conditions of resultant siphons in a class of Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 5, pp. 1206–1215, Sep. 2012, doi: 10.1109/TSMCA.2011.2170419.

[36] G. Liu, C. Jiang, and M. Zhou, "Two simple deadlock prevention policies for S3PR based on key-resource/operation-place pairs," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 4, pp. 945–957, Oct. 2010, doi: 10.1109/TASE.2010.2050059.

[37] Y. Chen, Z. Li, and M. Zhou, "Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 3, pp. 615–629, May 2012, doi: 10.1109/TSMCA.2011.2169956.

[38] Y. Chen and Z. Li, "Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 47, no. 5, pp. 1028–1034, May 2011, doi: 10.1016/j.automatica.2011.01.070.

[39] Y.-S. Huang, T.-H. Chung, and P.-J. Su, "Synthesis of deadlock prevention policy using Petri nets reachability graph technique," *Asian J. Control*, vol. 12, no. 3, pp. 336–346, Feb. 2010, doi: 10.1002/asjc.188.

[40] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 137–141, Feb. 2003, doi: 10.1109/TRA.2002.807555.

[41] M. Uzam, Z. Li, G. Gelen, and R. S. Zakariyya, "A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems," *J. Intell. Manuf.*, vol. 27, no. 5, pp. 1111–1129, Oct. 2016, doi: 10.1007/s10845-014-0938-z.

[42] Z. Wu Li, M. Chu Zhou, and N. Qi Wu, "A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern., C*, vol. 38, no. 2, pp. 173–188, Mar. 2008, doi: 10.1109/TSMCC.2007.913920.

[43] Z. Li and M. Zhou, "Clarifications on the definitions of elementary siphons in Petri nets," *IEEE Trans. Syst. Man, Cybern. A, Syst. Humans*, vol. 36, no. 6, pp. 1227–1229, Nov. 2006, doi: 10.1109/TSMCA.2006.878966.

[44] Z. Li and M. Zhao, "On controllability of dependent siphons for deadlock prevention in generalized Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 2, pp. 369–384, Mar. 2008, doi: 10.1109/TSMCA.2007.914741.

[45] Y. Chen, Z. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 374–393, Apr. 2011, doi: 10.1109/TASE.2010.2060332.

[46] M. Uzam, Z. Li, and U. S. Abubakar, "Think globally act locally approach for the synthesis of a liveness-enforcing supervisor of FMSs based on Petri nets," *Int. J. Prod. Res.*, vol. 54, no. 15, pp. 4634–4657, Aug. 2016, doi: 10.1080/00207543.2015.1098785.

[47] C. Li, Y. F. Chen, Z. Zhong, M. Uzam, Z. W. Li, N. Wu, and M. Zhang, "A think-globally-act-locally-based method of maximally permissive liveness-enforcing supervisors for flexible manufacturing systems," *Control Eng. Appl. Informat.*, vol. 23, no. 4, pp. 46–56, 2021.
[48] INA. (Jul. 31, 2003). *Integrated Net Analyzer, a Software Tool for Analysis of Petri Nets, Version 2.2*. [Online]. Available: http://www.informatik.hu-berlin.de/~starke/ina.html
[49] G. Gelen and M. Uzam, "Computation of the number of legal states for Petri net-based deadlock prevention problems," *Sigma J. Eng. Natural Sci.*, vol. 41, no. 3, pp. 493–502, Jun. 2023, doi: 10.14744/sigma.2023.00056.
[50] TINA. *TIme Petri Net Analyzer*. Accessed: Aug. 20, 2023. [Online]. Available: https://projects.laas.fr/tina/index.php
[51] M. Uzam, Z. Li, and M. Zhou, "Identification and elimination of redundant control places in Petri net based liveness enforcing supervisors of FMS," *Int. J. Adv. Manuf. Technol.*, vol. 35, nos. 1–2, pp. 150–168, Oct. 2007, doi: 10.1007/s00170-006-0701-5.
[52] L. Piroddi, R. Cordone, and I. Fumagalli, "Selective siphon control for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 6, pp. 1337–1348, Nov. 2008, doi: 10.1109/TSMCA.2008.2003535.
[53] M. Uzam, G. Gelen, and T. L. Saleh, "Think-globally-act-locally approach with weighted arcs to the synthesis of a liveness-enforcing supervisor for generalized Petri nets modeling FMSs," *Inf. Sci.*, vol. 363, pp. 235–260, Oct. 2016, doi: 10.1016/j.ins.2015.09.010.
[54] C. Z. Li, Y. Y. Li, Y. F. Chen, N. Q. Wu, Z. W. Li, P. Ma, and H. Kaid, "Synthesis of liveness-enforcing Petri net supervisors based on a think-globally-act-locally approach and a structurally minimal method for flexible manufacturing systems," *Comput. Informat.*, vol. 41, no. 5, pp. 1310–1336, 2022, doi: 10.31577/cai_2022_5_1310.
[55] M. Zhao and Y. Hou, "An iterative method for synthesizing non-blocking supervisors for a class of generalized Petri nets using mathematical programming," *Discrete Event Dyn. Syst.*, vol. 23, no. 1, pp. 3–26, Mar. 2013, doi: 10.1007/s10626-011-0124-9.
[56] C. Li, Y. Chen, Z. Li, and K. Barkaoui, "Synthesis of liveness-enforcing Petri net supervisors based on a think-globally-act-locally approach and vector covering for flexible manufacturing systems," *IEEE Access*, vol. 5, pp. 16349–16358, 2017, doi: 10.1109/ACCESS.2017.2720630.
[57] X. Cong, M. P. Fanti, A. M. Mangini, and Z. Li, "On-line verification of current-state opacity by Petri nets and integer linear programming," *Automatica*, vol. 94, pp. 205–213, 2018.

MURAT UZAM received the B.Sc. and M.Sc. degrees from the Electrical Engineering Department, Yıldız Technical University, İstanbul, Turkey, in 1989 and 1991, respectively, and the Ph.D. degree from the University of Salford, Salford, U.K., in 1998. From 1993 to 2010, he was with the Department of Electrical and Electronics Engineering, Niğde University, Turkey, as a Research Assistant, an Assistant Professor, an Associate Professor, and a Professor. He was a Professor with the Department of Electrical and Electronics Engineering, Melikşah University, Kayseri, Turkey, from 2011 to 2016. Since April 2020, he has been a Professor with the Department of Electrical and Electronics Engineering, Yozgat Bozok University, Yozgat, Turkey. He was a Visiting Researcher with INRIA, University of Metz; University of Rennes, France, in 1999; the University of Toronto, Toronto, ON, Canada, in 2003; and Xidian University, Xi'an, China, in 2013, 2015, 2019, and 2023. He has published 47 conference papers and 104 journals and magazine papers, 73 of which are indexed by Science Citation Index Expanded (SCIE). He has published three books in Turkish and four books in English by CRC Press (Taylor & Francis Group). Currently, according to Web of Science, his H-index is 18 and his papers have been cited more than 1600 times by the papers indexed in the SCIE. He has been serving as a reviewer for prestigious journals and conferences. According to Web of Science, the number of his verified reviews is 143. His current research interests include design and implementation of discrete event control systems modelled by petri nets and, in particular, deadlock prevention/liveness enforcing in flexible manufacturing systems, programmable logic controllers (PLCs), microcontrollers (especially PIC microcontrollers), and design of microcontroller-based PLCs.

ZHIWU LI (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively. He was a Visiting Professor with the University of Toronto, Technion (Israel Institute of Technology), Martin-Luther University at Halle Saale (supported by Alexander von Humboldt Foundation), University of Cagliari, Politecnico di Bari, Conservatoire National des Arts et Métiers (Cnam, supported by the Program of Research in Paris), and Meliksah University. He has published three monographs in Springer, in 2009 and 2023, and CRC Press, in 2013. His research was cited by leading business giants, including IBM, HP, ABB, Volvo, GE, GM, Mitsubishi, Ford Car, and Huawei. His current research interests include petri net theory and applications, supervisory control of discrete event systems, data modeling, and production automation.

MOHAMMED A. EL-MELIGY received the B.Sc. degree in information technology from Menoufia University, Egypt, in 2005. He has been a Software Engineer with King Saud University, Riyadh, Saudi Arabia, since 2009. His research interests include petri nets, supervisory control of discrete event systems, database software, and network administration.

MOHAMED A. SHARAF received the Ph.D. degree in industrial engineering from Chiba University, Japan. He is currently the Head of the Development and Quality Unit, College of Engineering, King Saud University, Saudi Arabia. He has published more than 30 articles in the areas of spare parts control, quality management, maintenance, six sigma methodology, and academic accreditation.

QIWEI TANG received the B.S. and M.S. degrees from the Guangdong University of Technology, in 2002 and 2005, respectively. He is currently with Hitachi Building Technology (Guangzhou) Company Ltd. His research interests include production process automation, intelligent algorithms, and elevator control.