IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 11, 2017, accepted July 24, 2017, date of publication August 3, 2017, date of current version August 29, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2735418

A Minimal Supervisory Structure to Optimally Enforce Liveness on Petri Net Models for Flexible Manufacturing Systems

MUHAMMAD BASHIR[1], ZHIWU LI[2,1], (Fellow, IEEE), MURAT UZAM, ABDULRAHMAN AL-AHMARI[3,4], NAIQI WU[2], (Senior Member, IEEE), DING LIU[1], (Member, IEEE), AND TING QU[5]

[1]School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[2]Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China
[3]Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[4]FARCAMT Chair, Advanced Manufacturing Institute, King Saud University, Riyadh 11421, Saudi Arabia
[5]School of Electrical and Information Engineering, Jinan University (Zhuhai Campus), Zhuhai 519070, China

Corresponding authors: Zhiwu Li (zhwli@xidian.edu.cn) and Naiqi Wu (nqwu@must.edu.mo)

This work was supported in part by the National Natural Science Foundation of China under Grant 61374068 and Grant U1401240, in part by the 111 Project under Grant B14042, and in part by the Science and Technology Development Fund (FDCT), MSAR, under Grant 078/2015/A3 and Grant 106/2016/A3.

ABSTRACT This paper presents a new method of computing a minimal supervisory structure that optimally enforces liveness on the Petri net models for flexible manufacturing systems (FMSs). The proposed method utilizes the structural properties of a Petri net model to avoid the computation of its reachability graph, which in general leads to the state explosion problem. This paper aims to design a single control place for each concurrent process of a Petri net model or a sub-net model, which thus provides a constant number of control places in a supervisor regardless of the number of resource places in a Petri net or sub-net model. It is shown that the structural size of a supervisor is minimal as the number of control places depends on the number of concurrent processes in the Petri net model. Precisely, two algorithms are developed in this paper. The first aims to compute active uncontrolled transitions and the second is concerned with a method to compute the generalized mutual exclusion constraints (GMECs) for each process of the Petri net model of an FMS. Furthermore, it provides an approach to design control places for each computed GMEC without solving integer linear programming problems, which greatly reduces the computational costs. When the computed control places are coupled with the uncontrolled Petri net model for an FMS, it optimally enforces liveness behavior of the Petri net model, and hence ensures the high utilization of resources in a considered system.

INDEX TERMS Liveness-enforcing supervisor, deadlock, flexible manufacturing system (FMS), Petri net.

I. INTRODUCTION

Flexible manufacturing systems (FMSs) that are usually highly automated production processes with complex structures ever-increasingly play an essential role in economic diversifications. Recently, most of developed countries have to pay special attention to novel production modes in which FMSs act as a critical part in boosting desired economic aspiration [72], [73]. However, FMSs require highly technological skills to manage and utilize all the resources incorporated. An FMS usually consists of two main parts: a physical system that is composed of manufacturing resources (such as machine tools, robots and transportation systems) shared by multiple jobs and a management system or decision making system responsible for the control of the physical system to achieve the goal of productivity with pre-established quality [22]. Due to the request of high throughput of such a system, deadlocks can occur because of high resource-sharing, which may degrade the performance of an FMS [13], [15], [23], [41], [63]. In general, deadlocks can occur in highly automated production processes such as semiconductor manufacturing or safety-critical systems and thus may lead to serious economy losses [74], [75].

Deadlocks can be tackled in the design and control stage of an FMS [10], [25], [40], [64]. In a resource allocation system, Coffman in [18] enumerates necessary conditions for the occurrence of a deadlock. They are popularly known as

Coffman conditions [14], [29], [64], i.e., (1) Mutual exclusion: a process can only utilize one resource at a time; (2) Hold and wait: processes that use some resources may need another new resource; (3) Non-preemption: it is infeasible to remove a resource that is held by a particular process, but a process can only release a resource by an explicit action of that process; and (4) Circular-wait: two or more processes form a circular chain where each process waits for a resource that is held by the next process in the chain. To prevent the occurrence of deadlocks, at least one of the four conditions should be broken. Several tools have been developed to deal with deadlocks in FMSs [1], [4], [14], [34], [42], [43], [63].

An effective and accurate supervisor can prevent the occurrence of deadlocks in an FMS [19], [44], [46], [47]. Petri nets, automata and graph theory are the three main methodologies to deal with this problem. Petri nets remain the most effective formalism to model and control an FMS as well as discrete event systems [38], [56]–[62]. Furthermore, they can verify the existence of deadlocks and prevent their occurrences in an FMS, since they are appropriate to describe both structural and behavioral properties of an FMS, such as conflicts, concurrency, casual dependency, liveness, and boundedness [10], [33], [39], [45], [48], [50], [52], [54], [70], [71], [76], [77]. There are four typical strategies for handling deadlocks in automated manufacturing systems [2], [10], [14], [16], [51], [55], [64]: (1) Deadlock ignoring, (2) Deadlock detection and recovery, (3) Deadlock avoidance, and (4) Deadlock prevention.

Two techniques exist for the analysis of a Petri net model to cope with deadlocks [1], [4], [5], [8], [11], [14], [44], [54], [78]: structural analysis, and reachability graph analysis. The reachability graph (RG) analysis usually requires a complete or partial enumeration of reachable states. Therefore, it suffers from the state explosion problem [53]. The theory of regions developed in [21] remains the most effective method of deadlock prevention for designing an optimal supervisor, ensuring that all the legal (safe), i.e., live states are preserved, although it is computationally expensive due to too many inequality constraints in the linear programming problems [21]. A supervisor is said to be *optimal* or *maximally permissive* if the supervisor of a plant (a plant is an alias of a system to be controlled) can prevent the reachability of all unsafe (illegal) states (a state is said to be *unsafe* if the initial marking is not reachable from it) and ensure the reachability of all safe states (a state is said to be *safe* if the initial marking is reachable from it). A system is said to be *optimally controlled* if it is supervised by an optimal supervisor. When such a system is modeled with Petri nets, it is called a optimally controlled (Petri net) system. That is to say, an optimally controlled (Petri net) system contains all the safe states and no unsafe states.

To upgrade and improve the ideas of theory of regions, the work in [35] and [36] divides the reachability graph (RG) of a Petri net model into two disjoint components: a live zone (LZ) and a deadlock zone (DZ). The former contains legal (safe) markings and the latter contains illegal (unsafe) markings. The partition of an RG is used to find the first-met bad markings (FBMs) in the DZ such that, once all FBMs are prohibited, all illegal (unsafe) markings in the DZ are not reachable. The deadlock prevention method in [35] and [36] is an iterative procedure in which at each iteration step, an FBM is controlled by designing a control place. The iterations are repeated until all FBMs in the DZ of a Petri net model are forbidden. However, this method does not guarantee maximally permissive behavior of the controlled system. It provides a sub-optimal (near-optimal) behavior. A supervisor is said to be *near-optimal* or *sub-optimal* if it is not optimal but very close to be "optimal". Usually, the behavior of a near-optimal supervisor should make the controlled system reach more than 90% legal (safe) states, while all the illegal states are prohibited. For example, suppose that a plant has h safe states. A near-optimal supervisor should make the plant contain more than $0.9h$ of safe states (of course no unsafe states are included).

For structural analysis, it utilizes some of the important properties of a Petri net structure, such as siphons, place and transition invariants, and resource transition circuits. Such properties are useful, as they can describe the relationships between behavioral properties of a Petri net and its structural components. A deadlock prevention policy derived from siphon control is usually sub-optimal, since it is difficult to ensure that each legal state is included in the controlled system. The main idea behind siphon control is to ensure that each strict minimal siphon is sufficiently marked, by adding control places to the Petri net model of an FMS [20], [26], [27], [30], [32], [34], [35], [43], [64], [66]. In general, the number of siphons grows exponentially with the structural size of a Petri net. The studies in [15], [23], and [24] develop an elementary siphon approach to reduce the computational complexity for designing a supervisor and the structural complexity of the supervisor for an FMS. However, it does not in general provide an optimal control strategy. The work in [23] adopts the elementary siphon-based approaches to reduce the number of siphons to be explicitly controlled. However, the concept does not in general provide a maximally permissive supervisor except for a limited class of Petri nets at some particular initial markings [67].

Liu *et al.* [28] develop a deadlock prevention policy using structural analysis by combining elementary siphons with varying arc weights for a Petri net model of an FMS. The policy employs the circular wait (CW) structure and the state of circular blockings (CBs) to investigate the existence of deadlocks in a Petri net model. Deadlocks are tackled by adjusting the weighted simple directed circuits (WSDCs) that describe the structure of CW with weights, which are tied to the occurrence of deadlocks. The advantage of the policy is that the information of the weighed arcs is included in the computation of siphons.

The generalized mutual exclusion constraints (GMECs) are developed in [65]. They can represent effectively the first-met bad markings (FBMs) from the deadlock zone of a Petri net model. A GMEC can express that the weighted token sum in a particular set of places is not greater than a given constant. The study in [3] develops a method to enforce a conjunction of GMECs on a controlled Petri net. The method addresses the structural properties of forward-concurrent-free nets. The developed method transforms a given conjunction of GMECs into a conjunction of admissible GMECs. However, the method does not specify the class of Petri nets considered in the work.

Furthermore, the work in [9] develops an approach to construct a controller that realizes an OR-AND GMEC. The developed method introduces a monitor switcher control structure which contains both places and transitions to enforce OR-AND GMECs. It proposes a two-stage procedure for designing a controller capable of enforcing OR-AND GMECs under the assumption that the GMECs are bounded. By using this approach, a compiled Petri net controller to enforce a given OR-AND GMEC can be obtained. The main advantage of such a compiled controller is the possibility of constructing a model of a closed-loop system as a place/transition net that can be validated using existing techniques such as structural analysis. The shortcoming of the developed method is that the OR-AND GMECs-based controller may create extra states that are not in the state space of the original uncontrolled Petri net model. Furthermore, the method, due to its generality, does not utilize the structural properties of the Petri net model for flexible manufacturing systems. It is worthy noting that the results from general Petri nets provide a spur and motivation to the development of deadlock control strategies for FMSs [69], [71], [72], [75].

In this paper, a method is proposed to enforce liveness for a safe Petri net model, called an α_n-S^{3}PR, for FMSs. It provides a novel, yet minimal supervisory structure for the Petri net model. A modified GMEC is used to design control places to enforce liveness on the Petri net model for such FMSs. The proposed method considers three cases of a safe Petri net model with an α_n-S^{3}PR structure. For each case, a minimal supervisory structure is derived to enforce liveness on the Petri net model, achieving optimal or near-optimal control purposes. The main contributions of the proposed method are summarized as follows:

(1) It uses structural analysis to compute active uncontrolled transitions for each concurrent process of a Petri net model to avoid the computation of all the reachable markings.

(2) For each active uncontrolled transition, a modified GMEC equation is computed and a control place is designed to enforce liveness of a plant net with its sub-optimal control.

(3) Experimental examples show that the proposed method can derive a structurally simple supervisor.

The proposed approach is different from the traditional methods (policies) available in the literature. In this method, we associate each concurrent process of the Petri net model or sub-net model to be controlled with a single control place, which precisely reduces the structural complexity of a supervisor compared with traditional methods. On the other hand, the complexity of the supervisory structure due to the traditional policies depends on the number of resources in the Petri net model, i.e., the larger the number of resources is, the more complex of the supervisory structure is. While, for the proposed method, its complexity depends solely on the number of concurrent processes of an FMS. In general, the number of concurrent processes is always less than that of available resources in a Petri net or sub-net model. Hence, the supervisory structure derived from the proposed method is always structurally simple regardless of the number of resources used in an FMS.

The remainder of this paper is organized as follows. Formal descriptions of Petri nets and notations used in this paper are presented in Section II. Section III describes Petri net models with an α_n-S^{3}PR or α_n-S^{4}PR structure. Section IV presents a method of deriving GMECs from α_n-S^{3}PR and α_n-S^{4}PR. Section V develops the method to compute control coefficients without solving integer linear programming, while Section VI reports a deadlock prevention policy. Experimental examples are provided in Section VII. A discussion for the computational complexity is provided in Section VIII. Finally, Section IX concludes this paper.

II. PRELIMINARIES

A Petri net is a four-tuple $N = (P, T, F, W)$, where P and T are finite and non-empty sets. P is a set of places and T is a set of transitions with $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. Places are graphically represented by circles while transitions by bars or square boxes. $W\colon (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ if $(x, y) \in F$, and $W(x, y) = 0$, otherwise, where $(x, y) \in (P \times T) \cup (T \times P)$ and $\mathbb{N}$ is the set of non-negative integers. $N = (P, T, F, W)$ is said to be *ordinary*, denoted as $N = (P, T, F)$, if $\forall f \in F$, $W(f) = 1$. Let $x \in P \cup T$ be a node in $N = (P, T, F, W)$. The preset of x, denoted by $^\bullet x$, is defined as $^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ and $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$ is called the postset of x. A marking M of a Petri net $N = (P, T, F, W)$ is a mapping from $M\colon P \rightarrow \mathbb{N}$, where M is a $|P|$-dimensional vector. Let $t \in T$ be a transition in $N = (P, T, F, W)$. Transition t is said to be enabled at a marking M, denoted by $M[t\rangle$, if $\forall p \in {^\bullet t}$, $M(p) \geq W(p, t)$. An enabled transition t can fire, leading to a new marking M', i.e., $\forall p \in P$, $M'(p) = M(p) - W(p, t) + W(t, p)$. A place $p \in P$ is said to be bounded if $\forall M \in R(N, M_0)$, $\exists k \in \mathbb{N}$, $M(p) \leq k$. A net system is said to be k-bounded if any place is k-bounded. A place $p \in P$ is said to be safe if it is 1-bounded. A net is said to be safe if all of its places are safe. Transition $t \in T$ in (N, M_0) is said to be live if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, such that $M'[t\rangle$. (N, M_0) is live if $\forall t \in T$, t is live at M_0. N is dead at M_0 if $\nexists t \in T$, $M_0[t\rangle$ holds. Let σ be a transition sequence. The Parikh vector of σ, denoted by $\overrightarrow{\sigma}$, is a column vector, represented by $\overrightarrow{\sigma}=[\#\sigma(t_1), \#\sigma(t_2), \ldots, \#\sigma(t_{|T|})]^T$, where $\#\sigma(t_i)$ denotes the number of occurrences of t_i in σ. A P-vector is a column vector $I\colon P \rightarrow \mathbb{Z}$, indexed by P, where $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$. A P-vector I is a place invariant if $I \neq 0$ and $I^T[N] = 0^T$. A T-vector is a column vector $H: T \rightarrow \mathbb{Z}$ indexed by T. A T-vector H is a transition invariant if $[N]H = \mathbf{0}$ and $H \neq \mathbf{0}$. The support of a place (transition) invariant $I(H)$ is denoted by $||I|| = \{p|I(p) \neq 0\}$ ($||H|| = \{t|H(t) \neq 0\}$). Let $\mathbb{N}^+$ denote the set of positive integers.

III. PETRI NET MODEL WITH α_n-S^{3}PR AND α_n-S^{4}PR STRUCTURES

Systems of simple sequential processes with resources (S^{3}PR) are extensively used and adopted in the analysis of FMSs [19], [25]. In this paper, two subclasses of an S^{3}PR and one subclass of S^{4}PR are considered.

Definition 1: A system of simple sequential processes with resources (S^{3}PR) is a Petri net $N = (P, T, F)$, satisfying:

1) $P = P_A \cup P_R \cup P^0$ is a partition of places with $P_A \cap P_R = \emptyset$, $P_A \cap P^0 = \emptyset$, and $P_R \cap P^0 = \emptyset$;
   a) $P_A = \bigcup_{i=1}^{m} P_{Ai}$ with $P_{Ai} \cap P_{Aj} = \emptyset$ for all $i \neq j$ is called a set of activity places;
   b) $P_R = \{r_1, r_2, \cdots, r_n\}$, $n > 0$, is called a set of resource places;
   c) $P^0 = \{p_1^0, p_2^0, \cdots, p_m^0\}$, $m > 0$, is called a set of idle places;
2) $T = \bigcup_{i=1}^{m} T_i$, with $T_i \cap T_j = \emptyset$ for all $i \neq j$, is a set of transitions;
3) $\forall i \in \{1, 2, \cdots, m\}, \forall p \in P_{Ai}$, ${}^{\bullet\bullet}p \cap P_R = p^{\bullet\bullet} \cap P_R$ and $|{}^{\bullet\bullet}p \cap P_R| = 1$,
4) N is strongly connected.

Definition 2: An S^{3}PR Petri net model $N = (P, T, F)$ satisfying the following statements is called an α_1-S^{3}PR:

1) $\forall i \in \{1, 2, \cdots, m\}, \forall r_i \in P_R$, $r_2 \in r_1^{\bullet\bullet}$, $r_3 \in r_2^{\bullet\bullet}$, $\cdots$, $r_m \in r_{(m-1)}^{\bullet\bullet}$;
2) $\forall r \in P_R$, $|{}^{\bullet}r \cap T_i| = |r^{\bullet} \cap T_i| = 1$ and $|{}^{\bullet}r \cap T_j| = |r^{\bullet} \cap T_j| = 1$, with $T_i \cap T_j = \emptyset$, for all $i \neq j$;
3) $\forall i \in \{1, 2, \cdots, m\}$, $r_i^{\bullet\bullet} = {}^{\bullet\bullet}r_i$.

Definition 3 [68]: An S^{4}PR is a connected generalized self-loop free Petri net $N = (P, F, T, W)$ where:

1) $P = P^0 \cup P_A \cup P_R$ is a partition such that a) $P_A = \bigcup_{i \in \mathbb{N}^+} P_{A_i}$ is a set of activity places, where for each $i \in \mathbb{N}^+$, $P_{A_i} \neq \emptyset$ and for each $i, j \in \mathbb{N}^+$, $i \neq j$, $P_{A_i} \cap P_{A_j} = \emptyset$; b) $P_0 = \bigcup_{i \in \mathbb{N}^+} \{p_i^0\}$ is a set of idle places; c) $P_R = \bigcup_{i \in \mathbb{N}^+} P_{R_i} = \{r_1, r_2, \cdots, r_n\}$, $n > 0$, is a set of resource places.
2) $T = \bigcup_{i \in \mathbb{N}^+} T_i$, where for each $i \in \mathbb{N}^+$, $T_i \neq \emptyset$, and for each $i, j \in \mathbb{N}^+$, $T_i \cap T_j = \emptyset$;
3) For each $i, j \in \mathbb{N}^+$, the subnet $N_i^S = N|N(\{P_i^0\} \cup P_{A_i}, T_i)$ is a strongly connected state machine such that every cycle contains p_i^0.
4) $\forall r \in P_{R_i}$, there exists a unique minimal P-semiflow $I_r \in \mathbb{N}^{|P|}$ such that $\{r\} = ||I_r|| \cap P_{R_i}$, $\{p_i^0\} \cap ||I_r|| = \emptyset$, $||I_r|| \neq \emptyset$, $I_r(r) = 1$.
5) $P_{A_i} = \bigcup_{r \in P_{R_i}} (||I_r|| \setminus \{r\})$.

Definition 4: A Petri net model (N, M_0) with $N = (P, T, F)$ being an S^{4}PR is called an α_2-S^{4}PR if it satisfies the following statements:

1) $\forall i \in \{1, 2, \cdots, m\}, \forall r_i \in P_R$, $r_2 \in r_1^{\bullet\bullet}$, $r_3 \in r_2^{\bullet\bullet}$, $\cdots$, $r_m \in r_{(m-1)}^{\bullet\bullet}$;
2) $\forall r \in P_R$, $|{}^{\bullet}r \cap T_i| = |r^{\bullet} \cap T_i| = 1$ and $|{}^{\bullet}r \cap T_j| = |r^{\bullet} \cap T_j| = 1$ with $T_i \cap T_j = \emptyset$, for all $i \neq j$;
3) $\forall i \in \{1, 2, \cdots, m\}$, $r_i^{\bullet\bullet} \cap P_R = {}^{\bullet\bullet}r_2 \cap P_R$;
4) $\forall r \in P_R, \exists t \in T$, $|{}^{\bullet}t| > 1$, ${}^{\bullet}t \cap P_R = t^{\bullet\bullet\bullet} \cap P_R = \{r\}$ and $|t^{\bullet\bullet\bullet} \cap P_R| = 1$.

Definition 5: An S^{3}PR Petri net model (N, M_0) with $N = (P, T, F)$ is called α_3-S^{3}PR if it satisfies the following statements:

1) $\forall i \in \{1, 2, \cdots, m\}, \forall r_i \in P_R$, $r_2 \in r_1^{\bullet\bullet}$, $r_3 \in r_2^{\bullet\bullet}$, $\cdots$, $r_m \in r_{(m-1)}^{\bullet\bullet}$;
2) $\forall r \in P_R$, $|{}^{\bullet}r \cap T_i| = |r^{\bullet} \cap T_i| = 1$ and $|{}^{\bullet}r \cap T_j| = |r^{\bullet} \cap T_j| = 1$, with $T_i \cap T_j = \emptyset$, for all $i \neq j$;
3) $\forall i \in \{1, 2, \cdots, m\}$, $r_i^{\bullet\bullet} \cap P_A = {}^{\bullet\bullet}r_i \cap P_A$;
4) $\forall i \in \mathbb{N}_m, \forall p \in P_{Ai}$, ${}^{\bullet\bullet}p \cap P_R \neq p^{\bullet\bullet} \cap P_R$, ${}^{\bullet}(p^{\bullet}) \cap P_R \neq ({}^{\bullet}p)^{\bullet} \cap P_R$, $\exists r \in P_R$, $({}^{\bullet}(p^{\bullet}) \cap P_R) \cup (({}^{\bullet}p)^{\bullet} \cap P_R) = \{r\}$, and $|{}^{\bullet}(p^{\bullet}) \cap P_R| = 1$.

In this paper, a Petri net N under consideration can be thought of as the composition of a set of subnets N_1, N_2, ..., N_m via shared places, which is denoted as $N = \bigcirc_{i=1}^{m} N_i$.

Definition 6 [67]: A directed circuit τ in a marked Petri net (N, M_0) is called a resource transition circuit (RTC) if it contains resource places and transition only. Let $\tau(t)$ and $\tau(r)$ denote the sets of all transitions and resource places in τ, respectively.

Definition 7: Let χ be an elementary circuit in a Petri net model (N, M_0). χ is called a minimal activity resource transition circuit (ARTC) if it contains activity places, resource places and transitions. Let $\chi(A)$, $\chi(r)$, and $\chi(t)$ denote the sets of activity places, resources, and transitions in χ, respectively.

Definition 8: Resource r is said to be shared if $\exists p, p' \in (||I_r|| \setminus \{r\})$, $p \in P_{Ai}$, $p' \in P_{Aj}$, $i \neq j$.

Precisely, an RTC τ does not contain any activity places and can be determined by its transition set $\tau(t)$ and resource set $\tau(r)$ such that $\tau = N[\tau(t) \cup \tau(r)]$. While an ARTC χ contains activity places, resource places and transitions, and can be determined by its activity place set $\chi(A)$, resource place set $\chi(r)$ and transition set $\chi(t)$ such that $\chi = N[\chi(A) \cup \chi(r) \cup \chi(t)]$.

The following properties are verified for a Petri net model with an α_1-S^{3}PR structure:

- For any RTC τ, $(\tau(t))^{\bullet} \cap \tau(r) = \tau(r)$;
- For any ARTC χ, $||\chi(A)|| = 1$ and $||\chi(t)|| = 2$;
- Let r_i be a resource place and τ_1 and τ_2 be two RTCs such that $r_i \in \tau_1$ and $r_i \in \tau_2$. The following properties hold:
  (a) ${}^{\bullet\bullet}(r_i) \cap \tau_1 = (r_i)^{\bullet\bullet} \cap \tau_1$ and
  (b) ${}^{\bullet\bullet}(r_i) \cap \tau_2 = (r_i)^{\bullet\bullet} \cap \tau_2$.

A Petri net model with an α_2-S^{4}PR structure has a dual mode of (i.e., two separate) distinct RTCs; each separate RTC has the following properties:

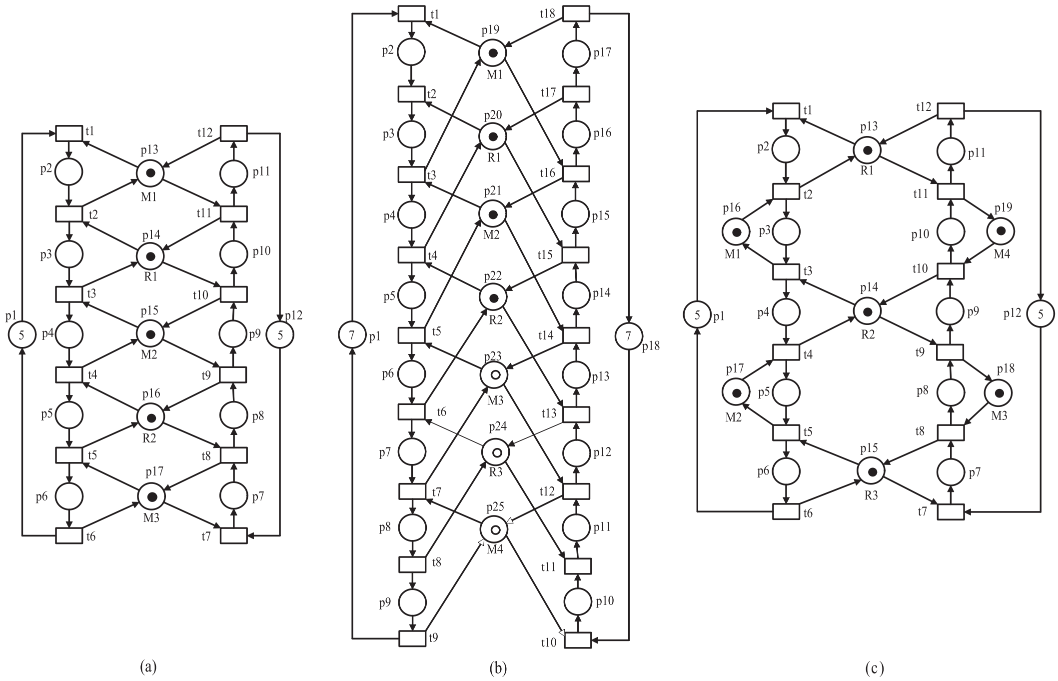

FIGURE 1. Petri net model with an (a) α_1-S^{3}PR structure (b) α_2-S^{4}PR structure, and (c) α_3-S^{3}PR structure.

- $(\tau(t))^{\bullet} \cap \tau(r) = \tau(r)$;
- $||\chi(A)|| = 2$, $||\chi(t)|| = 3$; and
- Let r_i be a resource place and τ_1 and τ_2 be two RTCs such that $r_i \in \tau_1$ and $r_i \in \tau_2$. The following properties hold:
  (a) $^{\bullet\bullet}(r_i) \cap \tau_1 = (r_i)^{\bullet\bullet} \cap \tau_1$ and
  (b) $^{\bullet\bullet}(r_i) \cap \tau_2 = (r_i)^{\bullet\bullet} \cap \tau_2$.

Fig. 1(a) shows a typical example of a Petri net with an α_1-S^{3}PR structure. Fig. 1(b) shows an example of a Petri net model with an α_2-S^{4}PR structure, where its RTCs are:

$$\tau_1 = \langle p_{19}, t_{16}, p_{21}, t_3, p_{19} \rangle,$$
$$\tau_2 = \langle p_{20}, t_{15}, p_{22}, t_4, p_{20} \rangle,$$
$$\tau_3 = \langle p_{21}, t_{14}, p_{23}, t_5, p_{21} \rangle,$$
$$\tau_4 = \langle p_{22}, t_{13}, p_{24}, t_6, p_{22} \rangle$$

and

$$\tau_5 = \langle p_{23}, t_{12}, p_{25}, t_7, p_{23} \rangle.$$

The ARTCs are

$$\chi_1 = \langle p_{19}, t_1, p_2, t_2, p_3, t_3, p_{19} \rangle,$$
$$\chi_2 = \langle p_{20}, t_2, p_3, t_3, p_4, t_4, p_{20} \rangle,$$
$$\chi_3 = \langle p_{21}, t_3, p_4, t_4, p_5, t_5, p_{21} \rangle,$$
$$\chi_4 = \langle p_{22}, t_4, p_5, t_5, p_6, t_6, p_{22} \rangle,$$
$$\chi_5 = \langle p_{23}, t_5, p_6, t_6, p_7, t_7, p_{23} \rangle,$$
$$\chi_6 = \langle p_{24}, t_6, p_7, t_7, p_8, t_8, p_{24} \rangle,$$
$$\chi_7 = \langle p_{25}, t_7, p_8, t_8, p_9, t_9, p_{25} \rangle,$$
$$\chi_8 = \langle p_{19}, t_{16}, p_{16}, t_{17}, p_{17}, t_{18}, p_{19} \rangle,$$
$$\chi_9 = \langle p_{20}, t_{15}, p_{15}, t_{16}, p_{16}, t_{17}, p_{20} \rangle,$$
$$\chi_{10} = \langle p_{21}, t_{14}, p_{14}, t_{15}, p_{15}, t_{16}, p_{21} \rangle,$$
$$\chi_{11} = \langle p_{22}, t_{13}, p_{13}, t_{13}, p_{14}, t_{15}, p_{22} \rangle,$$
$$\chi_{12} = \langle p_{23}, t_{12}, p_{12}, t_{13}, p_{14}, t_{14}, p_{23} \rangle,$$
$$\chi_{13} = \langle p_{24}, t_{11}, p_{11}, t_{12}, p_{12}, t_{13}, p_{24} \rangle$$

and

$$\chi_{14} = \langle p_{25}, t_{10}, p_{10}, t_{11}, p_{11}, t_{12}, p_{25} \rangle.$$

Let us consider the Petri net model with an α_3-S^{3}PR structure. The resource places in the net are partitioned into two sets: shared and unshared resource places, i.e., $P_R = R^S \cup R^U$, where R^S and R^U denote the sets of shared and unshared resource places, respectively. Fig. 1(c) shows a typical example of an α_3-S^{3}PR structure. From Fig. 1(c), the sets of shared and unshared resource places are $R^S = \{p_{13}, p_{14}, p_{15}\}$ and $R^U = \{p_{16}, p_{17}, p_{18}, p_{19}\}$, respectively. The following properties are verified for the Petri net model with an α_3-S^{3}PR structure:

- For any RTC τ, $^{\bullet\bullet}\tau(t) \cap \tau(t) = \tau(t)^{\bullet\bullet} \cap \tau(t)$, $^{\bullet\bullet}\tau(t) = \tau(t)^{\bullet\bullet}$, and $^{\bullet}\tau(t) \cap R(\tau) = \tau(t)^{\bullet} \cap \tau(r)$, and
- $||\chi(A)|| = 1$ and $||\chi(t)|| = 2$.

The ARTCs in the Petri net model shown in Fig. 1(c) are $\chi_1 = \langle p_{13}, t_1, p_2, t_2, p_{13} \rangle$, $\chi_2 = \langle p_{16}, t_2, p_3, t_3, p_{16} \rangle$, $\chi_3 = \langle p_{14}, t_3, p_4, t_4, p_{14} \rangle$, $\chi_4 = \langle p_{17}, t_4, p_5, t_5, p_{17} \rangle$, $\chi_5 = \langle p_{15}, t_5, p_6, t_6, p_{15} \rangle$, $\chi_6 = \langle p_{15}, t_7, p_7, t_8, p_{15} \rangle$, $\chi_7 = \langle p_{18}, t_8, p_8, t_9, p_{18} \rangle$, $\chi_8 = \langle p_{14}, t_9, p_9, t_{10}, p_{14} \rangle$, $\chi_9 = \langle p_{19}, t_{10}, p_{10}, t_{11}, p_{19} \rangle$, and $\chi_{10} = \langle p_{13}, t_{11}, p_{11}, t_{12}, p_{13} \rangle$.

Definition 9: Let (N_i^S, M_0^S) be the sub-net in the Petri net model with an α_2-S^{4}PR structure. The sub-net (N_i^S, M_0^S) has the following properties:

1) $N_i^S = (P_{A_i} \cup P^0 \cup P_{R_i}, T, F_i)$.
2) $\forall N_i^S$, $P_{R_i} \subseteq P_R$, $P_{A_i} \subseteq P_A$ and $F_i \subseteq F$.
3) $\forall N_i^S$, we have (a) ${}^{\bullet\bullet}(r_i) \cap \tau_1 = (r_i)^{\bullet\bullet} \cap \tau_1$ and (b) ${}^{\bullet\bullet}(r_i) \cap \tau_2 = (r_i)^{\bullet\bullet} \cap \tau_2$.
4) $\forall r \in P_{R_i}$, there exists a unique minimal P-semiflow $I_r \in \mathbb{N}^{||m||}$ such that $\{r\} = ||I_r|| \cap P_{R_i}$, $P_i^0 \cap ||I_r|| \neq \emptyset$, $||I_r|| \neq \emptyset$, $I_r(r) = 1$, where we assume $m = |P|$.
5) $P_{A_i} = \bigcup_{r \in P_{R_i}} (||I_r|| \setminus \{r\})$.

To ease the computation load of synthesizing a supervisor of a Petri net model with α_2-S^{4}PR structure for an FMS, the Petri net model can be split-up into two sub-nets with each having two concurrent processes and satisfying Definition 9. The Petri net model shown in Fig. 1(b) is split-up into two different sub-nets (N_1^S, M_0^S) and (N_2^S, M_0^S) as shown in Fig. 2(a) and (b), respectively.

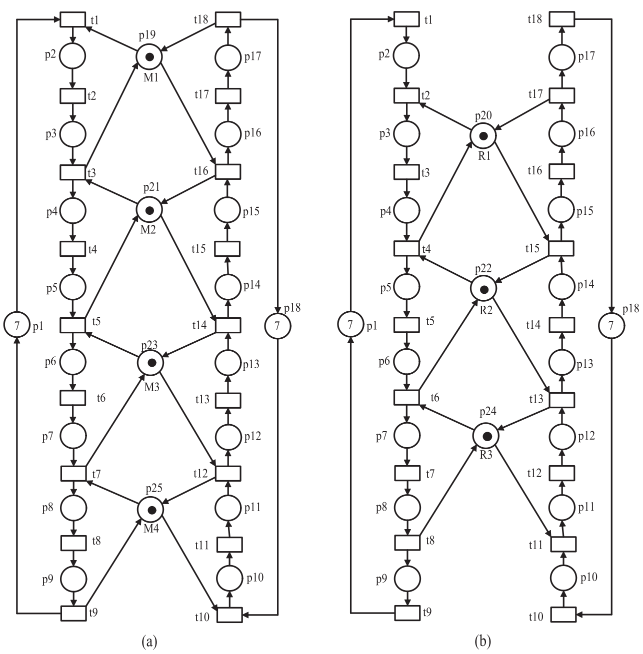

FIGURE 2. Sub-net model of (a) (N_1^S, M_0^S) and (b) (N_2^S, M_0^S).

An ARTC in a Petri net model (N, M_0) is either formed by a shared resource place or unshared resource place. For a better representation, with no ambiguity, let $\chi^j_{r_i^S}(A)$ be the set of activity places in an ARTC χ formed by a shared resource place r_i in process j and $\chi^j_{r_k^U}(A)$ be the set of activity places in an ARTC formed by an unshared resource place r_k in process j, where $i \in \{1, 2, \cdots, \alpha\}$ and $k \in \{\alpha + 1, \alpha + 2, \cdots, \beta\}$, indicating that there are α shared resource places and $\beta - \alpha$ unshared resource places, respectively, in a Petri net model. To further simplify the notation, let χ^j_{ci} denote the algebraic sum form of $\chi^j_{r_i^S}(A)$. Note that $j \in \{1, 2, \cdots, l\}$ in $\chi^j_{r_i^S}(A)$ $(\chi^j_{r_i^U}(A))$ represents the j-th concurrent process in the Petri net model.

Definition 10: If there exists an ARTC χ_r formed by an unshared resource place r involving the j-th process between the two ARTCs formed by shared resource places, i.e., $\chi^j_{r_i^S}(A) \to \chi^j_r(A) \to \chi^j_{r_{i+1}^S}(A)$, then χ_{r_i} is called an ARTC attached by χ_r. The extended algebraic sum form of $\chi^j_{r_i^S}(A)$ is defined as $\chi^j_{ci} = \chi^j_{r_i^S}(A) + \chi^j_r(A)$.

Let us consider the Petri net model shown in Fig. 1(a). The sets of activity places in ARTCs are $\chi^1_{r_1^S}(A) = \{p_2\}$, $\chi^1_{r_2^S}(A) = \{p_3\}$, $\chi^1_{r_3^S}(A) = \{p_4\}$, $\chi^1_{r_4^S}(A) = \{p_5\}$, $\chi^1_{r_5^S}(A) = \{p_6\}$, $\chi^2_{r_5^S}(A) = \{p_7\}$, $\chi^2_{r_4^S}(A) = \{p_8\}$, $\chi^2_{r_3^S}(A) = \{p_9\}$, $\chi^2_{r_2^S}(A) = \{p_{10}\}$, and $\chi^2_{r_1^S}(A) = \{p_{11}\}$. Their corresponding algebraic sum form of $\chi^j_{r_i^S}(A)$ is $\chi^1_{c1} = p_2$, $\chi^1_{c2} = p_3$, $\chi^1_{c3} = p_4$, $\chi^1_{c4} = p_5$, $\chi^1_{c5} = p_6$, $\chi^2_{c5} = p_7$, $\chi^2_{c4} = p_8$, $\chi^2_{c3} = p_9$, $\chi^2_{c2} = p_{10}$, and $\chi^2_{c1} = p_{11}$. The sets of activity places in ARTCs in the Petri net model shown in Fig. 2(a) are $\chi^1_{r_1^S}(A) = \{p_2, p_3\}$, $\chi^1_{r_2^S}(A) = \{p_4, p_5\}$, $\chi^1_{r_3^S}(A) = \{p_6, p_7\}$, $\chi^1_{r_4^S}(A) = \{p_8, p_9\}$, $\chi^2_{r_4^S}(A) = \{p_{10}, p_{11}\}$, $\chi^2_{r_3^S}(A) = \{p_{12}, p_{13}\}$, $\chi^2_{r_2^S}(A) = \{p_{14}, p_{15}\}$ and $\chi^2_{r_1^S}(A) = \{p_{16}, p_{17}\}$. Their corresponding algebraic sum form of $\chi^j_{r_i^S}(A)$ is $\chi^1_{c1} = p_2 + p_3$, $\chi^1_{c2} = p_4 + p_5$, $\chi^1_{c3} = p_6 + p_7$, $\chi^1_{c4} = p_8 + p_9$, $\chi^2_{c4} = p_{10} + p_{11}$, $\chi^2_{c3} = p_{12} + p_{13}$, $\chi^2_{c2} = p_{14} + p_{15}$ and $\chi^2_{c1} = p_{16} + p_{17}$. The sets of activity places in ARTCs in the Petri net model shown in Fig. 1(c) are $\chi^1_{r_1^S}(A) = \{p_2\}$, $\chi^1_{r_2^S}(A) = \{p_4\}$, $\chi^1_{r_3^S}(A) = \{p_6\}$, $\chi^2_{r_3^S}(A) = \{p_7\}$, $\chi^2_{r_2^S}(A) = \{p_9\}$, $\chi^2_{r_1^S}(A) = \{p_{11}\}$, $\chi^1_{r_4^U}(A) = \{p_3\}$, $\chi^1_{r_5^U}(A) = \{p_5\}$, $\chi^2_{r_6^U}(A) = \{p_8\}$, and $\chi^2_{r_7^U}(A) = \{p_{10}\}$. By Definition 10, their corresponding extended algebraic sum form of $\chi^j_{r_i^S}(A)$ is $\chi^1_{c1} = p_2 + p_3$, $\chi^1_{c2} = p_4 + p_5$, $\chi^1_{c3} = p_6$, $\chi^2_{c3} = p_7 + p_8$, $\chi^2_{c2} = p_9 + p_{10}$, and $\chi^2_{c1} = p_{11}$.

Definition 11: An active uncontrolled transition is a transition whose firing at a state leads a system to lose its liveness. Let Π denote the set of active uncontrolled transitions in a Petri net model.

Note that, the method of computing active uncontrolled transitions is presented in Algorithm 1.

Definition 12: Transition t_i^S is said to be a sink transition in a Petri net N with $N = (P^0 \cup P_A \cup P_R, T, F, W)$ if $t_i^S \in {}^{\bullet}(P^0)$. The set of sink transitions is denoted as $\Psi = \{t \mid t \in T, t \in {}^{\bullet}(P^0)\}$.

In the Petri net model shown in Fig. 1(a), the active uncontrolled transitions are $t_1^1 = t_1$ and $t_2^2 = t_7$, and the set of active uncontrolled transitions is $\Pi = \{t_1, t_7\}$. Firing t_1 at certain states would violate the desired specification requirements in the production sequence of process II. Similarly, firing t_7 at certain states would derive the system to undesired states in the production sequence of process I. To avoid the violation of the desired specifications, a separate supervisory structure is added to the uncontrolled Petri net model to supervise the firing of active uncontrolled transitions at each state by denying their firing at some specific states to prevent the violation of the specification requirements. Supervisory structure usually is constituted by control places to enforce liveness on a Petri net model. The sink transitions in the Petri net model shown in Fig. 1(a) are $t_1^S = t_6$ and $t_2^S = t_{12}$. The firing of sink transitions at any state cannot cause the system to violate the specification requirements. Those transitions are not included in the supervisory structure as there is no need to supervise their firings.

Definition 13: Let $\chi^j_{r_i^S}(A)$ be the set of activity places in an ARTC. $S(\hbar_j^N)$ is called a set of null places if ${}^\bullet(t_i^S) \subseteq \chi^j_{r_i^S}(A)$, then $S(\hbar_j^N) = \chi^j_{r_i^S}(A)$.

Definition 14: Let $\chi^j_{r_i^S}(A)$ be the set of activity places in an ARTC. $S(\hbar_j^A)$ is called a set of active places if $(t_i^j)^\bullet \subseteq \chi^j_{r_i^S}(A)$, then $S(\hbar_j^A) = \chi^j_{r_i^S}(A)$.

Definition 15: Let $S(\hbar_j^A)$ be a set of active places and $S(\hbar_j^N)$ be a set of null places in (N, M_0). The set of passive places is defined as $\Delta^j = P_{A_j} \backslash (S(\hbar_j^A) \cup S(\hbar_j^N))$.

Let us consider the Petri net model shown in Fig. 1(a). The sets of null places are $S(\hbar_1^N) = \{p_6\}$ and $S(\hbar_2^N) = \{p_{11}\}$. The sets of null places in the Petri net model shown in Fig. 2(a) are $S(\hbar_1^N) = \{p_8, p_9\}$ and $S(\hbar_2^N) = \{p_{16}, p_{17}\}$. While for the Petri net model shown in Fig. 1(c), the sets of null places are $S(\hbar_1^N) = \{p_6\}$ and $S(\hbar_2^N) = \{p_{11}\}$. The sets of active places in the Petri net model shown in Fig. 1(a) are $S(\hbar_1^A) = \{p_2\}$ and $S(\hbar_2^A) = \{p_7\}$. The sets of active places in the Petri net model shown in Fig. 2(a) are $S(\hbar_1^A) = \{p_2, p_3\}$ and $S(\hbar_2^A) = \{p_{10}, p_{11}\}$. The sets of active places in the Petri net model shown in Fig. 1(c) are $S(\hbar_1^A) = \{p_2, p_3\}$ and $S(\hbar_2^A) = \{p_7, p_8\}$.

The set of passive places is computed for each process of a Petri net model. In the Petri net model shown in Fig. 1(a), the sets of passive places are $\Delta^1 = \{p_3, p_4, p_5\}$ and $\Delta^2 = \{p_8, p_9, p_{10}\}$. The sets of passive places in the Petri net model shown in Fig. 2(a) are $\Delta^1 = \{p_4, p_5, p_6, p_7\}$ and $\Delta^2 = \{p_{12}, p_{13}, p_{14}, p_{15}\}$. The sets of passive places in the Petri net model shown in Fig. 1(c) are $\Delta^1 = \{p_4, p_5\}$ and $\Delta^2 = \{p_9, p_{10}\}$. The presence of tokens in the active places of process I with any passive place in process II would derive the states in the Petri net model to violate the specification requirements. Similarly, this is the same case to active places in process II with any passive place in process I.

IV. DERIVING GMECS FROM AN α_n-S^3PR

Generalized mutual exclusion constraints (GMECs) can be implemented through control places by using the marking-invariant law of Petri nets [53], [65] to form a place invariant associated with the activity places. Generally, a place invariant in a supervisor implementing a GMEC takes the form of $\alpha_1\lambda_1 + \alpha_2\lambda_2 + \cdots + \alpha_n\lambda_n \leq \beta$, where α_i, a non-negative integer, is called the control coefficient of the characteristic activity place in χ^j_{ci}, $\beta \in \{1, 2, \cdots\}$ is called the weighted token constant, λ_i is the marking of a characteristic activity place in χ^j_{ci}, and n is the number of activity places, i.e., $|P_A|$. The proposed method derives one GMEC for each process of a Petri net model with an α_n-S^3PR structure, which reduces the complexity of the supervisory structure. The objective function of the GMEC in the proposed method takes the form:

$$\alpha_j^A\lambda_j^A + \Sigma_{i\in\mathbb{N}}\alpha_i^j\lambda_i^j \leq \beta \tag{1}$$

where α_j^A is the control coefficient of a corresponding active place belonging to process I, and α_i^j is the control coefficient of a corresponding passive place in process II. With objective (1) for the GMEC, the constraints can be formulated by the combination of active places from the l'-th process with the places in the set of passive places from the l''-th process. The proposed method designs a global control place for each process in the Petri net model regardless of the number of resource places in it. To simplify the computational overhead of the proposed method, we consider two processes at a time to implement the GMECs. Suppose that process I is taken into account before process II in the Petri net model shown in Fig. 1(a). The GMEC on process I can be implemented as

$$\alpha_{l'}^A\lambda_{l'}^A + \alpha_{l''}^A\lambda_{l''}^A + \Sigma_{i\in\mathbb{N}}\alpha_i^{l''}\lambda_i^{l''} \leq \beta \tag{2}$$

satisfying the following constraints:

$$\begin{aligned}
\alpha_{l'}^A + \alpha_{l''}^A &\leq \beta + 1\\
\alpha_{l'}^A + \alpha_2^{l''} &\leq \beta + 1\\
\alpha_{l'}^A + \alpha_3^{l''} &\leq \beta + 1\\
&\vdots\\
\alpha_{l'}^A + \alpha_n^{l''} &\leq \beta + 1
\end{aligned}$$

Similarly, when we consider process II after implementing the GMEC on process I in the Petri net model shown in Fig. 1(a), the GMEC takes this form:

$$\alpha_{l''}^A\lambda_{l''}^A + \Sigma_{i\in\mathbb{N}}\alpha_i^{l'}\lambda_i^{l'} \leq \beta \tag{3}$$

satisfying the following constraints:

$$\begin{aligned}
\alpha_{l''}^A + \alpha_1^{l'} &\leq \beta + 1\\
\alpha_{l''}^A + \alpha_2^{l'} &\leq \beta + 1\\
&\vdots\\
\alpha_{l''}^A + \alpha_n^{l'} &\leq \beta + 1
\end{aligned}$$

Note that, the proposed approach in this paper considers two concurrent processes at a time to implement GMECs. Therefore, we have $j \in \{1, 2\}$. To deal with the general cases, let l' denote a process and l'' denote another process.

A. PETRI NET MODEL WITH AN α_1-S^3PR

Take as an example the Petri net model shown in Fig. 1(a) with two concurrent processes, i.e., processes I and II. Implementing the GMEC on process I before process II takes the form of Eq. (2) while implementing the GMEC on process II after implementing the GMEC on process I takes the form of Eq. (3). Suppose that process I is considered before process II. The objective function of the GMEC is

$$\alpha_1^A\lambda_1^A + \alpha_2^A\lambda_2^A + \alpha_7^2\lambda_7^2 + \alpha_8^2\lambda_8^2 + \alpha_9^2\lambda_9^2 \leq \beta \tag{4}$$

TABLE 1. Summary of parameters for the Petri net model shown in Fig. 1(a).

Set of activity place ($\chi^j_{r^S_i}(A)$)	Algebraic sum form (χ^j_{ci})	Set of active places $S(\hbar^A_j)$	Set of passive places $S(\hbar^j_i)$	Marking of active places (λ^A_j)	Marking of passive places (λ^j_i)	Control coefficient of λ^A_j (α^A_j)	Control coefficient of λ^j_i (α^A_j)
$\chi^1_{r^S_1}(A) = \{p_2\}$	$\chi^1_{c1} = p_2$	$S(\hbar^A_1) = \{p_2\}$	–	λ^A_1	–	α^A_1	–
$\chi^1_{r^S_2}(A) = \{p_3\}$	$\chi^1_{c2} = p_3$	–	$S(\hbar^1_2) = \{p_3\}$	–	λ^1_2	–	α^1_2
$\chi^1_{r^S_3}(A) = \{p_4\}$	$\chi^1_{c3} = p_4$	–	$S(\hbar^1_3) = \{p_4\}$	–	λ^1_3	–	α^1_3
$\chi^1_{r^S_4}(A) = \{p_5\}$	$\chi^1_{c4} = p_5$	–	$S(\hbar^1_4) = \{p_5\}$	–	λ^1_4	–	α^1_4
$\chi^1_{r^S_5}(A) = \{p_6\}$	$\chi^1_{c5} = p_6$	–	–	–	–	–	–
$\chi^2_{r^S_5}(A) = \{p_7\}$	$\chi^2_{c5} = p_7$	$S(\hbar^A_2) = \{p_7\}$	–	λ^A_2	–	α^A_2	–
$\chi^2_{r^S_4}(A) = \{p_8\}$	$\chi^2_{c4} = p_8$	–	$S(\hbar^2_7) = \{p_8\}$	–	λ^2_7	–	α^2_7
$\chi^2_{r^S_3}(A) = \{p_9\}$	$\chi^2_{c3} = p_9$	–	$S(\hbar^2_8) = \{p_9\}$	–	λ^2_8	–	α^2_8
$\chi^2_{r^S_2}(A) = \{p_{10}\}$	$\chi^2_{c2} = p_{10}$	–	$S(\hbar^2_9) = \{p_{10}\}$	–	λ^2_9	–	α^2_9
$\chi^2_{r^S_1}(A) = \{p_{11}\}$	$\chi^2_{c1} = p_{11}$	–	–	–	–	–	–

satisfying the following constraints:

$$\alpha^A_1 + \alpha^A_2 \leq \beta + 1$$
$$\alpha^A_1 + \alpha^2_7 \leq \beta + 1$$
$$\alpha^A_1 + \alpha^2_8 \leq \beta + 1$$
$$\alpha^A_1 + \alpha^2_9 \leq \beta + 1$$

On the other hand, when we consider process II after implementing the GMEC on process I, the GMEC on process II can be implemented as follows:

$$\alpha^A_2\lambda^A_2 + \alpha^1_2\lambda^1_2 + \alpha^1_3\lambda^1_3 + \alpha^1_4\lambda^1_4 \leq \beta \quad (5)$$

satisfying the following constraints:

$$\alpha^A_2 + \alpha^1_2 \leq \beta + 1$$
$$\alpha^A_2 + \alpha^1_3 \leq \beta + 1$$
$$\alpha^A_2 + \alpha^1_4 \leq \beta + 1$$

The unknown control coefficients in both Eqs. (4) and (5) can be evaluated using an integer linear programming problem (ILPP) solver. Table 1 presents the summary of the relationship among the parameters for the Petri net model shown in Fig. 1(a).

B. PETRI NET MODEL WITH AN α_2-S^4PR

The GMECs for a Petri net model with an α_2-S^{4}PR structure are obtained by splitting the Petri net model into several sub-net models of $N = \bigcirc_{i=m} N^S_i = (P_{A_i} \cup \{P^0\} \cup P_{R_i}, T, F_i)$, such that each sub-net can be treated in Eqs. (2) and (3) to derive their GMECs. Finally, the computed GMECs for all the sub-nets are the total GMECs for the whole Petri net model.

Let us deal with a sub-net model $(N^S_1,\ M^S_0)$ shown in Fig. 2(a). If we consider process I before process II, the objective function of the GMEC is:

$$\alpha^A_1\lambda^A_1 + \alpha^A_2\lambda^A_2 + \alpha^2_6\lambda^2_6 + \alpha^2_7\lambda^2_7 \leq \beta \quad (6)$$

satisfying the following constraints:

$$\alpha^A_1 + \alpha^A_2 \leq \beta + 1$$
$$\alpha^A_1 + \alpha^2_6 \leq \beta + 1$$
$$\alpha^A_1 + \alpha^2_7 \leq \beta + 1$$

Similarly, when we consider process II after implementing the GMEC on process I of the sub-net $(N^S_1,\ M^S_0)$ shown in Fig. 2(a), the GMEC on process II is expressed as:

$$\alpha^A_2\lambda^A_2 + \alpha^1_2\lambda^1_2 + \alpha^1_3\lambda^1_3 \leq \beta \quad (7)$$

satisfying the following constraints:

$$\alpha^A_2 + \alpha^1_2 \leq \beta + 1$$
$$\alpha^A_2 + \alpha^1_3 \leq \beta + 1$$

Similarly, the same procedure can be applied to sub-net $(N^S_2,\ M^S_0)$ as shown in Fig. 2(b). The GMECs derived from sub-net $(N^S_1,\ M^S_0)$ and $(N^S_2,\ M^S_0)$ are the total GMECs for the whole Petri net model shown in Fig. 1(b). The summary of the relationship among the parameters in the Petri net model shown in Fig. 2(a) is provided in Table 2.

C. PETRI NET MODEL WITH AN α_3-S^3PR

When deriving GMECs for a Petri net model with more than two concurrent processes, we first split-up the Petri net model into several sub-net models of $N = \bigcirc_{i=m} N^S_i = (P_{A_i} \cup \{P^0\} \cup P_{R_i}, T, F_i)$, such that each sub-net has two concurrent processes that satisfy Definition 9. Eqs. (2) and (3) are used to derive their GMECs for each sub-net. The GMECs for the whole Petri net model are those GMECs of all the sub-nets.

Let us consider a Petri net model shown in Fig. 1(c). To implement the GMEC on process I before process II, the objective function of the GMEC on process I is

$$\alpha^A_1\lambda^A_1 + \alpha^A_2\lambda^A_2 + \alpha^2_5\lambda^2_5 \leq \beta \quad (8)$$

satisfying the following constraints:

$$\alpha^A_1 + \alpha^A_2 \leq \beta + 1$$
$$\alpha^A_1 + \alpha^2_5 \leq \beta + 1$$

To implement the GMEC on process II after implementing the GMEC on process I of the Petri net model shown in Fig. 1(c), the GMEC on process II is expressed as

$$\alpha^A_2\lambda^A_2 + \alpha^1_2\lambda^1_2 \leq \beta \quad (9)$$

satisfying the following constraints:

$$\alpha^A_2 + \alpha^1_2 \leq \beta + 1$$

TABLE 2. Summary of parameters for the Petri net model shown in Fig. 2(a).

Set of activity places ($\chi^j_{r^S_i}(A)$)	Algebraic sum form (χ^j_{ci})	Set of active places $S(\hbar^A_j)$	Set of passive places $S(\hbar^j_i)$	Marking of active place (λ^A_j)	Marking of passive places (λ^j_i)	Control coefficient of λ^A_j (α^A_j)	Control coefficient of λ^j_i (α^A_j)
$\chi^1_{r^S_1}(A) = \{p_2, p_3\}$	$\chi^1_{c1} = p_2 + p_3$	$S(\hbar^A_1) = \{p_2 + p_3\}$	–	λ^A_1	–	α^A_1	–
$\chi^1_{r^S_2}(A) = \{p_4, p_5\}$	$\chi^1_{c2} = p_4 + p_5$	–	$S(\hbar^1_2) = \{p_4 + p_5\}$	–	λ^1_2	–	α^1_2
$\chi^1_{r^S_3}(A) = \{p_6, p_7\}$	$\chi^1_{c3} = p_6 + p_7$	–	$S(\hbar^1_3) = \{p_6 + p_7\}$	–	λ^1_3	–	α^1_3
$\chi^1_{r^S_4}(A) = \{p_8, p_9\}$	$\chi^1_{c4} = p_8 + p_9$	–	–	–	–	–	–
$\chi^2_{r^S_4}(A) = \{p_{10}, p_{11}\}$	$\chi^2_{c4} = p_{10} + p_{11}$	$S(\hbar^A_2) = \{p_{10} + p_{11}\}$	–	λ^A_2	–	α^A_2	–
$\chi^2_{r^S_3}(A) = \{p_{12}, p_{13}\}$	$\chi^2_{c3} = p_{12} + p_{13}$	–	$S(\hbar^2_6) = \{p_{12} + p_{13}\}$	–	λ^2_6	–	α^2_6
$\chi^2_{r^S_2}(A) = \{p_{14}, p_{15}\}$	$\chi^2_{c2} = p_{14} + p_{15}$	–	$S(\hbar^2_7) = \{p_{14} + p_{15}\}$	–	λ^2_7	–	α^2_7
$\chi^2_{r^S_1}(A) = \{p_{16}, p_{17}\}$	$\chi^2_{c1} = p_{16} + p_{17}$	–	–	–	–	–	–

TABLE 3. Summary of parameters for the Petri net model shown in Fig. 1(c).

Algebraic sum form (χ^j_{ci})	Set of active places $S(\hbar^A_j)$	Set of passive places $S(\hbar^j_i)$	Marking of active places (λ^A_j)	Marking of passive places (λ^j_i)	Control coefficient of λ^A_j (α^A_j)	Control coefficient of λ^j_i (α^A_l)
$\chi^1_{c1} = p_2 + p_3$	$S(\hbar^A_1) = \{p_2 + p_3\}$	–	λ^A_1	–	α^A_1	–
$\chi^1_{c2} = p_4 + p_5$	–	$S(\hbar^1_2) = \{p_4 + p_5\}$	–	λ^1_2	–	α^1_2
$\chi^1_{c3} = p_6$	–	–	–	–	–	–
$\chi^2_{c3} = p_7 + p_8$	$S(\hbar^A_2) = \{p_7 + p_8\}$	–	λ^A_2	–	α^A_2	–
$\chi^2_{c2} = p_9 + p_{10}$	–	$S(\hbar^2_5) = \{p_9 + p_{10}\}$	–	λ^2_5	–	α^2_5
$\chi^2_{c1} = p_{11}$	–	–	–	–	–	–

The control coefficients remain unknown and are to be found using a method developed in this paper. Table 3 summarizes the relationships among the parameters of the Petri net model shown in Fig. 1(c).

V. FROM ILPP TO LINEAR INEQUALITIES

The solution to the control coefficients is usually done using ILPP solver. However, many variables need to be generated in the process of solving ILPP, which makes the computation difficult. This section presents a new method to compute the control coefficients from the GMECs without solving an ILPP. Firstly, Eq. (1) is modified. The modified Eq. (1) provides an efficient way to compute the control coefficients using structural properties of the Petri net model.

A. PETRI NET MODEL WITH AN α_1-S^3PR

The modified Eq. (10) that greatly simplifies the computation of control coefficients in the GMECs can be expressed as

$$\alpha^A_j \lambda^A_j + \Sigma_{i\in\mathbb{N}} \lambda^j_i \leq \beta \tag{10}$$

where

$$\alpha^A_j = |\Delta^j| + x, \quad \beta = \alpha^A_j \quad \text{and } x \in \mathbb{N}$$

Generally, the modified GMEC on process I before process II in a Petri net model with an α_1-S^3PR structure is

$$\alpha^A_{l'} \lambda^A_{l'} + \lambda^A_{l''} + \Sigma_{i\in\mathbb{N}} \lambda^{l''}_i \leq \beta \tag{11}$$

with

$$\alpha^A_{l'} = |\Delta^{l''}| + 1 \quad \text{and } \beta = \alpha^A_{l'}$$

Similarly, the modified GMEC on process II after implementing the GMECs on process I is

$$\alpha^A_{l''} \lambda^A_{l''} + \Sigma_{i\in\mathbb{N}} \lambda^{l'}_i \leq \beta \tag{12}$$

with

$$\alpha^A_{l'} = |\Delta^{l'}| \quad \text{and } \beta = \alpha^A_{l'}$$

B. PETRI NET MODEL WITH AN α_2-S^4PR

For a Petri net model with an α_2-S^4PR structure, the modified GMEC can be expressed in the form of Eq. (10):

$$\alpha^A_j \lambda^A_j + \Sigma_{i\in\mathbb{N}} \lambda^j_i \leq \beta \tag{13}$$

where

$$\alpha^A_j = \frac{|\Delta^j| + z}{x}, \quad \beta = \alpha^A_j \quad \text{and } x, z \in \mathbb{N}$$

Implementing the GMEC on process I of a sub-net (N^S_i, M^S_0) shown in Fig. 2(a) before implementing the GMEC on process II is

$$\alpha^A_{l'} \lambda^A_{l'} + \lambda^A_{l''} + \Sigma_{i\in\mathbb{N}} \lambda^{l''}_i \leq \beta \tag{14}$$

with

$$\alpha^A_{l'} = \frac{(|\Delta^{l'}| + 2)}{2} \quad \text{and } \beta = \alpha^A_{l'}$$

Similarly, the modified GMEC on process II after implementing the GMEC on process I is

$$\alpha^A_{l''} \lambda^A_{l''} + \Sigma_{i\in\mathbb{N}} \lambda^A_{l'} \leq \beta \tag{15}$$

with

$$\alpha^A_{l''} = \frac{|\Delta^{l'}|}{2} \quad \text{and } \beta = \alpha^A_{l''}$$

C. PETRI NET MODEL WITH AN α_3-S^3PR

Finally, let us consider a Petri net model with an α_3-S^3PR structure. The modified GMEC is

$$\alpha_j^A \lambda_j^A + \Sigma_{i \in \mathbb{N}} \lambda_i^j \leq \beta \tag{16}$$

where

$$\alpha_j^A = \frac{|\Delta^j| + z}{x}, \quad \beta = 2\alpha_j^A + 1 \quad \text{and } x, z \in \mathbb{N}$$

Implementing the modified GMEC on process I before implementing the GMEC on process II is

$$\alpha_{l'}^A \lambda_{l'}^A + \lambda_{l''}^A + \Sigma_{i \in \mathbb{N}} \lambda_i^{l''} \leq \beta \tag{17}$$

with

$$\alpha_{l'}^A = \frac{(|\Delta^{l'}| + 2)}{2} \quad \text{and } \beta = 2\alpha_{l'}^A + 1$$

Similarly, the modified GMEC on process II after implementing the GMEC on process I is

$$\alpha_{l''}^A \lambda_{l''}^A + \Sigma_{i \in \mathbb{N}} \lambda_{l'}^A \leq \beta \tag{18}$$

with

$$\alpha_{l''}^A = \frac{|\Delta^{l'}|}{2} \quad \text{and } \beta = 2\alpha_{l''}^A + 1$$

The modified versions of GMECs ease the computational overhead as it greatly simplifies the evaluation of the unknown control coefficients in the modified GMECs.

D. FROM GMECs TO CONTROL PLACES

This section presents a method to design control places from the computed GMECs. For each computed GMEC from the uncontrolled Petri net model, a control place is designed to prohibit the reachability of the unsafe states.

The study carried out by [53] developed an efficient method to construct a supervisor based on the concept of place invariants. The supervisor consists of all necessary control places that would enforce the constraints of the Petri net model to follow the desired system specification. Let $[N_p]$ be the incidence matrix of a plant net. The control places together with the arcs connecting these places and transitions of the plant net can be represented by a matrix $[N_c]$. The incidence matrix $[N]$ of the controlled net is

$$[N] = \begin{bmatrix} N_p \\ N_c \end{bmatrix}$$

Control places are designed after computing all the necessary GMECs from a Petri net model. Let L_D be the incidence matrix of the GMEC and D be the row matrix of the places in the GMEC. The control place (V_i) is computed as follows:

$$V_i = -DL_D \tag{19}$$

where D is a row matrix with a dimension of $1 \times |P_i|$ and L_D is a matrix with the dimension of $|P_i| \times |T_i|$ while the dimension of V_i is $1 \times |T_i|$ with P_i being the places in the GMEC and T_i representing the transitions associated with the places in the GMEC. After including all the computed control places to the uncontrolled Petri net model, the liveness property of the Petri net model is enforced.

Suppose that $5\lambda_1^A + \lambda_5^2 + \lambda_6^2 + \lambda_7^2 + \lambda_8^2 + \lambda_9^2 \leq 5$ is the GMEC from a Petri net model shown in Fig. 1(a). The places associated with the markings of the GMEC are p_2, p_6, p_7, p_8, p_9 and p_{10} and the corresponding transitions to those places are t_1, t_2, t_6, t_7, t_8, t_9, t_{10} and t_{11}. Using Eq. (19), the control place is computed as follows:

$$V_i = -\begin{bmatrix} 5 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$V_i = \begin{bmatrix} -5 & 5 & -1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

VI. DEADLOCK PREVENTION POLICY

This section presents two algorithms to compute a Petri net supervisor. The first computes the active uncontrolled transitions in a Petri net model. The second designs a Petri net supervisor that enforces the uncontrolled Petri net model to satisfy system specification requirements.

Algorithm 1 Computation of Active Uncontrolled Transitions

Input: A Petri net (N, M_0) with $N = (P^0 \cup P_A \cup P_R, T, F)$ prone to deadlocks
Output: Set of active uncontrolled transitions Π

```
Π = ∅; j = 1, 2, ···, n.  /*j is the number of concurrent process in the net */
while j ≤ n do
    Identify (p_j^0) and P_{A_j} = {p_i, p_{i+1}, p_{i+2}, ···, p_n}
    Π* = ∅
    Compute (p_j^0)••; let p_i be the activity place in (p_j^0)••; choose a resource place r such that p_i ∈ ||I_r||, and compute ||χ_i(A)|| ∪ ||χ_i(r)|| = ||χ_(r,p_i)|| = {r, p|r ∈ P_R, p ∈ P_A}
    for (i = 1, |P_{A_j}|, i++) do
        if ||χ_(r,p_i)|| ≠ ||I_r|| then
            Choose an activity place p_{i+1} from p_i••; find a resource place r such that p_{i+1} ∈ ||I_r||; compute ||χ_(r,p_{i+1})|| = {r, p_{i+1}|r ∈ P_R, p_{i+1} ∈ P_A, p_{i+1} ∈ ||I_r||}
        else
            Π* = Π* ∪ •p_i
        end
    end
    j = j + 1, Π = Π ∪ Π*
end
Output the set of active uncontrolled transitions Π.
```

Algorithm 1 definitely computes all the active uncontrolled transitions from an uncontrolled Petri net model or sub-net model if such transitions exist. As aforementioned, firing an

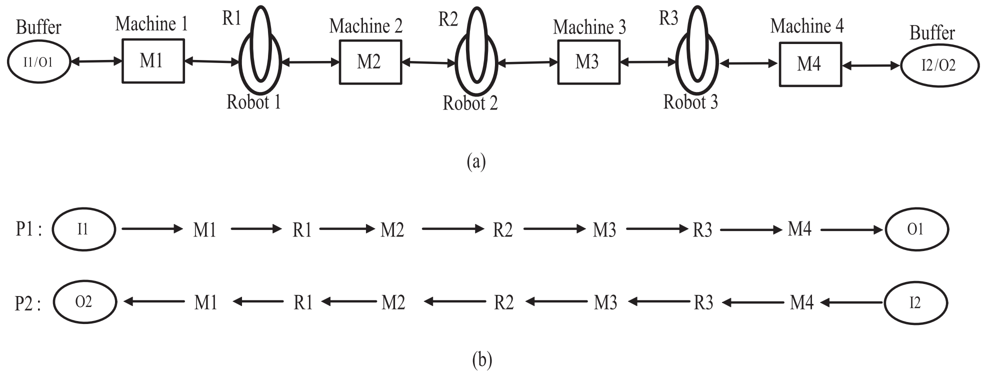

FIGURE 3. (a) The layout of an FMS and (b) its production routes.

active uncontrolled transition at some specific states of a Petri net model leads the system to the deadlock zone.

Theorem 1: Algorithm 1 can compute active uncontrolled transitions from an uncontrolled Petri net or sub-net model under consideration if such transitions exist.

Proof 1: Suppose that $t_i^j \in T$ exists in a Petri net model $N = (P,\ T,\ F)$ or sub-net model. It implies that there exists a minimal semi-flow $||I_{r^s}||$ formed by shared resource places r^s in the Petri net model or sub-net model. Since each process has a single idle place, the decision on the first-met $||I_{r^s}||$ along the directional flow of tokens for an idle place in the j-th process would result in $||\chi_i(A)|| \cup ||\chi_i(r)|| = \{r, p_i | r \in P_R, p_i \in P_A\}$. Therefore, $p_i \in P_A$ is an active place in the j-th process. Hence, each active place corresponds to an active uncontrolled transition in the Petri net model (i.e., the preset of p_i contains the active uncontrolled transition t_i^j). □

Algorithm 2 can compute all the necessary GMECs in a Petri net model or a sub-net model. Furthermore, it provides a method to design a control place for each computed GMEC. The computed control places enforce liveness of the Petri net model when they are included in the uncontrolled Petri net model.

Theorem 2: The computed control places by Algorithm 2 enforce liveness to an uncontrolled Petri net model.

Proof 2: Each GMEC generated from Algorithm 2 corresponds to an active uncontrolled transition computed from Algorithm 1. Since all the transitions in (N, M_0) are controllable, the control places derived from the GMECs in the Petri net model $N = (P, T, F)$ would make the active uncontrolled transitions controlled. Hence, liveness is assured. □

Example 1: Let us consider an FMS shown in Fig. 3. The FMS consists of four machines M1–M4, each of which can process only one part at a time, and three robots R1–R3, each of which can hold one part at a time. Parts enter the system through input/output buffers I1/O1 and I2/O2. Two part types P1 and P2 are considered in the production sequence. Initially it is assumed that there are no parts in the system.

Its equivalent representation of the FMS shown in Fig. 3 is depicted in Fig. 4 using Petri nets. The places in the Petri net model are represented as $P = P^0 \cup P_A \cup P_R$,

---

Algorithm 2 Computation of Control Places

Input: Petri net model (N, M_0) of an FMS with $N = (P^0 \cup P_A \cup P_R, T, F)$ prone to deadlocks

Output: Controlled Petri net model

1: Compute the set of active uncontrolled transitions using Algorithm 1.
2: Compute the set of active places using Definition 14.
3: Compute the set of null places using Definition 13.
4: Compute the set of passive places for each process using Definition 15.
5: Identify each process j in (N, M_0), $j = 1, 2$.
6: Compute the GMEC using,

$$GMEC = \begin{cases} \alpha_{l'}^A \lambda_{l'}^A + \lambda_{l''}^A + \Sigma_{p \in P} \lambda_i^{l''} \leq \beta, & j = 1; \\ \alpha_{l''}^A \lambda_{l''}^A + \Sigma_{p \in P} \lambda_{l'}^A \leq \beta, & j = 2. \end{cases}$$

7: Design control place V_i,

$$V_i = -DL_D$$

8: Add all computed control places to the uncontrolled Petri net model.
9: Output a controlled Petri net model with liveness.

---

where $P^0 = \{p_1, p_{16}\}$, $P_R = \{p_{17}, \cdots, p_{23}\}$ and $P_A = \{p_2, \cdots, p_{15}\}$. Places $p_2, p_3, p_4, p_5, p_6, p_7$ and p_8 represent the operations of M1, R1, M2, R2, M3, R3 and M4 for the production sequence of part type P1, respectively. Similarly, for the production sequence of part type P2, places $p_9, p_{10}, p_{11}, p_{12}, p_{13}, p_{14}$ and p_{15} represents the operations of M4, R3, M3, R2, M2, R1 and M1, respectively. Places p_1 and p_{16} represent the I1/O1 and I2/O2 buffers, respectively. While places $p_{17}, p_{18}, p_{19}, p_{20}, p_{21}, p_{22}$ and p_{23} represent the shared resources of M1, R1, M2, R2, M3, R3 and M4, respectively.

To demonstrate Algorithm 2, let us consider the Petri net model of Example 1 shown in Fig. 4. As aforementioned, the set of active uncontrolled transitions is $\Pi = \{t_1, t_9\}$ with $t_1^1 =$

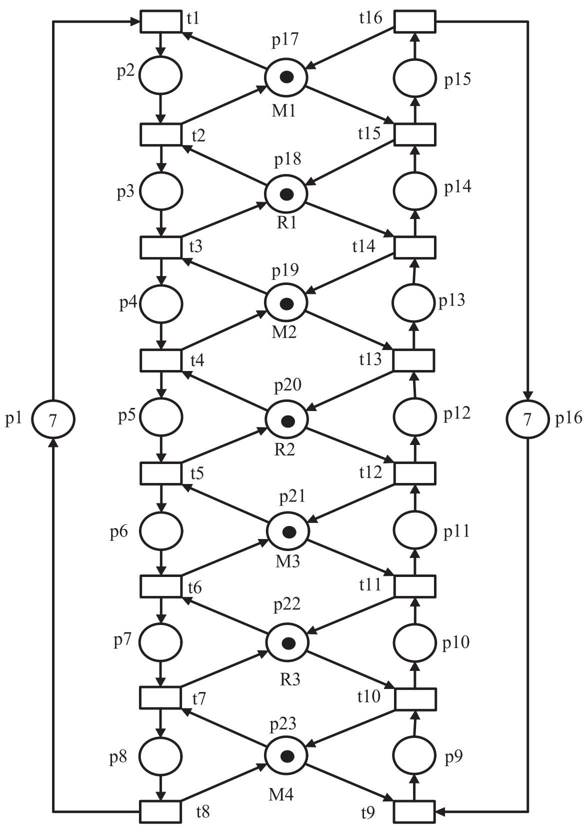

FIGURE 4. The Petri net model with an α_1-S^3PR for Example 1.

TABLE 4. Summary of the parameters for the supervisory structure of the Petri net model shown in Fig. 4.

S/N	GMECs	V_i	$^\bullet V$	$V^\bullet$	M_0
1	$\lambda_1^A + \lambda_2^A + \lambda_7^2 + \lambda_8^2 + \lambda_9^2 + \lambda_{10}^2 + \lambda_{11}^2 \leq 6$	V_1	$6t_2, t_{15}$	$6t_1, t_9$	6
2	$\lambda_2^A + \lambda_2^1 + \lambda_3^1 + \lambda_4^1 + \lambda_5^1 + \lambda_6^1 \leq 5$	V_2	$t_7, 5t_{10}$	$t_2, 5t_9$	5

t_1 and $t_2^2 = t_9$. The sets of active place are $S(\hbar_1^A) = \{p_2\}$ and $S(\hbar_2^A) = \{p_9\}$ and the sets of null place are $S(\hbar_1^N) = \{p_8\}$ and $S(\hbar_2^N) = \{p_{15}\}$, while the sets of passive places are $\Delta^1 = \{p_3, p_4, p_5, p_6, p_7\}$ and $\Delta^2 = \{p_{10}, p_{11}, p_{12}, p_{13}, p_{14}\}$. The GMECs are $\lambda_1^A + \lambda_2^A + \lambda_7^2 + \lambda_8^2 + \lambda_9^2 + \lambda_{10}^2 + \lambda_{11}^2 \leq 6$ with its associated places being $p_2, p_9, p_{10}, p_{11}, p_{12}, p_{13}$, and p_{14} and $\lambda_2^A + \lambda_2^1 + \lambda_3^1 + \lambda_4^1 + \lambda_5^1 + \lambda_6^1 \leq 5$ with its associated places being p_9, p_3, p_4, p_5, p_6, and p_7. The transitions corresponding to the places in the GMECs are $t_1, t_2, t_9, t_{10}, t_{11}, t_{12}, t_{13}, t_{14}$, and t_{15}; and $t_2, t_3, t_4, t_5, t_6, t_7, t_9$, and t_{10}, respectively. The control places corresponding to GMECs are, respectively,

$$V_1 = -\begin{bmatrix} 6 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}^T \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$V_1 = \begin{bmatrix} -6 & 6 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$V_2 = -\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 5 \end{bmatrix}^T \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 1 & -5 & 5 \end{bmatrix}$$

When the two control places are added to the uncontrolled Petri net model shown in Fig. 4, the resulting Petri net model is live with maximally permissive behavior of 255 states. Table 4 provides the detailed parameters of the computed control places, while Table 5 presents the performance comparison with some deadlock control policies available in the literature in terms of control places, control arcs and the number of reachable states.

TABLE 5. Performance comparison of the proposed method with some of the existing policies for a Petri net model shown in Fig. 4.

Policies	No. of control places	No. of control arcs	No. of reachable states
[7]	11	44	255
[37]	11	44	255
[10]	11	44	255
[6]	11	44	255
[49]	11	44	255
Proposed method	2	8	255

VII. EXPERIMENTAL EXAMPLES

This section presents an experimental result to expose the applicability of the proposed method. All the computation of the supervisors is done without computing the reachability graph or siphons of a Petri net model. The proposed method utilizes the resource transition circuits to compute a supervisor that enforces liveness on a Petri net model.

Example 2: Let us consider the Petri net model of a flexible manufacturing system shown in Fig. 5. There are 31 places and 22 transitions. The places have the set partition: $P^0 = \{p_1, p_{22}\}$, $P_R = \{p_{23} - p_{31}\}$ and $P_A = \{p_2 - p_{21}\}$. The Petri net model has 564 reachable states, with 287 and 277 states being in the live zone and deadlock zone, respectively.

The Petri net model shown in Fig. 5 (Example 2) has an α_2-S^4PR structure. It is decomposed into two sub-nets denoted by $N = N_1^S \circ N_2^S$. The control places of each sub-net can be computed via Algorithm 2. First, let us consider (N_1^S, M_0^S) shown in Fig. 6(a). The places in (N_1^S, M_0^S) have the set partition: $P^0 = \{p_1, p_{22}\}$, $P_{R_1} = \{p_{23}, p_{25}, p_{27}, p_{29}, p_{31}\}$ and $P_A = \{p_2 - p_{21}\}$. When the uncontrolled sub-net (N_1^S, M_0^S) is simulated using integrated net analyzer (INA), the sub-net has 1053 reachable states among which 485 states are in the live zone and 568 states are in the deadlock zone.

Using Algorithm 2, the set of active uncontrolled transitions is $\Pi = \{t_1, t_{12}\}$ with $t_1^1 = t_1$ and $t_2^2 = t_{12}$. The sets of active places are $S(\hbar_1^A) = \{p_2, p_3\}$ and $S(\hbar_2^A) = \{p_{12}, p_{13}\}$. The sets of null places are $S(\hbar_1^N) = \{p_{10}, p_{11}\}$ and $S(\hbar_2^N) = \{p_{20}, p_{21}\}$. The sets of passive places are $\Delta^1 = \{p_4, p_5, p_6, p_7, p_8, p_9\}$ and $\Delta^2 = \{p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$. The GMECs associated with processes I and II are,

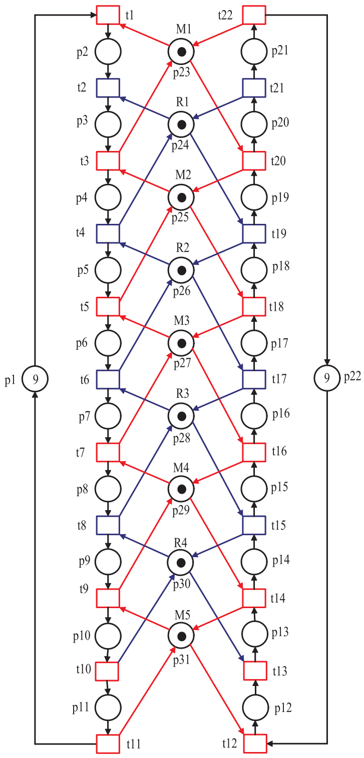

FIGURE 5. The Petri net model with an α_2-S^{4}PR structure for Example 2.

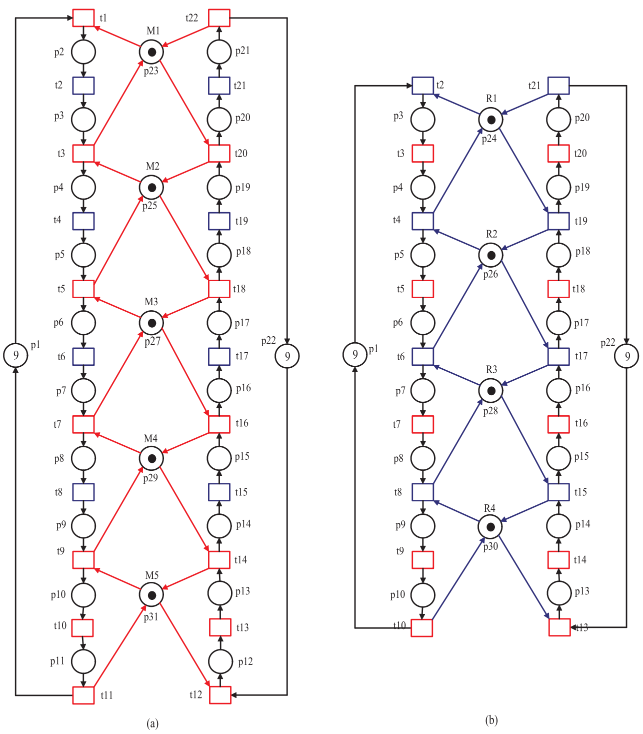

FIGURE 6. (a) The sub-net model (N_1^S, M_0^S), and (b) The sub-net model (N_2^S, M_0^S).

respectively, $4\lambda_1^1+\lambda_2^2+\lambda_7^2+\lambda_8^2+\lambda_9^2 \leq 4$ with its places associated with the GMEC being $p_2, p_3, p_{12}, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}$ and $3\lambda_2^2 + \lambda_2^1 + \lambda_3^1 + \lambda_4^1 \leq 2$ with its places associated with the GMEC being $p_4, p_5, p_6, p_7, p_8, p_9, p_{12}, p_{13}$. The transitions corresponding to the places in the GMECs are $t_1, t_2, t_3, t_{12}, t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}, t_{20}$ and $t_3, t_4, t_5, t_6, t_7, t_8, t_9, t_{12}, t_{13}, t_{14}$, respectively. The corresponding control places for the GMECs respectively are:

$$V_1 = -\begin{bmatrix} 4 \\ 4 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}^T \times \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$V_1 = \begin{bmatrix} -4 & 0 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$V_2 = -\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 3 \\ 3 \end{bmatrix}^T \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 1 & -3 & 0 & 3 \end{bmatrix}$$

When the two control places are added to the uncontrolled sub-net (N_1^S, M_0^S), the sub-net model is live with maximally permissive behavior of 485 good states.

Similarly, let us consider the uncontrolled sub-net (N_2^S, M_0^S) shown in Fig. 6(b). The places in the Petri net model (N_2^S, M_0^S) have the set partition: $P^0 = \{p_1, p_{22}\}$, $P_{R_2} = \{p_{24}, p_{26}, p_{28}, p_{30}\}$ and $P_A = \{p_3 - p_{10}, p_{13} - p_{20}\}$. When the sub-net model is simulated using INA, the sub-net model has 297 reachable states among which 161 states are in the live zone and 136 states are in the deadlock zone.

Applying Algorithm 2 to the sub-net (N_2^S, M_0^S), the set of active uncontrolled transitions is $\Pi = \{t_2, t_{13}\}$ with $t_1^1 = t_2$ and $t_2^2 = t_{13}$. The sets of active places are $S(\hbar_1^A) = \{p_3, p_4\}$ and $S(\hbar_2^A) = \{p_{13}, p_{14}\}$. The sets of null places are $S(\hbar_1^N) = \{p_9, p_{10}\}$ and $S(\hbar_2^N) = \{p_{19}, p_{20}\}$. The sets of passive places are $\Delta^1 = \{p_5, p_6, p_7, p_8\}$ and $\Delta^2 = \{p_{15}, p_{16}, p_{17}, p_{18}\}$. The GMECs of processes I and II are, respectively, $3\lambda_1^A + \lambda_2^A + \lambda_7^2 + \lambda_8^2 \leq 3$ with the places corresponding to the

GMEC being $p_3, p_4, p_{13}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}$ and $2\lambda_2^A + \lambda_2^1 + \lambda_3^1 \leq 1$ with the places associated with the GMEC being $p_5, p_6, p_7, p_8, p_{13}, p_{14}$. The corresponding control places of the GMECs respectively are:

$$V_1 = -\begin{bmatrix} 3\\3\\1\\1\\1\\1\\1\\1 \end{bmatrix}^T \times \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 \end{bmatrix}$$

$$V_1 = \begin{bmatrix} -3 & 0 & 3 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$V_2 = -\begin{bmatrix} 1\\1\\1\\1\\2\\2 \end{bmatrix}^T \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$V_2 = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 & -2 & 0 & 2 \end{bmatrix}$$

When the two control places are included in the sub-net $(N_2^S,\ M_0^S)$ shown in Fig. 6(b), the sub-net model $(N_2^S,\ M_0^S)$ is live with maximally permissive behavior of 161 states.

The computed control places for the sub-nets (N_1^S, M_0^S) and (N_2^S, M_0^S) are included in the Petri net model shown in Fig. 5. The resulting Petri net model is live with maximally permissive behavior of 485 good states. Table 6 provides the detailed parameters of the computed control places. While Table 7 compares the performance of the proposed method with some deadlock control policies available in the literature in terms of the number of control places, control arcs and reachable states.

TABLE 6. Summary of the supervisory structure parameters for the Petri net model shown in Fig. 5.

S/N	GMECs	V_i	${}^{\bullet}V$	$V^{\bullet}$	M_0
1	$4\lambda_1^A + \lambda_2^A + \lambda_7^2 + \lambda_8^2 + \lambda_9^2 \leq 4$	V_1	$4t_3, t_{20}$	$4t_1, t_{12}$	4
2	$3\lambda_2^A + \lambda_2^1 + \lambda_3^1 + \lambda_4^1 \leq 3$	V_2	$t_9, 3t_{14}$	$t_3, 3t_{12}$	3
3	$3\lambda_1^A + \lambda_2^A + \lambda_6^2 + \lambda_7^2 \leq 3$	V_3	$3t_4, t_{20}$	$3t_2, t_{12}$	3
4	$2\lambda_2^A + \lambda_2^1 + \lambda_3^1 \leq 2$	V_4	$t_8, 2t_{15}$	$t_4, 2t_{13}$	2

Example 3: Consider the Petri net model of an FMS shown in Fig. 7. It has 33 places and 21 transitions. The places have the set partition: $P^0 = \{p_1, p_{20}, p_{21}\}$, $P_R = \{p_{22} - p_{33}\}$ and $P_A = \{p_2 - p_{19}\}$. The Petri net model has 42880 reachable states with 39800 and 3080 states being in the live and deadlock zones, respectively.

TABLE 7. Performance comparison between the proposed method and some existing policies for the Petri net model shown in Fig. 5.

Policies	No. of control places	No. of arcs	No. of reachable states
[7]	12	48	485
[10]	12	48	485
[27]	12	48	485
[6]	12	48	485
[49]	12	48	485
proposed method	4	16	485

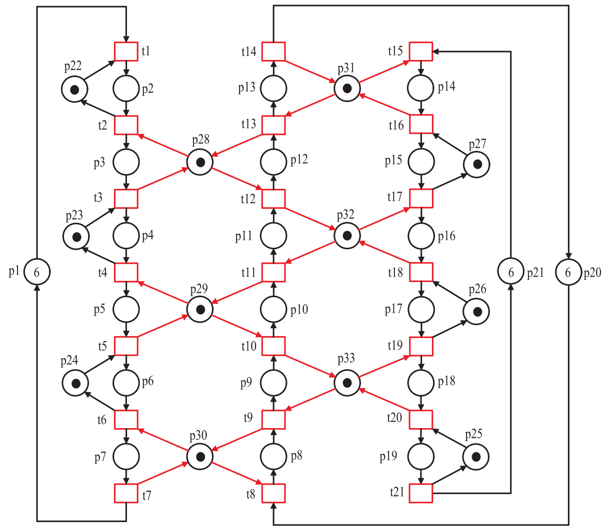

FIGURE 7. The Petri net model of an FMS with α_3-S^3PR for Example 3.

The Petri net model shown in Fig. 7 is an α_3-S^3PR structure. Firstly, we decompose the Petri net model into $(N_1^S,\ M_0^S)$ and $(N_2^S,\ M_0^S)$. Secondly, the control places are obtained via Algorithm 2. The control places of all the sub-nets are the total control places in the supervisory structure for the Petri net model shown in Fig 7.

Consider sub-net $(N_1^S,\ M_0^S)$ shown in Fig. 8(a). By Algorithm 2, the set of active uncontrolled transitions is $\Pi = \{t_2, t_8\}$ with $t_1^1 = t_2$ and $t_2^2 = t_8$. The sets of active places are $S(\hbar_1^A) = \{p_3, p_4\}$ and $S(\hbar_2^A) = \{p_8, p_9\}$. The sets of null places are $S(\hbar_1^N) = \{p_7\}$ and $S(\hbar_2^N) = \{p_{12}, p_{13}\}$. The sets of passive places in processes I and II are $\Delta^1 = \{p_5, p_6\}$ and $\Delta^2 = \{p_{10}, p_{11}\}$, respectively. The GMECs of processes I and II are, respectively, $2\lambda_1^A + \lambda_2^A + \lambda_4^2 \leq 5$ with its places associated with the GMEC being $p_3, p_4, p_8, p_9, p_{10}, p_{11}$ and $\lambda_2^A + \lambda_2^1 \leq 3$ with its places associated with the GMEC being p_5, p_6, p_8, p_9 respectively. The corresponding control places for the GMECs are respectively:

$$V_1 = \begin{bmatrix} 2\\2\\1\\1\\1\\1 \end{bmatrix}^T \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$V_1 = \begin{bmatrix} -2 & 0 & 2 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

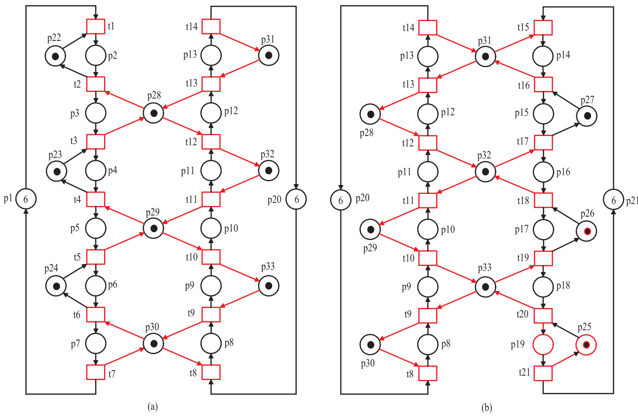

FIGURE 8. (a) The sub-net model (N_1^S, M_0^S) of Petri net model shown in Fig. 7, and (b) The sub-net model (N_2^S, M_0^S) of Petri net model shown in Fig. 7.

$$V_2 = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}^T \begin{bmatrix}1 & -1 & 0 & 0 & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 0 & 1 & -1\end{bmatrix}$$

$$V_2 = \begin{bmatrix}-1 & 0 & 1 & -1 & 0 & 1\end{bmatrix}$$

By including the two control places in the sub-net (N_1^S, M_0^S) shown in Fig. 8(a), the sub-net model is live with 1408 good states.

Now let us consider the sub-net (N_2^S, M_0^S) shown in Fig. 8(b). By Algorithm 2, the set of active uncontrolled transitions is $\Pi = \{t_9, t_{15}\}$ with $t_1^1 = t_9$ and $t_2^2 = t_{15}$. The sets of active places are $S(\hbar_1^A) = \{p_9, p_{10}\}$ and $S(\hbar_2^A) = \{p_{14}, p_{15}\}$. The sets of null places are $S(\hbar_1^N) = \{p_{13}\}$ and $S(\hbar_2^N) = \{p_{18}, p_{19}\}$. The sets of passive places are $\Delta^1 = \{p_{11}, p_{12}\}$ and $\Delta^2 = \{p_{16}, p_{17}\}$. The GMECs of processes I and II are, respectively, $2\lambda_1^A + \lambda_2^A + \lambda_4^2 \leq 5$ with its places associated with the GMEC being $p_9, p_{10}, p_{14}, p_{15}, p_{16}, p_{17}$ and $\lambda_2^A + \lambda_2^1 \leq 3$ with its places associated with the GMEC being $p_{11}, p_{12}, p_{14}, p_{15}$, respectively. The corresponding control places for the GMECs are:

$$V_1 = \begin{bmatrix}2\\2\\1\\1\\1\\1\end{bmatrix}^T \begin{bmatrix}1 & -1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1\end{bmatrix}$$

$$V_1 = \begin{bmatrix}-2 & 0 & 2 & -1 & 0 & 0 & 0 & 1\end{bmatrix}$$

$$V_2 = \begin{bmatrix}1\\1\\1\\1\end{bmatrix}^T \begin{bmatrix}1 & -1 & 0 & 0 & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 0 & 1 & -1\end{bmatrix}$$

$$V_2 = \begin{bmatrix}-1 & 0 & 1 & -1 & 0 & 1\end{bmatrix}$$

If the two control places are included in the uncontrolled sub-net model (N_2^S, M_0^S) shown in Fig. 8(b), the sub-net model is live with 1408 states. When the four computed control places of the sub-nets are included in the Petri net model shown in Fig. 7, the controlled Petri net model is live with 36192 good states. Table 8 presents the detailed parameters in the supervisory structure for the Petri net model shown in Fig. 7. Table 9 compares the performance results of the proposed method with some available policies in the literature.

TABLE 8. Summary of the supervisory structure parameters for the Petri net model shown in Fig. 7.

S/N	GMECs	V_i	$^\bullet V$	$V^\bullet$	M_0
1	$2\lambda_1^A + \lambda_2^A + \lambda_5^2 \leq 5$	V_1	$2t_4, t_{12}$	$2t_2, t_8$	5
2	$2\lambda_2^A + \lambda_3^A + \lambda_8^3 \leq 5$	V_2	$2t_{11}, t_{19}$	$2t_9, t_{15}$	5
3	$\lambda_2^A + \lambda_2^1 \leq 3$	V_3	t_6, t_{10}	t_4, t_8	3
4	$\lambda_3^A + \lambda_5^2 \leq 3$	V_4	t_{13}, t_{17}	t_{11}, t_{15}	3

TABLE 9. Performance comparison between the proposed method and some existing policies for the Petri net model shown in Fig. 7.

Policies	No. of control places	No. of arcs	No. of reachable states
[10]	10	40	39800
[36]	10	40	39800
[26]	10	40	39800
[6]	12	48	39076
[49]	12	48	39076
[proposed method]	4	16	36192

VIII. DISCUSSION

Suppose that there are n concurrent processes in (N, M_0) with $N = (P_A \cup P^0 \cup P_R, T, F)$. Obviously, n is less than the structural size of net N. We have two nested loops in Algorithm 1 to search the active uncontrolled transitions in (N, M_0). The first while-loop is executed n times such that it can visit each process in (N, M_0), while in the worst case the for-loop is executed $|P_A|$ times to find the active uncontrolled transitions (i.e., when $|I_r| = |P_A|$, where $r \in P_R$ is a resource place). Note that $|P_A| < |P|$. The overall computational complexity of Algorithm 1 (finding active uncontrolled transitions in (N, M_0)) is $\mathcal{O}(n|P|)$. However, in general, the worst case is almost unlikely to occur, since the active uncontrolled transitions can be usually found at the first iteration.

Next we analyze the computational complexity of Algorithm 2. We first deal with the problem of solving $\alpha_{l'}^A \lambda_{l'}^A + \lambda_{l''}^A + \Sigma_{i=1}^n \lambda_{l''}^i \leq \beta$. We aim to find its integer solutions for this equation. Branch-and-bound search algorithms can be used to achieve this purpose. We assume that there are n GMECs in a Petri net model (N, M_0) with $N = (P_A \cup P^0 \cup P_R, T, F)$ (Note that the number of GMECs is equal to that of concurrent processes, which is a main contribution of this research). The weighted token constant is denoted by β. In $\alpha_{l'}^A \lambda_{l'}^A + \lambda_{l''}^A + \Sigma_{i=1}^n \lambda_{l''}^i \leq \beta$, we have at most $|P_A| + 1$ variables, i.e., $\alpha_1, \alpha_2, \ldots, \alpha_{|P_A|}$, and β. The lower bound of variable α_i ($i \in \{1, 2, \ldots, |P_A|\}$) is 1 and its upper bound is $\sum_{i=1}^n \beta_i$. For variable β, it falls into the interval $[1, n\sum_{i=1}^{n}\beta_i]$. As the bounds of each variable are known, the branch-and-bound algorithm to find an integer solution to $\alpha_{l'}^{A}\lambda_{l'}^{A} + \lambda_{l''}^{A} + \Sigma_{i=1}^{n}\lambda_{l''}^{i} \leq \beta$ is of polynomial complexity [12].

Finally, we analyze the computational complexity of control place design via $V_i = -DL_D$. V_i is the product of matrices D and L_D. As aforementioned, matrix D is a row matrix with a dimension of $1 \times |P_i|$ and L_D is a matrix with the dimension of $|P_i| \times |T_i|$. The computational complexity of multiplying the two matrices is $\mathcal{O}(|P_i| \times |T_i|)$. By considering that there are n GMECs, the computational complexity of Algorithm 2 is $\mathcal{O}(n \times |P| \times |T|)$=$\mathcal{O}(n|P||T|)$. In summary, the proposed method in this paper has polynomial time complexity. Note that n is the number of concurrent processes in a system. It is less than the number of places, i.e., $n < |P|$ is definitely true.

However, a reachability graph based method is usually computationally prohibitive since a complete state enumeration is required. Such a method can be briefly presented as follows. Let $G(N, M_0)$ be the reachability graph of a Petri net model (N, M_0). $G(N, M_0)$ consists of a live zone (LZ) and a deadlock zone (DZ). Usually LZ and DZ are used to denote the sets of markings (states) in LZ and DZ, respectively. Formally, markings (states) in LZ (DZ) are expressed as $LZ = \{M|M \in R(N, M_0), \exists\sigma \in T^*, M[\sigma\rangle M_0\}$ ($DZ = \{M|M \in R(N, M_0), \nexists\sigma \in T^*, M[\sigma\rangle M_0\}$).

Let (N, M_0) be a net system with $N = (P, T, F, W)$. A marking $M \in DZ$ is said to be an FBM (first-met bad marking) in $G(N, M_0)$ if $M'[t\rangle M$, where $M' \in LZ$, $t \in T$. The set of FBMs is represented by $\mathcal{M}_{FBM} = \{M \in DZ \mid \exists M' \in LZ, t \in T, \text{ s.t. } M'[t\rangle M\}$ [10].

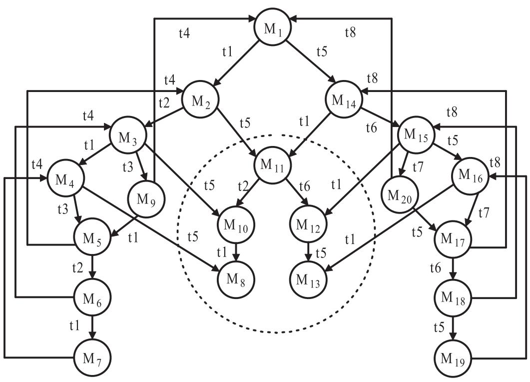

FIGURE 9. The reachability graph of a Petri net model.

Fig. 9 shows the two regions of the reachability graph, where the inner circle represents the DZ in the reachability graph while the rest of the region represents the LZ. From the reachability graph shown in Fig. 9, the set of FBMs is $\mathcal{M}_{FBM} = \{M_8, M_{10}, M_{11}, M_{12}, M_{13}\}$. The main ideas of a reachability graph based deadlock control policy of FMSs is to eliminate all the FBMs in the reachability graph. However, the number of FBMs increases exponentially with the size of the Petri net model, making the problem intractable.

IX. CONCLUSIONS

This paper presents a new method of computing a supervisory structure for a safe Petri net model with an α_n-S^{3}PR structure. The proposed method utilizes resource transition circuits in a Petri net model of a flexible manufacturing system to avoid the computation of the reachability graph (or siphons). The paper proposes two algorithms. The first one is used to compute active uncontrolled transitions from (N, M_0) and the second algorithm is used to compute the GMECs for each process of the Petri net model or sub-net model. Furthermore, a method is presented to design a control place for each computed GMEC. The proposed method is efficient as it provides minimal supervisory structure with sub-optimal (near-optimal) behavior. The major advantage derived from the proposed method is as follows. It designs a single control place at each concurrent process in the Petri net model or sub-net model, which significantly reduces the supervisory structure. Furthermore, it provides the constant number of control places in the supervisory structure that depends on the number of concurrent processes in the Petri net model or sub-net model regardless of the number of resources used. Finally, it can reduce huge costs at the stage of implementation and maintenance due to its minimal supervisory structure. The limitation of the proposed method is that it works for safe Petri net models with an α_n-S^{3}PR or α_n-S^{4}PR structure. Our future work would focus on extending the work to generalized S^{3}PR models.

REFERENCES

[1] 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. 2, pp. 413–421, Jun. 1995.

[2] C. Zhong, Z. Li, Y. Chen, and A. Al-Ahmari, "On nonexistence of a maximally permissive liveness-enforcing pure net supervisor," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 43, no. 1, pp. 29–37, Jan. 2013.

[3] J. L. Luo, W. M. Wu, H. Y. Su, and J. Chu, "Supervisor synthesis for enforcing a class of generalized mutual exclusion constraints on Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 6, pp. 1237–1246, Nov. 2009.

[4] M. Bashir, B. B. Muhammad, and Z. W. Li, "Minimal supervisory structure for flexible manufacturing system using Petri nets," in *Proc. IEEE Int. Conf. Control, Autom. Robot.*, Hongkong, Apr. 2016, pp. 291–296.

[5] M. Bashir, Z. W. Li, M. Uzam, N. Q. Wu, and A. Al-Ahmari, "On structural reduction of liveness-enforcing Petri net supervisors for flexible manufacturing systems: An algebraic approach," *IMA J. Math. Control Inf.*, 2017. [Online]. Available: https://doi.org/10.1093/imamci/dnx024

[6] Y. S. Huang, Y. L. Pan, and P. J. Su, "Transition-based deadlock detection and recovery policy for FMSs using graph technique," *ACM Trans. Embedded Computing Syst.*, vol. 12, no. 1, 2013, Art. no. 11, doi: 10.1145/2406336.2406347.

[7] 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, 2010.

[8] 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.

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

[10] Y. F. Chen, Z. W. 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.

[11] 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.

[12] W. Zhang. (2015). *Branch and Bound Search Algorithms and Their Computational Complexity*. Dept. Inf. Sci. Inst., Univ. Southern California, Los Angeles, CA, USA, Tech. Rep. ISI/RR-96-443, 1996. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a314598.pdf

[13] J. F. Zhang, M. Khalgui, Z. W. Li, G. Frey, O. Mosbahi, and H. B. Salah, "Reconfigurable coordination of distributed discrete event control systems," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 323–330, Jan. 2015.

[14] Y. F. Chen and Z. W. Li, "Design of a maximally permissive Liveness enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 47, no. 5, pp. 1028–1038, 2011.

[15] J. Ye, Z. Li, and A. Giua, "Decentralized supervision of Petri nets with a coordinator," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 6, pp. 955–966, Jun. 2015.

[16] Z. Ma, Z. Li, and A. Giua, "Characterization of admissible marking sets in Petri nets with conflicts and synchronizations," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1329–1341, Mar. 2017.

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

[18] E. G. Coffman, M. J. Elphick, and A. Shoshani, "Systems deadlocks," *ACM Comput. Surv.*, vol. 3, no. 2, pp. 66–78, 1971.

[19] J. Ezepeleta, J. M. Colom, and J. Martinez, "A Petri nets based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995.

[20] 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.

[21] A. Ghaffari, N. Reng, and X. L. 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. 137–142, Feb. 2003.

[22] Z. D. Han and G. Lee, "Application of Petri nets for deadlock analysis and avoidance in flexible manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 25, no. 7, pp. 735–742, 2005.

[23] Z. W. Li and M. C. Zhou, "A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 182–188, Jan. 2008.

[24] Z. W. Li and M. C. Zhou, "Control of elementary and dependent siphons in Petri nets and their application," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 133–148, Jan. 2008.

[25] 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.

[26] L. Piroddi, R. Cordone, and I. Fumagalli, "Combined siphon and marking generation for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 3, pp. 650–661, May 2009.

[27] 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.

[28] D. Liu, Z. Li, and M. Zhou, "Hybrid liveness-enforcing policy for generalized Petri net models of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 1, pp. 85–97, Jan. 2013.

[29] A. R. Wang, Z. W. Li, and M. C. Zhou, "An effective algorithm to find elementary siphons in a class of Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 4, pp. 912–922, Jul. 2009.

[30] S. A. Reviolitis, "Implicit siphon control and its role in the liveness-enforcing supervision of sequential resource allocation systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 37, no. 3, pp. 319–324, May 2007.

[31] M. Uzam, Z. W. Li, and M. C. Zhou, "Identification and elimination of redundant control places in Petri net based liveness enforcing supervisors of FMS," *Int. J. Adv. Manuf. Technol.*, vol. 35, no. 1, pp. 150–168, Nov. 2007.

[32] F. J. Yang, N. Q. Wu, Y. Qiao, M. C. Zhou, and Z. W. Li, "Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 502–516, Mar. 2017.

[33] Y. Tong, Z. W. Li, C. Seatzu, and A. Giua, "Verification of state-based opacity using Petri nets," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2823–2837, Jun. 2017.

[34] M. Uzam, Z. W. 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.

[35] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri nets models with resource and the theory of regions," *Int. J. Adv. Manuf. Technol.*, vol. 19, no. 3, pp. 193–208, 2002.

[36] M. Uzam and M. C. Zhou, "An improved iterative synthesis method for liveness-enforcing supervisor of flexible manufacturing system," *Int. J. Prod. Res.*, vol. 44, no. 10, pp. 1987–2030, 2006.

[37] M. Uzam and M. C. 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. 35, no. 6, pp. 1193–1202, May 2007.

[38] H. C. Liu, J. X. You, Z. W. Li, and G. D. Tian, "Fuzzy Petri nets for knowledge representation and reasoning: A literature review," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 45–56, Apr. 2017.

[39] Y. F. Chen and Z. W. Li, "On structural minimality of optimal supervisors for flexible systems," *Automatica*, vol. 48, no. 10, pp. 2647–2656, 2012.

[40] Z. W. Li, H. S. Hu, and A. R. Wang, "Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 4, pp. 517–526, Jul. 2007.

[41] Y. S. Huang and Y. L. Pan, "An improved maximally permissive deadlock prevention policy based on the theory of regions and reduction approach," *IET Control Theory Appl*, vol. 5, no. 9, pp. 1069–1078, 2011.

[42] Z. W. Li, N. Q. Wu, and M. C. Zhou, "Deadlock control of automated manufacturing system based on Petri nets—A literature review," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 437–460, Jul. 2012.

[43] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Short-term scheduling of crude-oil operations: Petri net-based control-theoretic approach," *IEEE Robot. Autom. Mag.*, vol. 22, no. 2, pp. 64–76, Jun. 2015.

[44] Y. F. Chen, Z. W. Li, K. Barkaoui, and A. Giua, "On the enforcement of a class of nonlinear constraints on Petri nets," *Automatica*, vol. 55, pp. 116–124, May 2015.

[45] Y. Hou, N. Q. Wu, M. C. Zhou, and Z. W. Li, "Pareto-optimization for scheduling of crude oil operations in refinery via genetic algorithm," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 517–530, Mar. 2017.

[46] Y. Chen, Z. Li, K. Barkaoui, and M. Uzam, "New Petri net structure and its application to optimal supervisory control: Interval inhibitor arcs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 10, pp. 1384–1400, Oct. 2014.

[47] Y. F. Chen, Z. W. Li, K. Barkaoui, N. Q. Wu, and M. C. Zhou, "Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 364–379, Feb. 2017.

[48] Y. Tong, Z. W. Li, and A. Giua, "On the equivalence of observation structures for Petri net generators," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2448–2462, Sep. 2016.

[49] M. Uzam, Z. W. Li, and U. S. Abubakara, "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.

[50] J. O. Moody and P. J. Antsaklis, "Petri net supervisors for DES with uncontrollable and unobservable transitions," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 462–476, Mar. 2000.

[51] X. Wang, Z. W. Li, and W. M. Wonham, "Dynamic multiple-period reconfiguration of real-time scheduling based on timed DES supervisory control," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 101–111, Feb. 2016.

[52] L. P. Bai, N. Q. Wu, Z. W. Li, and M. C. Zhou, "Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 10, pp. 1456–1467, Oct. 2016.

[53] K. Yamalidou, J. Moody, M. Lemmon, and P. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, vol. 32, no. 1, pp. 15–28, 1996.

[54] N. Q. Wu, M. C. Zhou, L. P. Bai, and Z. W. Li, "Short-term scheduling of crude oil operations in refinery with high-fusion-point oil and two transportation pipelines," *Enterprise Inf. Syst.*, vol. 10, no. 6, pp. 581–610, 2016.

[55] Y. F. Chen, Z. W. Li, A. Al-Ahmari, N. Q. Wu, and T. Qu, "Deadlock recovery for flexible manufacturing systems modeled with Petri nets," *Inf. Sci.*, vol. 381, pp. 290–303, Mar. 2017.

[56] S. G. Wang, D. You, and M. C. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S^4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017, doi: 10.1109/TAC.2017.2677859.
[57] S. G. Wang, D. You, M. C. Zhou, and C. Seatsu, "Characterization of admissible marking sets in Petri nets with uncontrollable transitions," *IEEE Trans. Autom. Control*, vol. 61, no. 7, pp. 1953–1958, Jul. 2016.
[58] S. G. Wang, D. You, and C. Seatzu, "A novel approach for constraint transformation in Petri nets with uncontrollable transitions," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2665479.
[59] S. G. Wang, D. You, and C. Y. Wang, "Optimal supervisor synthesis for Petri nets with uncontrollable transitions: A bottom-up algorithm," *Inf. Sci.*, vol. 363, pp. 261–273, Oct. 2016.
[60] S. G. Wang, C. Y. Wang, and M. C. 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.
[61] S. Wang, C. Wang, and M. Zhou, "Design of optimal monitor-based supervisors for a class of Petri nets with uncontrollable transitions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 5, pp. 1248–1255, Sep. 2013.
[62] D. You, S. G. Wang, and M. C. Zhou, "Synthesis of monitor-based liveness-enforcing supervisors for S^3PR with ξ-resources," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 6, pp. 967–975, Jun. 2015.
[63] K. Barkaoui, A. Chaoui, and B. Zouari, "Supervisory control of discrete event systems based on structure theory of Petri nets," in *Proc. IEEE Conf. Syst., Man, Cybern.*, Paris, France, Oct. 1997, pp. 3750–3755.
[64] F. Chu and X.-L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. Robot. Autom.*, vol. 13, no. 6, pp. 793–804, Dec. 1997.
[65] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable transitions," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Chicago, IL, USA, Oct. 1992, pp. 974–979.
[66] Z. W. Li, G. Y. Liu, M. H. Hanisch, and M. C. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 1, pp. 178–191, Jan. 2012.
[67] K. Xing, M. C. Zhou, H. Liu, and F. Tian, "Optimal Petri-net-based polynomial-complexity deadlock-avoidance policies for automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 39, no. 1, pp. 188–199, Jan. 2009.
[68] T. Tricas, F. García-Vallés, J. M. Colom, and J. Eapeleta, "A Petri net structure-based deadlock prevention solution for sequential resource allocation systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 271–277.
[69] K. Barkaoui and M. Minoux, "A polynomial-time graph algorithm to decide liveness of some basic classes of bounded Petri nets," in *Proc. Int. Conf. Appl. Theory Petri Nets*, 1992, pp. 62–75.
[70] E. Best, "Structure theory of Petri nets: The free choice hiatus," in *Petri Nets: Central Models and Their Properties*. Berlin, Germany: Springer, pp. 168–205, 1986.
[71] E. Best and S. P. Thiagarajan, "Some classes of live and safe Petri nets," in *Concurrency Nets*. Berlin, Germany: Springer, 1987, pp. 71–94.
[72] F. Dicesare, G. Harhalakis, J. M. Proth, M. S. Suarez, and F. Vernadat, *Practice of Petri Nets in Manufacturing*. London, U.K.: Chapman & Hall, 1993.
[73] M. H. T. Hack, "Analysis of production schemata by Petri nets," Dept. Elect. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. MAC-TR-94, 1972.
[74] H. Anatol and F. Commoner, "Events and conditions: Introduction," in *Proc. Rec. Project MAC Conf. Concurrent Syst. Parallel Comput.*, 1970, pp. 3–5.
[75] A. Kovalyov and J. Esparza, "A polynomial algorithm to compute the concurrency relation of free–choice signal transition graphs," in *Proc. Int. Workshop Discrete Event Syst. (WODES)*, Edinburgh, Scotland, U.K., Aug. 1996, pp. 1–6.
[76] C. Rovetto, T. J. Concepción, and E. E. Cano, "Virtual control policy for binary ordered resources Petri net class," *Sensors*, vol. 16, no. 8, p. 1307, 2016, doi: 10.3390/s16081307.
[77] G. J. Liu, W. Reisig, C. J. Jiang, and M. C. Zhou, "A Branching-process-based method to check soundness of workflow systems," *IEEE Access*, vol. 4, pp. 4104–4118, 2016, doi: 10.1109/ACCESS.2016.2597061).
[78] G. Liu, "Complexity of the deadlock problem for Petri nets modeling resource allocation systems," *Inf. Sci.*, vol. 363, pp. 190–197, Oct. 2016.

MUHAMMAD BASHIR received the B.Eng. degree in electrical engineering from Bayero University Kano, Nigeria, in 2010, and the master's degree in electrical and computer engineering from Meliksah University Kayseri, Turkey, in 2014. He is currently pursuing the Ph.D. degree in control theory and control engineering with the School of Electro-Mechanical Engineering, Xidian University, Xi'an, China. His current research interests include supervisory control of discrete event system, Petri net applications, system modeling, and PLC programming.

ZHIWU LI (M'06–SM'07–F'16) received the B.S., M.S., and Ph.D. degrees from Xidian University in 1989, 1992, and 1995, respectively. He joined Xidian University in 1992. He is currently a Professor with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau. He published two monographs in Springer and CRC Press and over 70 papers in *Automatica* and IEEE Transactions (mostly regular). His research interests include discrete event systems and Pertri nets. His work was cited by engineers and researchers from more than 40 countries and areas, including prestigious research and development institutes, such as IBM, Volvo, HP, GE, GM, and ABB. He was a recipient of a Alexander von Humboldt Research Grant and Research in Paris. He was selected in 2014, 2015, and 2016 Thomson Reuters Highly Cited Researchers in the category of Engineering. He serves (served) an Associate Editor of the IEEE Transactions on Automation Science and Engineering, the IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Human Beings, the IEEE Transactions on Systems, Man, and Cybernetics: Systems, and *Information Sciences* (Elsevier).

MURAT UZAM was born in Söke, Turkey, in 1968. He 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. He was with Nigde University, Turkey, from 1993 to 2010. He was a Visiting Researcher with INRIA, the University of Metz, and the University of Rennes, France, in 1999; the University of Toronto, Toronto, ON, Canada, in 2003; and Xidian University, Xi'an, China, in 2013 and 2015, respectively. He was a Professor with the Department of Electrical and Electronics Engineering, Melikşah University, Kayseri, Turkey, from 2011 to 2016. He has published 47 conference papers and 84 journal and magazine papers, 60 of which are indexed by Science Citation Index. He has two books published in Turkish and one book published in English by CRC Press (Taylor & Francis Group). In total, his papers have been cited over 700 times by the papers indexed in the SCI. According to Web of Science, his H-Index is 13. His current research interests include design and implementation of discrete event control systems modeled 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. He has been serving as a reviewer for prestigious journals and conferences.

ABDULRAHMAN AL-AHMARI received the Ph.D. degree in manufacturing systems engineering from The University of Sheffield in 1998. He is currently the Dean of the Advanced Manufacturing Institute, a Supervisor of the Raytheon Chair for Systems Engineering, and a Professor of industrial engineering with King Saud University, Riyadh, Saudi Arabia. He has published papers in leading journals of industrial and manufacturing engineering. He led a number of funded projects from different organizations in Saudi Arabia. His research interests are advanced manufacturing technologies, Petri nets, analysis and design of manufacturing systems, computer integrated manufacturing, optimization of manufacturing operations, FMS and cellular manufacturing systems, and applications of DSS in manufacturing.

NAIQI WU (M'04–SM'05) received the B.S. degree in electrical engineering from the Anhui University of Technology, Huainan, China, in 1982, and the M.S. and Ph.D. degrees in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 1985 and 1988, respectively. From1988 to 1995, he was with the Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, and Shantou University, Shantou, China, from 1995 to 1998. He moved to the Guangdong University of Technology, Guangzhou, China, in 1998. He joined the Macau University of Science and Technology, Taipa, Macau, in 2013, where he is currently a Professor with the Institute of Systems Engineering. He has authored or co-authored one book, five book chapters, and over 130 peer-reviewed journal papers. His research interests include production planning and scheduling, manufacturing system modeling and control, discrete event systems, Petri net theory and applications, intelligent transportation systems, and energy systems. He was an Associate Editor of the IEEE Transactions on Systems, Man, and Cybernetics—Part C, the IEEE Transactions on Automation Science and Engineering, and the IEEE Transactions on Systems, Man, and Cybernetics: Systems, and an Editor-in-Chief of the *Industrial Engineering Journal*, and is an Associate Editor of *Information Sciences* and the IEEE/CAA Journal of Automatica Sinica.

DING LIU (M'08) received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 2003, 2006, and 2012, respectively. He is currently an Associate Professor with the School of Electro-Mechanical Engineering and the Key Laboratory of Electronic Equipment Structure Design, Xidian University. Since 2005, he has published 21 papers, including international journal and conference ones, which all focus on the deadlock prevention policies for automated manufacturing systems modeled with Petri nets. His research interests include Petri nets, discrete event systems, and automata. He served as the Chair of organizing committee of WODES 2016 and the member of international program committees of multiple versions of the IEEE CASE and the IEEE ICNSC.

TING QU received the B.S. and M.S. degrees in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2001 and 2004, respectively, and the Ph.D. degree in industrial and manufacturing systems engineering from The University of Hong Kong, Hong Kong, in 2008. From 2008 to 2010, he was a Post-Doctoral Research Fellow and then a Research Assistant Professor with the Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong. He joined the Guangdong University of Technology, Guangzhou, China, in 2010. He moved to Jinan University, Zhuhai Campus, Zhuhai, China, in 2016, where he is currently a Professor with the School of Electrical and Information Engineering. He has published over 100 technical papers with more than half of them in reputable journals. His research interests include IoT-based smart manufacturing systems, logistics and supply chain management, and industrial product/production service systems. He serves as an Associate Editor (Asia) of the *International Journal of Computer Integrated Manufacturing* and the Editorial Board Member of the *Industrial Engineering Journal*.