Lexicographic Multiobjective Integer Programming for Optimal and Structurally Minimal Petri Net Supervisors of Automated Manufacturing Systems

Bo Huang, *Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, GongXuan Zhang, *Senior Member, IEEE*, Ahmed Chiheb Ammari, *Member, IEEE*, Ahmed Alabdulwahab, *Senior Member, IEEE*, and Ayman G. Fayoumi, *Senior Member, IEEE*

***Abstract*—Based on Petri net (PN) models of automated manufacturing systems, this paper proposes a deadlock prevention method to obtain a maximally permissive (optimal) supervisor while minimizing its structure. The optimal supervisor can be achieved by forbidding all first-met bad markings (FBMs) and permitting all legal markings in a PN model. An FBM obtained via a single transition's firing at a legal marking is a deadlock or marking that inevitably evolves into a deadlock. A lexicographic multiobjective integer programming problem with multiple objectives to be achieved sequentially is formulated to design such an optimal and structurally minimal supervisor. As a nonlinear function, the quantity of its directed arcs is minimized. A conversion method is proposed to convert the nonlinear model into a linear one. With the premise that each place in the supervisor is associated with a nonnegative place invariant, the controlled net holds all legal markings of the net model, and the supervisor has the minimal structure. Finally, some examples are used to illustrate the application of the proposed approach.**

***Index Terms*—Automated manufacturing system (AMS), deadlock prevention, discrete event system, linear programming problem, Petri net (PN).**

Manuscript received November 14, 2014; accepted January 9, 2015. Date of publication April 21, 2015; date of current version October 13, 2015. This work was supported in part by the Deanship of Scientific Research, King Abdulaziz University, Jeddah, Saudi Arabia, under Grant 23-135-35-HiCi, in part by the National Natural Science Foundation of China under Grant 61203173 and Grant 61272420, in part by the Zijin Intelligent Program of the Nanjing University of Science and Technology under Grant AB41379, in part by the Foundation of the Key Laboratory of Advanced Process Control for Light Industry (Jiangnan University) of the Ministry of Education of China under Grant APCLI1403. This paper was recommended by Associate Editor M. K. Tiwari.

B. Huang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China, and also with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA.

M. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA, and also with the Renewable Energy Research Group, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: zhou@njit.edu).

G. Zhang is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China.

A. C. Ammari is with the Renewable Energy Research Group, Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia, and also with the MMA Laboratory, INSAT Institute, Carthage University, Tunis 1080, Tunisia.

A. Alabdulwahab is with the Renewable Energy Research Group, Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia.

A. G. Fayoumi is with the Department of Information Systems, King Abdulaziz University, Jeddah 21589, Saudi Arabia.

Digital Object Identifier 10.1109/TSMC.2015.2415765

I. Introduction

In an industrial automated manufacturing system (AMS), different types of jobs are handled concurrently, sharing a limited number of resources such as numerically controlled machines, robots, buffers, sensors, and inspection stations. In such resource-sharing systems, the competition for shared resources by different jobs may cause deadlocks, which may eventually stall all activities in the systems. Hence, deadlock avoidance and prevention are critically important to ensure the highest performance of AMS.

There are mainly three tools used to deal with deadlocks of AMS: 1) graph theory [1], [2]; 2) automata [3], [4]; and 3) Petri nets (PNs) [5]–[8]. Among these tools, PNs are suitable to model and analyze the behavior of AMS [9]–[13] and address the deadlock issues [14]–[21]. Generally, there are three deadlock resolution approaches: 1) deadlock detection and recovery [22]–[24]; 2) deadlock avoidance [25]–[30]; and 3) deadlock prevention [2], [14], [31]–[35]. Our research [11], [36]–[38], which generates some deadlock-free schedules in advance and applies them to AMSs to optimize the measure of performance under consideration, belongs to the third approach, i.e., deadlock prevention. Deadlock prevention requires an off-line computation mechanism to add constraints to a system for preventing deadlock states from being reached and does not need on-line detection, recovery, and avoidance procedures by taking a full view of the system. This paper focuses on it.

Two kinds of analysis methods are mainly used to analyze PNs: 1) structural analysis [17], [23], [31], [39]–[43] and 2) reachability graph analysis [44]–[46]. The former allows one to derive a control policy by special PN structures, e.g., resource-transition circuits and siphons. The resulting control law is usually simple, but not optimal in general. The latter can lead to a controlled model with the maximally permissive or highly permissive behavior. However, its computational burden is always heavy. Maximal permissiveness implies that all legal markings are kept when a supervisor is added to the plant model. The existing supervisor synthesis methods often construct a PN supervisor (controller) that consists of places and arcs based on the concept of place invariants (PIs) to prevent deadlocks. The places of the obtained supervisor are called control places or monitors.

Three criteria commonly used to evaluate a supervisor in a controlled net are: 1) behavioral permissiveness; 2) structural complexity; and 3) computational complexity. A supervisor with maximal permissiveness is called an optimal one that often implies high utilization of system resources. A structurally minimal supervisor can reduce the control implementation cost. A supervisor design algorithm with low computational complexity can deal with complex AMS.

The theory of region proposed in [47] can be used to derive an optimal supervisor for a PN model if such a supervisor exists. To improve its computational efficiency, Uzam and Zhou [45] developed an iterative approach for the deadlock control of AMS. They divide a reachability graph into a live zone (LZ) and deadlock zone (DZ). A marking in LZ is called a legal one that can reach the initial marking, and that in DZ is a deadlock or will inevitably lead to deadlocks. First-met bad markings (FBMs) are those in DZ that are immediately reachable from some in LZ. At each iteration, an FBM is selected from the reachability graph. To prevent this FBM from being reached, a control place is derived by constructing a PI of the PN by using an invariant-based control method [48]. Then, this control place and its related arcs are added to the PN. This process is iteratively carried out till the controlled PN becomes live. Although the method in [45] is easy and straightforward, it cannot guarantee the optimality in general.

Chen *et al*. [46] proposed a reachability graph-based method that can definitely obtain optimal liveness-enforcing supervisors for AMSs modeled by PN if such supervisors exist. An optimal control place of the supervisor is designed by a PI at each iteration, which is achieved by solving an integer linear program (ILP) to forbid as many FBMs as possible and permit all legal markings. To address the computational complexity problem, a vector covering method is used to reduce the numbers of considered markings in the sets of FBMs and legal markings. However, their method has the structural complexity problem since it cannot ensure the fewest control places. Hence, Chen and Li [49] developed a method that can design an optimal supervisor with the fewest control places.

In fact, all the above studies do not consider the cost of arcs added to control places in the stage of control implementation. In the real world, the implementation cost of a supervisor can be evaluated by two components: 1) control places and 2) added arcs. A control place represents a processing unit such as a programmable logic controller, microcontroller, and computer. It needs data collected from a process and signal transmission via transducers such as sensors and actuators that are modeled by the directed arcs between a plant net and its supervisor. Cordone and Piroddi [50] described a branch-and-bound algorithm to design an optimal supervisor such that one of the cost functions (e.g., generalized mutual exclusion constraint coefficients and the weights of the arcs in a supervisor) is minimized. However, they do not consider the simultaneous minimization of several cost functions. The method proposed in [51] considers the implementation cost of control places and added arcs by solving an ILP. It can obtain a small number of control places, but fails to ensure the minimality in general.

In this paper, an ILP is proposed, aiming to minimize the structural complexity of the supervisor with respect to both control places and added arcs while the controlled system is still live and optimal. First, the vector covering approach is used to reduce the numbers of legal markings and FBMs to be considered. Then, a lexicographic multiobjective ILP (LMILP) is formulated and solved. This paper has made the following contributions.

1) An LMILP is formulated to design an optimal supervisor for the PN model of an AMS. All control places and associated arcs in it can then be computed.
2) If there exists an optimal supervisor for the plant, the one with the fewest arcs while ensuring the fewest control places can definitely be found.
3) A linear conversion method that converts our nonlinear model into a linear one is proposed to facilitate the solution of the original nonlinear program.

This method can be used to design optimal supervisors for all classes of PNs to model AMSs if such supervisors exist, such as PPN [52], S^3PR [14], ES^3PR [53], S^4PR [54], S^*PR [55], S^2LSPR [56], S^3PGR^2 [57], and S^3PMR [40].

Section II reviews the preliminaries of PNs. The supervisor computation by a PI in [48] and the method to synthesize an optimal supervisor in [46] are briefly recalled in Section III. Section IV presents a method to design an optimal supervisor with the fewest control places and associated arcs. A deadlock prevention method is given in Section V. Section VI provides some examples and compares our method with the existing ones. Finally, the conclusion is given in Section VII.

II. Preliminaries

A. Petri Nets [6], [7], [9]

A PN is defined as $N = (P, T, F, W)$ where $P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places and $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is the flow relation, depicted by arcs with arrows between places and transitions. $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{Z}^+$ (the set of nonnegative integers) is a mapping from arcs to weights. $M : P \rightarrow \mathbb{Z}^+$ is a marking assigned to each place $p \in P$, $M(p)$ tokens. (N, M_0) is called a net system. A net is said to be pure if $\forall p_i \in P$, $\forall t_j \in T$, $W(p_i, t_j) \cdot W(t_j, p_i) = 0$. The input incidence matrix is $[N^-] = \{W(p_i, t_j)\}$, and the output one is $[N^+] = \{W(t_j, p_i)\}$. The incidence matrix of a pure net is $[N] = [N^+] - [N^-]$.

Given $x \in P \cup T$, its preset ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$ and postset $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$. Given a marking M, a transition $t \in T$ is enabled if $\forall p \in^\bullet t, M(p) \geq W(p, t)$. Firing an enabled transition t generates a marking M' satisfying $M'(p) = M(p) - W(p, t) + W(t, p)$. A marking M is reachable from M_0 if there is a sequence of transition firings from M_0 to M. $R(N, M_0)$ is the set of all markings reachable

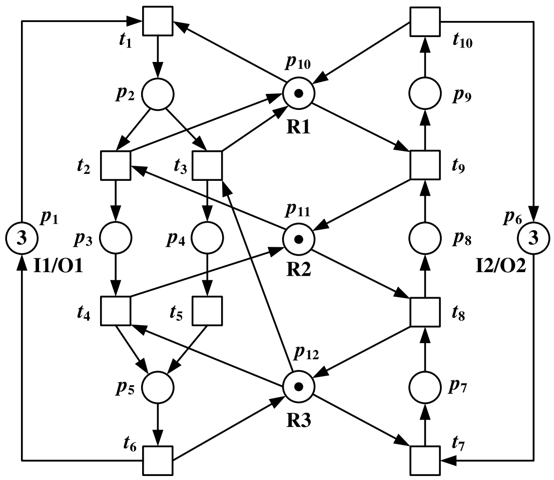


Fig. 1. PN model of an AMS.

from M_0. A reachability graph $G(N, M_0)$ is a directed graph whose nodes are markings in $R(N, M_0)$ and arcs are labeled by transitions of N.

A transition $t \in T$ is live under M_0 if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, t is enabled under M'. A PN is said to be live under M_0 if $\forall t \in T$, t is live under M_0. A PN is deadlock-free, if at least one transition is enabled at every reachable marking. A place vector is a vector $I : P \to \mathbb{Z}$. Place vector I is said to be a PI if $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$. Let I be a PI of (N, M_0), then $I^T M = I^T M_0, \forall M \in R(N, M_0)$. A PI is called a P-semiflow if its elements are all nonnegative.

B. Reachability Graph Analysis

$G(N, M_0)$ consists of two parts: 1) LZ and 2) DZ. The markings in LZ are legal ones that can reach initial marking M_0 from at least one of their successors. The set of legal markings is

$$\mathcal{M}_L = \{M | M \in R(N, M_0) \wedge M_0 \in R(N, M)\}. \tag{1}$$

DZ consists of deadlocks and the markings that cannot reach M_0 and inevitably lead to deadlocks. An FBM is a marking in DZ, which is the very first one from LZ to DZ [45]. The set of all FBMs is

$$\mathcal{M}_F = \{M \in \text{DZ} | \exists M' \in \text{LZ}, t \in T, \text{s.t. } M'[t\rangle M\}. \tag{2}$$

We consider an example AMS with three shared resources R1-3 (such as machines and robots), two loading buffers I1-2, and two unloading buffers O1-2. A resource processes only one part at a time. Two types of parts, i.e., J1 and J2, are processed in the system. The production sequences are

J1: I1 → R1 → R2 (or R3) → R3 → O1
J2: I2 → R3 → R2 → R1 → O2.

Note that J1 is processed twice by R3 in one of its production sequences. Fig. 1 shows its PN model that has 12 places and ten transitions. Its reachability graph is given in Fig. 2 where a compact multiset formalism $\sum_i M(p_i)p_i$ is used to denote a marking M for conciseness. For this reachability graph, $M_9 = p_1 + p_2 + p_3 + 2p_6 + p_7$, $M_{11} = 2p_1 + p_3 + 2p_6 + p_7 + p_{10}$, and $M_{22} = 2p_1 + p_2 + p_6 + p_7 + p_8$ are FBMs and all others are legal markings. If all FMBs are forbidden by a supervisor, the resulting controlled net can keep running in LZ, but never enters DZ. Hence, it is live.

III. Supervisor Computation

A. Supervisor Computation by PI

The following definitions about supervisor computation are primarily due to [48] and [49]. Let $[N_p]$ be the incidence matrix of the plant PN. $[N_c]$ represents the flow relation between the control places in the supervisor and transitions in the plant net. $[N]$ is the incidence matrix of the controlled net consisting of a plant net and its supervisor, that is

$$[N] = \begin{bmatrix} N_p \\ N_c \end{bmatrix}. \tag{3}$$

The goal of designing a supervisor is to enforce the controlled net to satisfy the constraints as follows:

$$[L] \cdot \mu_p \leq b \tag{4}$$

where $[L]$ is an $n_c \times n$ nonnegative integer matrix, μ_p represents the marking vector of the PN model, b is an $n_c \times 1$ vector, n_c is the number of control places in the supervisor, and n is the number of places in the plant PN. By adding an $n_c \times 1$ vector μ_c to the left side of (4), we have

$$[L] \cdot \mu_p + \mu_c = b \tag{5}$$

where μ_c is the marking of the control places in the supervisor.

The supervisor $[N_c]$ can be computed as follows:

$$[N_c] = -[L] \cdot [N_p] \tag{6}$$

where $[N_c](i,j) > 0$ indicates that an arc should be added from t_j to control place p_{c_i} with $W(t_j, p_{c_i}) = [N_c](i,j)$, while $[N_c](i,j) < 0$ implies an arc from p_{c_i} to t_j with $W(p_{c_i}, t_j) = -[N_c](i,j)$.

Let μ_{c_0} be the initial marking of the controlled net, which must also satisfy (5). Thus, μ_{c_0} is computed as follows:

$$\mu_{c_0} = b - [L] \cdot \mu_{p_0}. \tag{7}$$

B. Optimal Supervisor Synthesis

Places in the PN model of an AMS can be classified into three types: 1) idle; 2) resource; and 3) activity places [5], [14], [57]. Tokens of an idle place represent the maximal number of concurrent operations that can happen in a production sequence. Resource places model production resources, e.g., robots and machines, and their initial tokens represent available resource units. An activity place represents an operation to be performed for a part in a production sequence, and it has no token in the initial marking. For example, the places in the PN in Fig. 1 are divided into idle places $\{p_1, p_6\}$, resource places $\{p_{10} - p_{12}\}$, and activity places $\{p_2 - p_5, p_7 - p_9\}$. For the PN model of an AMS, only the tokens in activity places are considered to construct a PI for designing supervisors [58]. In the following, P_A is used to

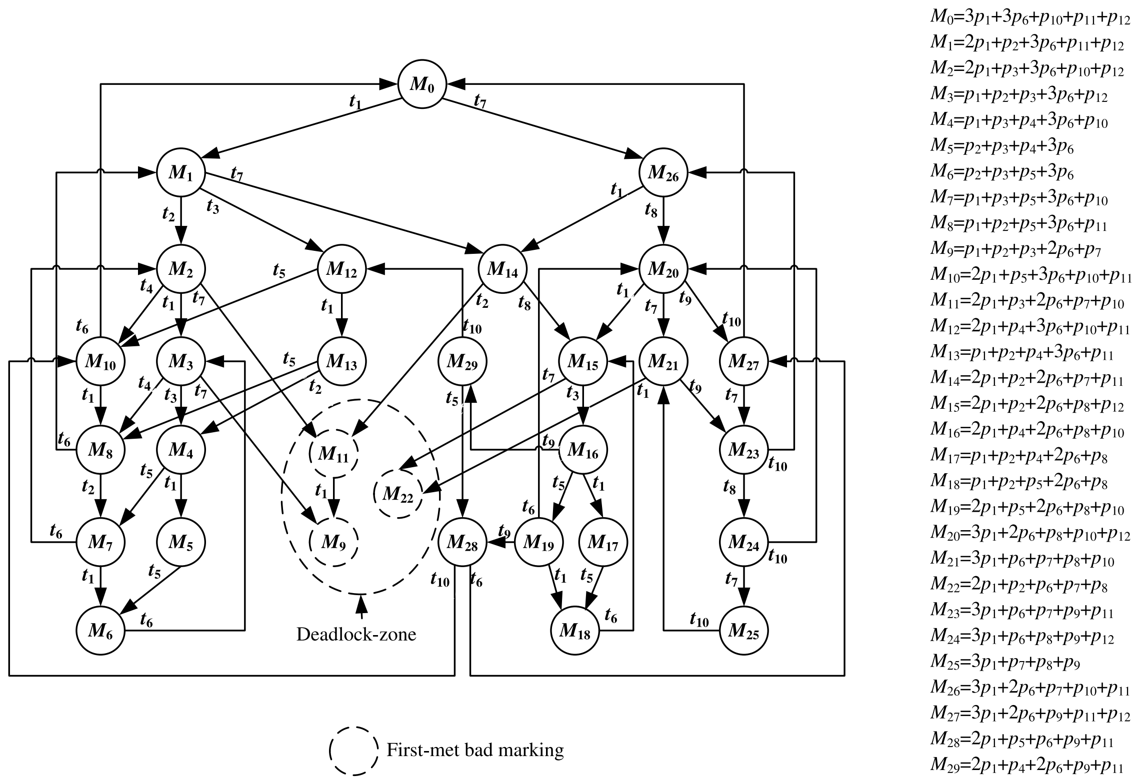


Fig. 2. Reachability graph of the PN in Fig. 1.

denote the set of activity places and $\mathbb{N}_A$ represents $\{i|p_i \in P_A\}$ for expedience.

In general, an optimal supervisor can be obtained by designing a supervisor that forbids all FBMs and permits all legal markings [46]. An FBM $M \in \mathcal{M}_F$ can be forbidden if

$$\sum_{i\in\mathbb{N}_A} l_i \cdot \mu_i \le \beta \tag{8}$$

where

$$\beta = \sum_{i\in\mathbb{N}_A} l_i \cdot M(p_i) - 1 \tag{9}$$

and $l_i (i \in \mathbb{N}_A)$ is a coefficient of a PI.

Any legal marking $M' \in \mathcal{M}_L$ is guaranteed to be reachable in the controlled net by the constraints as follows:

$$\sum_{i\in\mathbb{N}_A} l_i \cdot M'(p_i) \le \beta, \ \forall M' \in \mathcal{M}_L. \tag{10}$$

For an FBM, by substituting (9) into (10), the constraints for permitting all legal markings become

$$\sum_{i\in\mathbb{N}_A} l_i \cdot (M'(p_i) - M(p_i)) \le -1, \ \forall M' \in \mathcal{M}_L. \tag{11}$$

Constraints (11) determine $l_i (i \in \mathbb{N}_A)$ of a PI. If l_i cannot be obtained, then FBM M cannot be separated from the set of legal markings by the resulting supervisor. Thus, for an FBM M, if each $l_i (i \in \mathbb{N}_A)$ of a PI satisfies (11), then the PI designed with (8) forbids M and permits all legal markings. In this case, the obtained control place is optimal since all legal markings of the plant net are reachable in the controlled net.

However, the numbers of legal markings and FBMs in $G(N, M_0)$ are often large since the size of $G(N, M_0)$ grows exponentially with the PN size. Chen *et al*. [46] proposed a vector covering method to reduce the number of markings to be considered in the process of supervisor synthesis.

Definition 1: $\forall M, M' \in R(N, M_0)$, $M \ge_A M'$ if $\forall p \in P_A$, $M(p) \ge M'(p)$.

If $M \ge_A M'$ and M' is forbidden by a supervisor obtained by (8) and (9), we need not consider M since it is definitely forbidden in (8) and (9).

Definition 2: $\mathcal{M}_F^*$ is a minimal covered set of $\mathcal{M}_F$ if

1) $\mathcal{M}_F^* \subseteq \mathcal{M}_F$;
2) $\forall M \in \mathcal{M}_F$, $\exists M' \in \mathcal{M}_F^*$ such that $M \ge_A M'$;
3) $\forall M \in \mathcal{M}_F^*$, $\nexists M'' \in \mathcal{M}_F^*$ such that $M \ge_A M''$ and $M \ne M''$.

Definition 3: $\mathcal{M}_L^*$ is a minimal covering set of $\mathcal{M}_L$ if

1) $\mathcal{M}_L^* \subseteq \mathcal{M}_L$;
2) $\forall M \in \mathcal{M}_L$, $\exists M' \in \mathcal{M}_L^*$ subject to $M' \ge_A M$;
3) $\forall M \in \mathcal{M}_L^*$, $\nexists M'' \in \mathcal{M}_L^*$ subject to $M'' \ge_A M$ and $M \ne M''$.

If a supervisor forbids all markings in $\mathcal{M}_F^*$, all FBMs in $\mathcal{M}_F$ are forbidden. If it permits all markings in $\mathcal{M}_L^*$, all legal markings in $\mathcal{M}_L$ are permitted. Thus, only two reduced sets, $\mathcal{M}_F^*$ and $\mathcal{M}_L^*$, should be considered to design an optimal supervisor to forbid all FMBs and permit all legal markings of the plant net. Then, for an FBM M, the constraints for permitting all legal markings are reduced into

$$\sum_{i\in\mathbb{N}_A} l_i \cdot \left(M'(p_i) - M(p_i)\right) \le -1, \ \forall M' \in \mathcal{M}_L^*. \tag{12}$$

IV. Structurally Minimal Supervisor Synthesis

This section presents a mathematical programming technique to minimize the structure of an optimal supervisor.

A. Definition of LMILP

An LMILP is an ILP with multiple objectives that are prioritized as follows:

$$\begin{aligned} &\text{lex min} \quad \{O_1(x), O_2(x), \ldots, O_r(x)\} \\ &\text{subject to} \;\; Ax \geq b \\ &\qquad\qquad\quad x \geq 0 \end{aligned}$$

where $x \in \mathbb{Z}^n$, $A \in \mathbb{Z}^{m\times n}$, $b \in \mathbb{Z}^m$, and $O_i(x) : \mathbb{Z}^n \rightarrow \mathbb{Z}$, for $i = 1, \ldots, r$. The lexicographic method assumes that the objectives are ranked in the order of importance [59]. We can assume, without loss of generality, that the objective functions are in the order of importance such that $O_1(x)$ is the most important and $O_r(x)$ the least important to decision makers. We first obtain the minimal value of $O_1(x)$ subject to the constraints, denoted as $O_1^*(x)$. Next, an ILP is solved with objective $O_2(x)$ subject to the above constraints, and $O_1(x) \leq O_1^*(x)$. The process is continued till all r objectives have been handled.

B. Optimal and Structurally Minimal Supervisor Design

More than one FBM may be forbidden by a PI. Given the PI I_j for $M_j \in \mathcal{M}_F^*$, there exists $M_k \in \mathcal{M}_F^* (k \neq j)$ is also forbidden by I_j if

$$\sum_{i\in\mathbb{N}_A} l_{j,i} \cdot \big(M_k(p_i) - M_j(p_i)\big) \geq -\Gamma \cdot \big(1 - f_{j,k}\big) \qquad \forall M_k \in \mathcal{M}_F^* \text{ and } k \neq j \quad (13)$$

where $l_{j,i}$ is the coefficient of PI I_j, Γ is a positive integer that should be chosen big enough, and $f_{j,k} \in \{0, 1\}$. In (13), $f_{j,k} = 1$ represents that I_j designed to forbid M_j also forbids M_k and $f_{j,k} = 0$ represents that I_j does not forbid M_k.

A set of variables $q_j (j \in \mathbb{N}_F^*)$ for the PI I_j are introduced to satisfy the following constraints:

$$f_{j,k} \leq q_j, \forall j, k \in \mathbb{N}_F^* \text{ and } k \neq j \quad (14)$$

where $\mathbb{N}_F^* = \{i | M_i \in \mathcal{M}_F^*\}$ and $q_j \in \{0, 1\}$. In (14), $q_j = 1$ represents that I_j is selected to design a control place in the supervisor and $q_j = 0$ implies not. Constraints (14) indicate that, only when I_j is selected, it can forbid M_j and other FBMs satisfying (13). To ensure that any FBM M_j in $\mathcal{M}_F^*$ is forbidden, $f_{k,j}$ and q_j should satisfy

$$q_j + \sum_{k\in\mathbb{N}_F^*, k\neq j} f_{k,j} \geq 1. \quad (15)$$

To design a structurally minimal supervisor for the PN model of an AMS, our first objective is to minimize the number of control places in a supervisor. In this case, the number of PIs selected to compute control places should be minimal. Thus, we employ

$$O_1 = \sum_{j\in\mathbb{N}_F^*} q_j. \quad (16)$$

Next, we minimize the number of arcs added in the supervisor when objective O_1 is minimized. A function sign() is introduced such that $\text{sign}([N_c](j, n)) = 0$ if $[N_c](j, n) = 0$ and $\text{sign}([N_c](j, n)) = 1$, otherwise, where N_c is the incident matrix of the supervisor, $j \in \mathbb{N}_F^*$, and $n \in \mathbb{N}_T = \{i | t_i \in T\}$. $\text{sign}([N_c](j, n)) = 1$ indicates that there exists an arc between control place p_{c_j} and transition t_n; while $\text{sign}([N_c](j, n)) = 0$ implies not. To minimize the number of added arcs in a supervisor, we seek to minimize

$$O_2 = \sum_{j\in\mathbb{N}_F^*} \sum_{n\in\mathbb{N}_T} \text{sign}([N_c](j, n)). \quad (17)$$

Function sign() allows us to only consider whether there is an arc to be added from or to a control place without taking into account its weights. Thus, the optimal supervisor with the minimal number of added arcs while ensuring the minimal number of control places can be found by solving the following lexicographic multiobjective integer program:

$$\text{lex min} \quad \{O_1, O_2\} \quad (18)$$

$$\text{subject to } (12) - (15). \quad (19)$$

However, the above program is not linear since O_2 is not. To convert it into an LMILP, two sets of auxiliary variables related to $l_{j,i}$ and q_j are introduced. The first set of variables is $u_{j,n} \in \{0, 1\}$ that represents whether there is an arc from control place p_{c_j} to transition $t_n \in T$ in the controlled net. Let $[N_c](j, \cdot)$ denote the incidence vector of p_{c_j}, where $[N_c](j, n) = W(t_n, p_{c_j}) - W(p_{c_j}, t_n)$. According to (6), $[N_c](j, n) = -\sum_{i\in\mathbb{N}_A} l_{j,i} \cdot [N_p](i, n)$. Thus, we have

$$-\sum_{i\in\mathbb{N}_A} l_{j,i} \cdot \big[N_p\big](i, n) \geq -\Gamma \cdot u_{j,n} - \Gamma \cdot \big(1 - q_j\big) \qquad \forall j \in \mathbb{N}_F^* \text{ and } \forall n \in \mathbb{N}_T. \quad (20)$$

Constraints (20) imply that if $q_j = 1$ and $[N_c](j, n) \leq -1$, then $u_{j,n} = 1$. In this case, there exists an arc from p_{c_j} to t_n in the controlled net. The second set of variables $v_{j,n} \in \{0, 1\}$ is introduced to represent whether there exists an arc from $t_n \in T$ to p_{c_j}. Then, we have

$$\sum_{i\in\mathbb{N}_A} l_{j,i} \cdot \big[N_p\big](i, n) \geq -\Gamma \cdot v_{j,n} - \Gamma \cdot \big(1 - q_j\big) \qquad \forall j \in \mathbb{N}_F^* \text{ and } \forall n \in \mathbb{N}_T. \quad (21)$$

Constraints (21) indicate that if $q_j = 1$ and $[N_c](j, n) \geq 1$, then $v_{j,n} = 1$. That is to say, there exists an arc from t_n to p_{c_j} in the controlled net.

Theorem 1: Consider an LMILP

$$\text{lex min} \quad \big\{O_1, O_2'\big\} \quad (22)$$

$$\text{subject to } (12) - (15), (20), \text{ and } (21) \quad (23)$$

where O_2' is the number of added arcs in the supervisor, that is

$$O_2' = \sum_{j\in\mathbb{N}_F^*} \sum_{n\in\mathbb{N}_T} \big(u_{j,n} + v_{j,n}\big). \quad (24)$$

Its solution gives the same results as that of (18) and (19).

Proof: If I_j is selected to construct a control place ($q_j = 1$), the newly added constraints (20) and (21) impose that $u_{j,n}+v_{j,n}$ has to be equal or greater than 1 if $\text{sign}([N_c](j, n)) = 1$. Since the cost of the objective function has to be minimized,

$u_{j,n}+v_{j,n}$ is selected to be 1 with $\mathrm{sign}([N_c](j,n)) = 1$. In addition, $u_{j,n} + v_{j,n}$ is set to 0 if $\mathrm{sign}([N_c](j,n)) = 0$ according to the definition of function sign().

On the other hand, if PI I_j is not selected to construct a control place ($q_j = 0$), no arc related to I_j is to be added, i.e., $\forall n \in \mathbb{N}_T$, $\mathrm{sign}([N_c](j,n)) = 0$. In this case, $u_{j,n} + v_{j,n}$ is also equal to 0. Thus, the integer program in (22) and (23) has the same results as that of (18) and (19). ■

The above LMILP can be used to obtain an optimal supervisor with the minimal number of added arcs while ensuring the fewest control places for the PN models of AMSs. In order to obtain the simplest expressions possible for the constraints, we minimize the PI coefficient $l_{j,i}$ as the third objective. Hence, we have

$$O_3 = \sum_{j\in\mathbb{N}_F^*} \sum_{i\in\mathbb{N}_A} l_{j,i}. \tag{25}$$

Combining all the above objectives and constraints, we have the following LMILP that is denoted as the lexicographic minimizations of the numbers of control places and added arcs (LMPA) problem:

LMPA

$$\text{lex min} \quad \{O_1, O_2', O_3\} \tag{26}$$

subject to

$$\sum_{i\in\mathbb{N}_A} l_{j,i} \cdot \big(M_l(p_i) - M_j(p_i)\big) \le -1 \quad \forall M_j \in \mathcal{M}_F^* \text{ and } \forall M_l \in \mathcal{M}_L^* \tag{27}$$

$$\sum_{i\in\mathbb{N}_A} l_{j,i} \cdot \big(M_k(p_i) - M_j(p_i)\big) \ge -\Gamma \cdot \big(1 - f_{j,k}\big) \quad \forall M_j, M_k \in \mathcal{M}_F^* \text{ and } j \ne k \tag{28}$$

$$f_{j,k} \le q_j, \forall j, k \in \mathbb{N}_F^* \text{ and } j \ne k \tag{29}$$

$$q_j + \sum_{k\in\mathbb{N}_F^*, k\ne j} f_{k,j} \ge 1, \forall j \in \mathbb{N}_F^* \tag{30}$$

$$-\sum_{i\in\mathbb{N}_A} l_{j,i} \cdot \big[N_p\big](i,n) \ge -\Gamma \cdot u_{j,n} - \Gamma \cdot \big(1 - q_j\big) \quad \forall j \in \mathbb{N}_F^* \text{ and } \forall n \in \mathbb{N}_T \tag{31}$$

$$\sum_{i\in\mathbb{N}_A} l_{j,i} \cdot \big[N_p\big](i,n) \ge -\Gamma \cdot v_{j,n} - \Gamma \cdot \big(1 - q_j\big) \quad \forall j \in \mathbb{N}_F^* \text{ and } \forall n \in \mathbb{N}_T \tag{32}$$

$$l_{j,i} \in \{0, 1, 2, \ldots\}, \forall i \in \mathbb{N}_A \text{ and } \forall j \in \mathbb{N}_F^*$$
$$f_{j,k} \in \{0, 1\}, \forall j, k \in \mathbb{N}_F^* \text{ and } j \ne k$$
$$q_j \in \{0, 1\}, \forall j \in \mathbb{N}_F^*$$
$$u_{j,n}, v_{j,n} \in \{0, 1\}, \forall j \in \mathbb{N}_F^* \text{ and } \forall n \in \mathbb{N}_T.$$

Note that, the coefficient $l_{j,i}(j \in \mathbb{N}_F^*$ and $i \in \mathbb{N}_A)$ is nonnegative and LMPA minimizes all objectives under this restriction.

The objective function in LMPA represents that the optimality is obtained first in terms of the number of control places, second the number of added arcs, and third the simplicity of the PIs coefficients of the controlled net. How many constraints and variables does LMPA have? Let $|T|$ and $|P_A|$ be the numbers of transitions and activity places in a PN model of AMS, respectively. (27) has $|\mathcal{M}_F^*|\cdot|\mathcal{M}_L^*|$ constraints. Since $M_k \ne M_j$, (28) has $|\mathcal{M}_F^*| \cdot (|\mathcal{M}_F^*| - 1)$. Similarly, (29) has $|\mathcal{M}_F^*| \cdot (|\mathcal{M}_F^*| - 1)$ constraints and (30) has $|\mathcal{M}_F^*|$ constraints. Finally, the types of constraints (31) and (32) have $|\mathcal{M}_F^*| \cdot |T|$ constraints each. The total number of all constraints in LMPA is $|\mathcal{M}_F^*| \cdot (2|T| + 2|\mathcal{M}_F^*| + |\mathcal{M}_L^*| - 1)$. Similarly, the number of variables $l_{j,i}(i \in \mathbb{N}_A, j \in \mathbb{N}_F^*)$ is $|\mathcal{M}_F^*|\cdot|P_A|$. The number of variables $q_j(j \in \mathbb{N}_F^*)$ is the same as the number of markings in $\mathcal{M}_F^*$, i.e., $|\mathcal{M}_F^*|$. Since $j \in \mathbb{N}_F^*$ and $n \in \mathbb{N}_T$, the number of variables $u_{j,n}$ is $|\mathcal{M}_F^*| \cdot |T|$. Similarly, the number of variables $v_{j,n}$ is also $|\mathcal{M}_F^*| \cdot |T|$. The number of variables $f_{j,k}$ is $|\mathcal{M}_F^*| \cdot (|\mathcal{M}_F^*| - 1)$ because $j, k \in \mathcal{M}_F^*$ and $j \ne k$. Therefore, the total number of all variables in LMPA is $|\mathcal{M}_F^*|\cdot(2|T| + |P_A| + |\mathcal{M}_F^*|)$. Table I summarizes the numbers of constraints and variables in LMPA.

Algorithm 1 Deadlock Prevention Method by Using LMPA

Input: A PN model (N, M_0) of an AMS.
Output: A controlled PN system.

Generate $G(N, M_0)$ and compute $\mathcal{M}_F$ and $\mathcal{M}_L$;
Compute $\mathcal{M}_F^*$ and $\mathcal{M}_L^*$;
$V_P := \emptyset$, $V_A := \emptyset$; /* V_P and V_A denote the sets of control places and added arcs in a supervisor. */
Solve LMPA proposed in Section IV-B. If it has no solution, exit;
for each $q_j = 1$ **do**
  Use $l_{j,i}$ in the solution as the coefficient of a PI and design a control place p_{c_j} and the set of arcs A_j associated to p_{c_j} by the method proposed in Section III-A;
  $V_P := V_P \cup \{p_{c_j}\}$, $V_A := V_A \cup \{A_j\}$;
end for
Add all control places in V_P and all arcs in V_A to (N, M_0) and output the resulting controlled net;
End.

V. Deadlock Prevention Method

This section presents a deadlock prevention method by using LMPA to obtain an optimal supervisor having the minimal number of added arcs while ensuring the fewest control places.

First, Algorithm 1 generates all markings in $\mathcal{M}_F$ and $\mathcal{M}_L$ of $G(N, M_0)$. Next, it computes $\mathcal{M}_F^*$ and $\mathcal{M}_L^*$ by using a vector covering method. Then, LMPA is solved to decide control places and associated arcs added in the supervisor. The resulting supervisor can forbid any $M \in \mathcal{M}_F$ by at least one PI and permit all legal markings in the PN model. The objectives of LMPA are first to minimize the number of control places in the supervisor, and then minimize the number of arcs to be added, and finally simplify the PIs coefficients. Last, such control places and added arcs constitute an optimal supervisor. All control places and associated arcs in the supervisor can be obtained by solving LMPA. The biggest advantage of Algorithm 1 is that it can definitely obtain an optimal supervisor with the minimal number of added arcs while ensuring the number of control places is the smallest if there exists such a PN supervisor.

TABLE I
NUMBERS OF CONSTRAINTS AND VARIABLES OF LMPA

Constraint	Number	Variable	Number
(27)	$\|\mathcal{M}_F^*\| \cdot \|\mathcal{M}_L^*\|$	$l_{j,i}$	$\|\mathcal{M}_F^*\| \cdot \|P_A\|$
(28)	$\|\mathcal{M}_F^*\| \cdot (\|\mathcal{M}_F^*\| - 1)$	q_j	$\|\mathcal{M}_F^*\|$
(29)	$\|\mathcal{M}_F^*\| \cdot (\|\mathcal{M}_F^*\| - 1)$	$u_{j,n}$	$\|\mathcal{M}_F^*\| \cdot \|T\|$
(30)	$\|\mathcal{M}_F^*\|$	$v_{j,n}$	$\|\mathcal{M}_F^*\| \cdot \|T\|$
(31)	$\|\mathcal{M}_F^*\| \cdot \|T\|$	$f_{j,k}$	$\|\mathcal{M}_F^*\| \cdot (\|\mathcal{M}_F^*\| - 1)$
(32)	$\|\mathcal{M}_F^*\| \cdot \|T\|$		
total	$\|\mathcal{M}_F^*\| \cdot (2\|T\| + 2\|\mathcal{M}_F^*\| + \|\mathcal{M}_L^*\| - 1)$	total	$\|\mathcal{M}_F^*\| \cdot (2\|T\| + \|P_A\| + \|\mathcal{M}_F^*\|)$

Theorem 2: Assume that every control place computed via Algorithm 1 is associated with a P-semiflow. Then, Algorithm 1 obtains an optimal supervisor with the minimal number of added arcs while ensuring the fewest control places for the PN model of an AMS if and only if LMPA has an optimal solution.

Proof: First, we prove that if LMPA has an optimal solution, each control place is associated with a P-semiflow, and Algorithm 1 can generate an optimal supervisor with the minimal number of added arcs while ensuring the fewest control places. As shown in [46, Th. 6], each control place is associated with a PI. Since the coefficient $l_{j,i}(j \in \mathbb{N}_F^*$ and $i \in \mathbb{N}_A)$ of any PI in LMPA is nonnegative, the PIs computed in Algorithm 1 are all P-semiflows. According to (30), LMPA forbids all FMBs since any $M \in \mathcal{M}_F^*$ is forbidden by at least one PI. On the other hand, LMPA permits all legal markings in $G(N, M_0)$ by (27). Thus, the supervisor obtained is optimal. According to the lexicographic objectives of LMPA, it first ensures that the number of control places is minimized, then the number of added arcs is minimized, and finally the coefficients of the constraints are minimized. Therefore, the obtained supervisor is optimal and has the minimal number of added arcs while ensuring the fewest control places.

Then, we aim to prove that if there exists an optimal supervisor with the premise that each control place in the supervisor is associated with a P-semiflow, then LMPA has an optimal solution. First, we prove that any P-semiflow for an optimal control purpose satisfies (27) in LMPA. Suppose that there exists a P-semiflow that does not satisfy (27) for some markings in $\mathcal{M}_L^*$. As Section III describes, the P-semiflow forbids those legal markings. So, the obtained control place designed for this P-semiflow is not optimal. By contradiction, it is proven that any P-semiflow considered for designing an optimal supervisor satisfies (27). Thus, there exists a solution which satisfies (27) for each FBM and all legal markings. Therefore, LMPA has a solution. In addition, since the objective function of LMPA is to lexicographically minimize O_1, O_2', and O_3, LMPA has a solution that is optimal. ■

We use the PN model shown in Fig. 1 to illustrate the proposed algorithm. It has 30 reachable markings, 27 of which are legal and three are FBMs. By using a vector covering method, we have $M_F^* = \{p_3 + p_7, p_2 + p_7 + p_8\}$ and $M_L^* = \{p_2 + p_3 + p_4, p_2 + p_3 + p_5, p_2 + p_7, p_2 + p_4 + p_8, p_2 + p_5 + p_8, p_7 + p_8 + p_9, p_5 + p_9, p_4 + p_9\}$. Note that only the tokens in activity places are considered for the design of a supervisor.

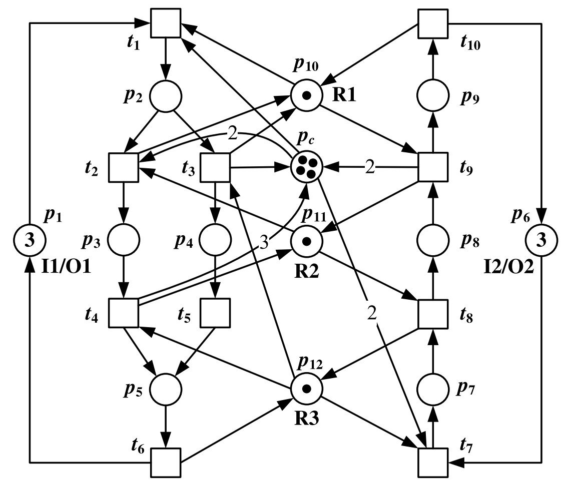


Fig. 3. Controlled system of the model in Fig. 1 by Algorithm 1.

The LMPA of the model, which has 62 constraints and 58 variables, is given in the Appendix. It has an optimal solution with $q_2 = 1$, $f_{2,1} = 1$ $l_{1,2} = l_{1,5} = l_{2,1} = 1$, $l_{2,2} = 3$, $l_{2,5} = l_{2,6} = 2$, $u_{2,1} = u_{2,2} = u_{2,7} = v_{2,3} = v_{2,4} = v_{2,9} = 1$, and all other variables equal zero. Since $q_2 = 1$, a control place p_{c_2} is designed for $I_2 : \ \mu_2 + 3\mu_3 + 2\mu_7 + 2\mu_8 + \mu_{p_{c_2}} = b$ as described in (5). According to the supervisor design method in Section III-A, we can obtain the preset, postset, and initial marking of p_{c_2}, i.e., ${}^\bullet p_{c_2} = \{t_3, 3t_4, 2t_9\}$, $p_{c_2}^\bullet = \{t_1, 2t_2, 2t_7\}$, and $M_0(p_{c_2}) = b = 4$. Thus, a control place associated with six arcs is obtained for this net. Table II shows the detailed results. Adding the control place and its arcs to the plant net, we obtain a live controlled net with 27 legal markings, as shown in Fig. 3. By Algorithm 1, we guarantee that this resulting controlled net is optimal and the obtained supervisor has the minimal number of arcs while guaranteeing the fewest control places.

VI. EXPERIMENTS

In this section, some well-studied AMSs are tested to show the proposed method in a computer with Intel i3 Core 2.93 GHz CPU and 4 GB memory. Integrated Net Analyzer (INA) [60] is used to compute $G(N, M_0)$, and a C++ program is developed to generate $\mathcal{M}_L^*$ and $\mathcal{M}_F^*$. Then, we use Lingo [61] as a linear program solver to find an optimal solution for LMPA. Finally, the liveness and behavioral permissiveness of the controlled net are verified via INA.

TABLE II
CONTROL PLACES AND ADDED ARCS OBTAINED FOR THE MODEL IN FIG. 1

i	FBM$_i$	I_i	${}^\bullet p_{c_i}$	$p_{c_i}^\bullet$	$M_0(p_{c_i})$
2	$p_2+p_7+p_8$	$\mu_2+3\mu_3+2\mu_7+2\mu_8 \leqslant 4$	$t_3, 3t_4, 2t_9$	$t_1, 2t_2, 2t_7$	4

TABLE III
CONTROL PLACES AND ADDED ARCS OBTAINED FOR THE MODEL IN FIG. 4

i	FBM$_i$	I_i	${}^\bullet p_{c_i}$	$p_{c_i}^\bullet$	$M_0(p_{c_i})$
5	$p_2+p_4+p_6+p_9+p_{10}$	$\mu_2+2\mu_3+\mu_4+2\mu_5+2\mu_6+3\mu_9+3\mu_{10} \leqslant 9$	$2t_7, 3t_{11}$	$t_1, t_2, t_4, 3t_9$	9
7	$p_2+p_4+p_{12}$	$\mu_2+2\mu_3+\mu_4+2\mu_{11}+2\mu_{12} \leqslant 3$	$t_4, 2t_5, 2t_{13}$	$t_1, t_2, 2t_{11}$	3

TABLE IV
PERFORMANCE COMPARISON OF SOME DEADLOCK CONTROL METHODS FOR THE MODEL IN FIG. 4

Parameters	[47]	[62]	[63]	[46]	[49]	Proposed method
No. control places	6	9	5	8	2	2
No. added arcs	32	42	23	37	12	12
No. Reachable markings	205	205	205	205	205	205

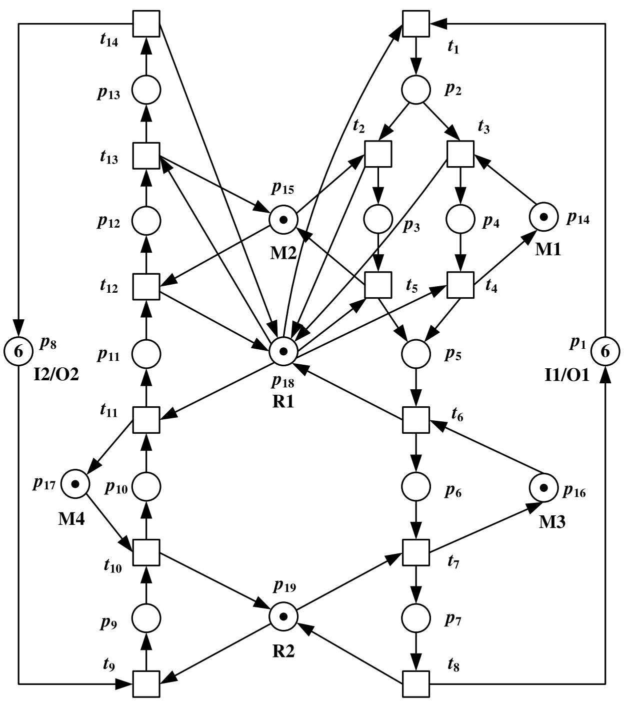


Fig. 4. PN model of an AMS [47], [62].

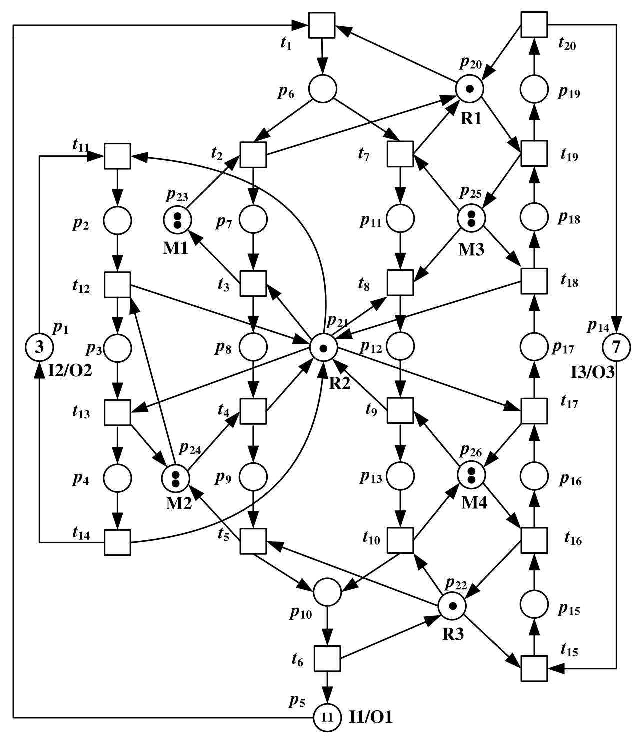


Fig. 5. S^{3}PR model in [14].

Consider an AMS [47], [62]–[64] with two robots R1-2, four machines M1-4, two loading buffers I1-2, and two unloading buffers O1-2. Two types of parts, J1 and J2, are processed in the system. The production sequences are

J1: I1 → R1 → M1 (or M2) → R1 → M3 → R2 → O1

J2: I2 → R2 → M4 → R1 → M2 → R1 → O2.

Its PN model is shown in Fig. 4. It has 19 places and 14 transitions. The places in the model are divided into idle places $\{p_1, p_8\}$, resource places $\{p_{14}-p_{19}\}$, and activity places $P_A = \{p_2-p_7, p_9-p_{13}\}$. There are 282 markings in $G(N, M_0)$, 205 and 54 of which are legal markings and FBMs, respectively. By using the vector covering method, $\mathcal{M}_L^*$ and $\mathcal{M}_F^*$ have 26 and eight markings, respectively. For this net, LMPA has 552 constraints and 376 variables. It takes 45 CPU seconds to solve it, yielding a solution with only two control places and 12 added arcs in the supervisor, as shown in Table III. Table IV shows the results from the methods available in literature and our method for the example regarding the numbers of control places, added arcs, and reachable markings of the controlled net. The results show that the best supervisor is obtained by the proposed method and the one in [49].

Then, a more complex AMS [14] is tested. There are three robots R1-3, four kinds of machines M1-4, three loading buffers I1-3, and three unloading buffers O1-3. Three types

TABLE V
CONTROL PLACES AND ADDED ARCS OBTAINED FOR THE MODEL IN FIG. 5

i	FBM_i	I_i	$^\bullet p_{c_i}$	$p_{c_i}^\bullet$	$M_0(p_{c_i})$
6	$p_6+2p_7+2p_9+p_{11}+p_{13}+p_{15}+p_{16}+p_{18}$	$72\mu_2+72\mu_3+2\mu_6+2\mu_7+73\mu_8+73\mu_9+2\mu_{11}+2\mu_{12}+387\mu_{13}+231\mu_{15}+231\mu_{16}+\mu_{17}+\mu_{18}\leqslant 1003$	$73t_5$, $387t_{10}$, $72t_{13}$, $230t_{17}$, t_{19}	$2t_1$, $71t_3$, $385t_9$, $72t_{11}$, $231t_{15}$	1003
13	$p_8+2p_9+p_{15}+2p_{16}$	$\mu_2+\mu_3+\mu_4+\mu_8+\mu_9+250\mu_{12}+250\mu_{13}+3\mu_{15}+497\mu_{16}\leqslant 999$	t_5, $250t_{10}$, t_{14}, $497t_{17}$	t_3, $250t_8$, t_{11}, $3t_{15}$, $494t_{16}$	999
16	$p_6+p_7+2p_9+p_{11}+p_{15}+2p_{16}+p_{17}$	$146\mu_6+146\mu_7+17\mu_8+17\mu_9+146\mu_{11}+23\mu_{12}+23\mu_{13}+34\mu_{15}+34\mu_{16}+125\mu_{17}+141\mu_{18}\leqslant 844$	$129t_3$, $17t_5$, $123t_8$, $23t_{10}$, $141t_{19}$	$146t_1$, $34t_{15}$, $91t_{17}$, $16t_{18}$	844
27	$2p_3+p_8$	$\mu_2+\mu_3+\mu_8\leqslant 2$	t_4, t_{13}	t_3, t_{11}	2
32	$2p_{11}+p_{17}$	$\mu_2+\mu_3+\mu_8+\mu_9+3\mu_{10}+375\mu_{11}+3\mu_{12}+3\mu_{13}+122\mu_{15}+125\mu_{16}+250\mu_{17}\leqslant 999$	$3t_6$, $372t_8$, t_{13}, $250t_{18}$	t_3, $2t_5$, $375t_7$, t_{11}, $122t_{15}$, $3t_{16}$, $125t_{17}$	999

TABLE VI
PERFORMANCE COMPARISON OF SOME DEADLOCK CONTROL METHODS FOR THE MODEL IN FIG. 5

Parameters	[14]	[41]	[39]	[45]	[46]	[63]	[65]	[49]	Proposed method
No. control places	18	6	16	19	17	13	6	5	5
No. added arcs	106	32	88	112	101	82	45	55	43
No. Reachable markings	6287	6287	12656	21562	21581	21581	21581	21581	21581

of parts, J1–3, are processed with their production sequences

J1: I1 → R1 → M1 → R2 → M2 → R3 → O1
or I1 → R1 → M3 → R2 → M4 → R3 → O1
J2: I2 → R2 → M2 → R2 → O2
J3: I3 → R3 → M4 → R2 → M3 → R1 → O3.

Its PN model is shown in Fig. 5. It has 26 places and 20 transitions. The places are divided into idle places $\{p_1, p_5, p_{14}\}$, resource places $\{p_{20}\text{–}p_{26}\}$, and activity places $P_A = \{p_2\text{–}p_4, p_6\text{–}p_{13}, p_{15}\text{–}p_{19}\}$. There are 26 750 reachable markings in $G(N, M_0)$, 21 581 and 4211 of which are legal ones and FBMs, respectively. By using the vector covering method, $\mathcal{M}_L^*$ and $\mathcal{M}_F^*$ have 393 and 34 markings, respectively. For this example, LMPA has 17 000 constraints and 3060 variables, which are too many to obtain the final result in a reasonable time. Instead, by solving it for about 300 h, we interrupt it and still obtain an optimal controlled net while requiring five control places and 43 arcs in the supervisor, as shown in Table V. Table VI shows the state-of-the-art results for the PN model in Fig. 5 regarding the numbers of control places, added arcs, and reachable markings of the controlled net. We can see that the supervisor obtained by our method is optimal since the number of reachable markings of the controlled net is equal to $|\mathcal{M}_L| = 21581$. In addition, it has only five control places and 43 added arcs, both of which are the smallest in comparison with optimal or near-optimal supervisors obtained with the methods in [45], [46], [49], [63], and [65]. It significantly beats the nonoptimal ones [14], [39] in both optimality and structural simplicity. It has one control place less than the one in [41], 11 arcs more than that in [41], but over three times better in terms of the behavioral optimality than that in [41]. We conclude that overall the proposed method is the best among all existing ones. In addition, the more time this LMPA has run, the better results it may achieve.

The advantages of the proposed method are that: 1) all control places and added arcs in a supervisor can be computed by solving an LMPA and 2) the controlled net is optimal and the supervisor obtained is structurally minimal. The disadvantage is that the problem to be solved is NP-hard in theory, which is the same as the methods in [44]–[46], [49], and [51].

VII. CONCLUSION

This paper proposes an LMILP method to design optimal supervisors with the fewest control places and arcs for the PN models of AMSs if such supervisors exist. Such formulation is never seen in literature to the best knowledge of the authors. The supervisors obtained by the proposed method are optimal and have fewer added control places and arcs in comparison with the optimal controllers obtained by the existing methods.

High computational complexity is a problem to be handled by the proposed method. Our solution process requires a whole reachability graph. The reachability graph grows exponentially with the net size, resulting in the so-called state explosion problem. LMPA is a linear programming problem whose computational cost greatly depends on the number of its constraints. Therefore, our future work will focus on how to solve it iteratively and to identify and eliminate its redundant constraints [66] so as to alleviate its computational burden. In addition, Γ is a positive integer whose ideal value should be infinity. Its selection may be critical, since feasible solutions may be lost if it is too small, while large Γ may cripple an LMPA solution process. Thus, its relations with the solution speed and quality also interest us.

APPENDIX

LMPA OF THE PN MODEL SHOWN IN FIG. 1

LMPA

$$\text{lex min}\left\{\sum_{j\in\mathbb{N}_F^*} q_j, \sum_{j\in\mathbb{N}_F^*}\sum_{n\in\mathbb{N}_T}\left(u_{j,n}+v_{j,n}\right), \sum_{j\in\mathbb{N}_F^*}\sum_{i\in\mathbb{N}_A} l_{j,i}\right\}$$

subject to

$$l_{1,1} + l_{1,3} - l_{1,5} \leq -1$$
$$l_{1,1} + l_{1,4} - l_{1,5} \leq -1$$
$$l_{1,1} - l_{1,2} \leq -1$$
$$l_{1,1} - l_{1,2} + l_{1,3} - l_{1,5} + l_{1,6} \leq -1$$
$$l_{1,1} - l_{1,2} + l_{1,4} - l_{1,5} + l_{1,6} \leq -1$$
$$-l_{1,2} + l_{1,6} + l_{1,7} \leq -1$$
$$-l_{1,2} + l_{1,4} - l_{1,5} + l_{1,7} \leq -1$$
$$-l_{1,2} + l_{1,3} - l_{1,5} + l_{1,7} \leq -1$$
$$l_{2,2} + l_{2,3} - l_{2,5} - l_{2,6} \leq -1$$
$$l_{2,2} + l_{2,4} - l_{2,5} - l_{2,6} \leq -1$$
$$-l_{2,6} \leq -1$$
$$l_{2,3} - l_{2,5} \leq -1$$
$$l_{2,4} - l_{2,5} \leq -1$$
$$-l_{2,1} + l_{2,7} \leq -1$$
$$-l_{2,1} + l_{2,4} - l_{2,5} - l_{2,6} + l_{2,7} \leq -1$$
$$-l_{2,1} + l_{2,3} - l_{2,5} - l_{2,6} + l_{2,7} \leq -1$$
$$l_{1,1} - l_{1,2} + l_{1,6} \geq -\Gamma \cdot (1 - f_{1,2})$$
$$-l_{2,1} + l_{2,2} - l_{2,6} \geq -\Gamma \cdot (1 - f_{2,1})$$
$$f_{1,2} \leq q_1$$
$$f_{2,1} \leq q_2$$
$$q_1 + f_{2,1} \geq 1$$
$$q_2 + f_{1,2} \geq 1$$
$$-l_{1,1} \geq -\Gamma \cdot (u_{1,1} - q_1 + 1)$$
$$l_{1,1} - l_{1,2} \geq -\Gamma \cdot (u_{1,2} - q_1 + 1)$$
$$l_{1,1} - l_{1,3} \geq -\Gamma \cdot (u_{1,3} - q_1 + 1)$$
$$l_{1,2} - l_{1,4} \geq -\Gamma \cdot (u_{1,4} - q_1 + 1)$$
$$l_{1,3} - l_{1,4} \geq -\Gamma \cdot (u_{1,5} - q_1 + 1)$$
$$l_{1,4} \geq -\Gamma \cdot (u_{1,6} - q_1 + 1)$$
$$-l_{1,5} \geq -\Gamma \cdot (u_{1,7} - q_1 + 1)$$
$$l_{1,5} - l_{1,6} \geq -\Gamma \cdot (u_{1,8} - q_1 + 1)$$
$$l_{1,6} - l_{1,7} \geq -\Gamma \cdot (u_{1,9} - q_1 + 1)$$
$$l_{1,7} \geq -\Gamma \cdot (u_{1,10} - q_1 + 1)$$
$$-l_{2,1} \geq -\Gamma \cdot (u_{2,1} - q_2 + 1)$$
$$l_{2,1} - l_{2,2} \geq -\Gamma \cdot (u_{2,2} - q_2 + 1)$$
$$l_{2,1} - l_{2,3} \geq -\Gamma \cdot (u_{2,3} - q_2 + 1)$$
$$l_{2,2} - l_{2,4} \geq -\Gamma \cdot (u_{2,4} - q_2 + 1)$$
$$l_{2,3} - l_{2,4} \geq -\Gamma \cdot (u_{2,5} - q_2 + 1)$$
$$l_{2,4} \geq -\Gamma \cdot (u_{2,6} - q_2 + 1)$$
$$-l_{2,5} \geq -\Gamma \cdot (u_{2,7} - q_2 + 1)$$
$$l_{2,5} - l_{2,6} \geq -\Gamma \cdot (u_{2,8} - q_2 + 1)$$
$$l_{2,6} - l_{2,7} \geq -\Gamma \cdot (u_{2,9} - q_2 + 1)$$
$$l_{2,7} \geq -\Gamma \cdot (u_{2,10} - q_2 + 1)$$
$$l_{1,1} \geq -\Gamma \cdot (v_{1,1} - q_1 + 1)$$
$$-l_{1,1} + l_{1,2} \geq -\Gamma \cdot (v_{1,2} - q_1 + 1)$$
$$-l_{1,1} + l_{1,3} \geq -\Gamma \cdot (v_{1,3} - q_1 + 1)$$
$$-l_{1,2} + l_{1,4} \geq -\Gamma \cdot (v_{1,4} - q_1 + 1)$$
$$-l_{1,3} + l_{1,4} \geq -\Gamma \cdot (v_{1,5} - q_1 + 1)$$
$$-l_{1,4} \geq -\Gamma \cdot (v_{1,6} - q_1 + 1)$$
$$l_{1,5} \geq -\Gamma \cdot (v_{1,7} - q_1 + 1)$$
$$-l_{1,5} + l_{1,6} \geq -\Gamma \cdot (v_{1,8} - q_1 + 1)$$
$$-l_{1,6} + l_{1,7} \geq -\Gamma \cdot (v_{1,9} - q_1 + 1)$$
$$-l_{1,7} \geq -\Gamma \cdot (v_{1,10} - q_1 + 1)$$
$$l_{2,1} \geq -\Gamma \cdot (v_{2,1} - q_2 + 1)$$
$$-l_{2,1} + l_{2,2} \geq -\Gamma \cdot (v_{2,2} - q_2 + 1)$$
$$-l_{2,1} + l_{2,3} \geq -\Gamma \cdot (v_{2,3} - q_2 + 1)$$
$$-l_{2,2} + l_{2,4} \geq -\Gamma \cdot (v_{2,4} - q_2 + 1)$$
$$-l_{2,3} + l_{2,4} \geq -\Gamma \cdot (v_{2,5} - q_2 + 1)$$
$$-l_{2,4} \geq -\Gamma \cdot (v_{2,6} - q_2 + 1)$$
$$l_{2,5} \geq -\Gamma \cdot (v_{2,7} - q_2 + 1)$$
$$-l_{2,5} + l_{2,6} \geq -\Gamma \cdot (v_{2,8} - q_2 + 1)$$
$$-l_{2,6} + l_{2,7} \geq -\Gamma \cdot (v_{2,9} - q_2 + 1)$$
$$-l_{2,7} \geq -\Gamma \cdot (v_{2,10} - q_2 + 1)$$
$$l_{j,i} \in \{0, 1, 2, \ldots\}, \forall i \in \mathbb{N}_A \text{ and } \forall j \in \mathbb{N}_F^*$$
$$f_{j,k} \in \{0, 1\}, \forall j, k \in \mathbb{N}_F^* \text{ and } j \neq k$$
$$q_j \in \{0, 1\}, \forall j \in \mathbb{N}_F^*$$
$$u_{j,n}, v_{j,n} \in \{0, 1\}, \forall j \in \mathbb{N}_F^* \text{ and } \forall n \in \mathbb{N}_T$$
$$\mathbb{N}_A = \{1, 2, 3, \ldots, 7\}$$
$$\mathbb{N}_F^* = \{1, 2\}$$
$$\mathbb{N}_T = \{1, 2, 3, \ldots, 10\}.$$

References

[1] M. P. Fanti, B. Maione, and B. Turchiano, "Comparing digraph and Petri net approaches to deadlock avoidance in FMS," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 30, no. 5, pp. 783–798, Oct. 2000.

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

[3] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, Jan. 1989.

[4] A. Nazeem, S. Reveliotis, Y. Wang, and S. Lafortune, "Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory: The linear case," *IEEE Trans. Autom. Control*, vol. 56, no. 8, pp. 1818–1833, Aug. 2011.

[5] M. Zhou and F. DiCesare, "Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 515–527, Aug. 1991.

[6] M. Zhou and N. Wu, *System Modeling and Control With Resource-Oriented Petri Nets*. Boca Raton, FL, USA: CRC, 2010.

[7] Z. Li and M. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. London, U.K.: Springer, 2009.

[8] J. Li, M. Zhou, and X. Dai, "Reduction and refinement by algebraic operations for Petri net transformation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 5, pp. 1244–1255, Sep. 2012.

[9] M. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Boston, MA, USA: Kluwer, 1993.

[10] S. Wang, C. Wang, M. Zhou, and Z. Li, "A method to compute strict minimal siphons in a class of Petri nets based on loop resource subsets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 1, pp. 226–237, Jan. 2012.

[11] B. Huang, Y. Sun, and Y. Sun, "Scheduling of flexible manufacturing systems based on Petri nets and hybrid heuristic search," *Int. J. Prod. Res.*, vol. 46, no. 16, pp. 4553–4565, Aug. 2008.

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

[13] Y. Chen, Z. Li, and M. Zhou, "Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 549–563, Apr. 2014.

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

[15] 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.

[16] Z. Li, H. Hu, and A. 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.

[17] Z. Li and M. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Ind. Inf.*, vol. 2, no. 4, pp. 313–325, Nov. 2006.

[18] H. Hu, M. Zhou, and Z. Li, "Supervisor optimization for deadlock resolution in automated manufacturing systems with Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 4, pp. 794–804, Oct. 2011.

[19] H. Hu, M. Zhou, and Z. Li, "Liveness and ratio-enforcing supervision of automated manufacturing systems using Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 2, pp. 392–403, Mar. 2012.

[20] H. Hu, M. Zhou, Z. Li, and Y. Tang, "Deadlock-free control of automated manufacturing systems with flexible routes and assembly operations using Petri nets," *IEEE Trans. Ind. Inf.*, vol. 9, no. 1, pp. 109–121, Feb. 2013.

[21] N. Wu, M. Zhou, and G. Hu, "One-step look-ahead maximally permissive deadlock control of AMS by using Petri nets," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 1, pp. 10:1–10:23, Jan. 2013.

[22] V. D. Gligor and S. H. Shattuck, "On deadlock detection in distributed systems," *IEEE Trans. Softw. Eng.*, vol. SE-6, no. 5, pp. 435–440, Sep. 1980.

[23] T. Kumaran, W. Chang, H. Cho, and R. Wysk, "A structured approach to deadlock detection, avoidance and resolution in flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 32, no. 10, pp. 2361–2379, Oct. 1994.

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

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

[26] J. Ezpeleta and L. Recalde, "A deadlock avoidance approach for nonsequential resource allocation systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 1, pp. 93–101, Jan. 2004.

[27] F. Hsieh, "Fault-tolerant deadlock avoidance algorithm for assembly processes," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 1, pp. 65–79, Jan. 2004.

[28] N. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 29, no. 2, pp. 192–204, May 1999.

[29] N. Wu and M. Zhou, "Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 658–669, Oct. 2001.

[30] N. Wu and M. Zhou, "Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1193–1202, Dec. 2005.

[31] 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.

[32] J. Luo, W. Wu, H. 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.

[33] H. H. Xiong and M. Zhou, "Deadlock-free scheduling of an automated manufacturing system based on Petri nets," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2. Albuquerque, NM, USA, 1997, pp. 945–950.

[34] I. B. Abdallah, H. A. Elmaraghy, and T. Elmekkawy, "Deadlock-free scheduling in flexible manufacturing systems using Petri nets," *Int. J. Prod. Res.*, vol. 40, no. 12, pp. 2733–2756, Aug. 2002.

[35] H. Liu, K. Xing, W. Wu, M. Zhou, and Z. Gao, "Deadlock prevention for flexible manufacturing systems via controllable siphon basis of Petri nets," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 3, pp. 519–529, Mar. 2015.

[36] B. Huang, Y. Sun, Y. Sun, and C. Zhao, "A hybrid heuristic search algorithm for scheduling FMS based on Petri net model," *Int. J. Adv. Manuf. Technol.*, vol. 48, nos. 9–12, pp. 925–933, Jun. 2010.

[37] B. Huang, X. Shi, and N. Xu, "Scheduling FMS with alternative routings using Petri nets and near admissible heuristic search," *Int. J. Adv. Manuf. Technol.*, vol. 63, nos. 9–12, pp. 1131–1136, Dec. 2012.

[38] B. Huang, R. Jiang, and G. Zhang, "Search strategy for scheduling flexible manufacturing systems simultaneously using admissible heuristic functions and nonadmissible heuristic functions," *Comput. Ind. Eng.*, vol. 71, pp. 21–26, May 2014.

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

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

[41] 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.

[42] K. Xing, M. Zhou, F. Wang, H. Liu, and F. Tian, "Resource-transition circuits and siphons for deadlock control of automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 41, no. 1, pp. 74–84, Jan. 2011.

[43] H. Liu, K. Xing, M. Zhou, L. Han, and F. Wang, "Transition cover-based design of Petri net controllers for automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 44, no. 2, pp. 196–208, Feb. 2014.

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

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

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

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

[48] 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, Jan. 1996.

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

[50] R. Cordone and L. Piroddi, "Monitor optimization in Petri net control," in *Proc. IEEE Conf. Autom. Sci. Eng.*, Trieste, Italy, 2011, pp. 413–418.

[51] Y. Chen, Z. Li, and K. Barkaoui, "Maximally permissive liveness-enforcing supervisor with lowest implementation cost for flexible manufacturing systems," *Inf. Sci.*, vol. 256, pp. 74–90, Jan. 2014.

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

[53] F. Tricas, F. Garcia-Valles, J. Colom, and J. Ezpeleta, "A structural approach to the problem of deadlock prevention in processes with resources," in *Proc. Int. Workshop Discrete Event Syst. (WODES)*, Cagliari, Italy, Aug. 1998, pp. 273–278.

[54] F. Tricas, F. Garcia-Valles, J. Colom, and J. Ezpeleta, "An iterative method for deadlock prevention in FMS," in *Discrete Event Systems*. New York, NY, USA: Springer, 2000, pp. 139–148.

[55] 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.

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

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

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

[59] J. P. Ignizio and L. C. Thomas, "An enhanced conversion scheme for lexicographic, multiobjective integer programs," *Eur. J. Oper. Res.*, vol. 18, no. 1, pp. 57–61, Oct. 1984.

[60] (Jun. 1, 2014). *Integrated Net Analyzer—INA (Version 2.2)*. [Online]. Available: http://www2.informatik.hu-berlin.de/starke/ina.html

[61] (Jun. 1, 2014). *Linear Interactive and General Optimizer—Lingo (Version 12)*. [Online]. Available: http://www.lindo.com

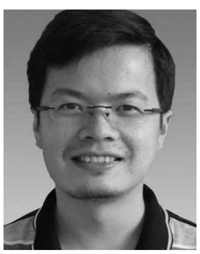
[62] Z. Li, M. Zhou, and M. Jeng, "A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 182–188, Mar. 2008.

[63] 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.

[64] 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.

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

[66] B. Huang, H. Zhu, G. Zhang, and X. Lu, "On further reduction of constraints in nonpure Petri net supervisors for optimal deadlock control of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 3, pp. 542–543, Mar. 2015.



Bo Huang (M'14) received the B.S. and Ph.D. degrees from the Nanjing University of Science and Technology, Nanjing, China, in 2002 and 2006, respectively.

He joined the Nanjing University of Science and Technology, in 2007, where he is currently an Associate Professor with the School of Computer Science and Engineering. His current research interests include Petri net theory and applications, control and scheduling of discrete event systems, and computer-integrated systems.

MengChu Zhou (S'88–M'90–SM'93–F'03) received the B.S. degree in control engineering from the Nanjing University of Science and Technology, Nanjing, China, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, and the Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, NY, USA, in 1983, 1986, and 1990, respectively.

He joined the New Jersey Institute of Technology, Newark, NJ, USA, in 1990, where he is currently a Distinguished Professor of Electrical and Computer Engineering. His current research interests include Petri nets, sensor networks, Web services, big data, semiconductor manufacturing, transportation, and energy systems. He has authored over 580 publications including 11 books, over 280 journal papers (majority in the IEEE TRANSACTIONS), and 22 book chapters.

Prof. Zhou is the Founding Editor of the IEEE Press Book Series on Systems Science and Engineering, and an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, and the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS. He is a Fellow of the International Federation of Automatic Control and American Association for the Advancement of Science. He is a Life Member of the Chinese Association for Science and Technology-USA and served as President in 1999.

GongXuan Zhang (SM'12) received the B.S. degree in electronic computer from Tianjin University, Tianjin, China, in 1983, and the M.S. and Ph.D. degrees in applied computer from the Nanjing University of Science and Technology, Nanjing, China, in 1991 and 2005, respectively.

He joined the Nanjing University of Science and Technology in 1991, where he is currently a Professor and a Vice Dean of the School of Computer Science and Engineering. From 2001 to 2002, he was a Visiting Professor with the Distributed Computing Group, School of Computer Science and Information, Royal Melbourne Institute of Technology, Melbourne, VIC, Australia. His current research interests include Web services, distributed computing, multicore, and parallel processing. He has served on the program committees in a couple of conferences. He has over 80 publications and has led or participated in 38 research projects supported by the National Science Foundation of China, and the Provincial Science Foundation of Jiangsu.

Ahmed Chiheb Ammari (M'12) received the B.S. degree from the National School of Engineering, Monastir, Tunisia, in 1993, and the M.S. and Ph.D. degrees from the Institut National Polytechnique de Grenoble, Grenoble, France, in 1993 and 1996, respectively, all in electrical engineering.

Since 1997, he has been a Faculty Member of Electrical Engineering with the Institut National des Sciences Appliquées et de Technologies, Carthage University, Tunis, Tunisia. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia. His current research interests include multicore and multiprocessor system-on-chip, power management, inductive link optimization for wireless power transfer, energy storage systems, and smart grid modeling and control.

Ahmed Alabdulwahab (SM'13) received the B.S. degree in electrical and computer engineering from King Abdulaziz University, Jeddah, Saudi Arabia, in 1996, and the M.S. and Ph.D. degrees in electrical engineering from the University of Saskatchewan, Saskatoon, SK, Canada, in 2000 and 2003, respectively.

He was a Consultant with the Electricity and Co-Generation Regulatory Authority in Saudi Arabia and a Visiting Scientist with Kinectrics, Toronto, ON, Canada, and the University of Manchester, Manchester, U.K. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, King Abdulaziz University, Jeddah, Saudi Arabia. His current research interests include power system planning and reliability evaluations and complex system modeling and control.

Ayman G. Fayoumi (SM'12) received the M.S. and Ph.D. degrees in computer engineering in 2001 and 2005, respectively, and the MBA degree in business administration from Colorado State University, Fort Collins, CO, USA, in 2005.

He has been an IT Consultant with several government and private organizations for ten years. He is currently an Associate Professor with the Faculty of Computing and Information Technology, King Abdulaziz University and a Founder of the Information Technologies House of Expert, Saudi Arabia. He chairs the National Committee for Innovation and Technology Commission (ITC) and Information Technology and Communication Standardization, Saudi Arabia. His areas of expertise are optical networking, IT strategic planning, smart IT solutions, and cloud computing. He has published over 40 research papers in refereed conferences and journals. He was listed in the Chancellors List of the Highest Academic Honors for Graduate Students, USA, in 2005.

Dr. Fayoumi is an editorial board member of the *Journal of Computing and Applications*, the International Association for Computer Scientists and Engineers.