Two Simple Deadlock Prevention Policies for $\mathrm{S}^3\mathrm{PR}$ Based on Key-Resource/Operation-Place Pairs

GuanJun Liu, ChangJun Jiang, and MengChu Zhou, *Fellow, IEEE*

***Abstract*—This paper proposes the concept of Key-resource/operation-place Pairs (KP) of $\mathrm{S}^3\mathrm{PR}$ (systems of simple sequential processes with resources). Based on KP, two policies are presented to prevent deadlocks in flexible manufacturing systems (FMS) that can be modeled by $\mathrm{S}^3\mathrm{PR}$. The idea is to control some key resource places only to guarantee that all strict minimal siphons (SMS) never become empty, thereby making the controlled system live. It enables one to design two easy-to-implement control policies. The first one can guarantee that the controlled system is live, and the second one can also make the controlled system live if there is no SMS containing any control place in the controlled system. At last, a well-known FMS example is used to illustrate the proposed concept and policies.**

***Note to Practitioners*—This work is motivated by the problem of deadlock prevention in flexible manufacturing systems (FMSs). Petri-net-based deadlock prevention policies are popular. We propose a new concept of Petri nets, KP. Based on it, we develop two easy-to-implement deadlock prevention policies for FMS. The idea behind them is simple, i.e., deadlocks can be prevented by controlling some key resources such as machines and robots. Compared with such policies as iterative control policy and elementary-siphon-based policy, the proposed ones can make the supervisor simpler. They also enjoy higher adaptability to the change of the resource quantities. In other words, when some new resources are put in service or some resources have to quit, the corresponding supervisor requires trivial adjustment. This is certainly important in achieving the goal of agile manufacturing and automation. However, the computational complexity of the proposed policies is still high despite their low control implementation cost since all strict minimal siphons have to be computed in order to generate the KP-based supervisors.**

***Index Terms*—Deadlock prevention, discrete-event systems (DES), flexible manufacturing systems (FMSs), Petri nets.**

Manuscript received November 28, 2009; revised February 21, 2010; accepted April 22, 2010. Date of publication June 01, 2010; date of current version October 06, 2010. This paper was recommended for publication by Associate Editor N. Wu and Editor Y. Narahari upon evaluation of the reviewers' comments. This paper was supported in part by the National Natural Science Foundation of China under Grant 90818023 and Grant 90718012, in part by the National Basic Research Program of China under Grant 2010CB328100, in part by the National High Technology Research and Development Program of China under Grant 2009AA01Z401, in part by the Science and Technology Foundation of Shanghai under Grant 08JC1419300, and in part by the Ministry of Education of China under Grant IRT0744.

G. J. Liu is with the Department of Computer Science and Technology, Tongji University, Shanghai 201804, China (e-mail: liugj1116@163.com).

C. J. Jiang is with the Key Laboratory of the Ministry of Education for Embedded System and Service Computing, Tongji University, Shanghai 201804, China.

M. C. Zhou is with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA and also with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China.

Digital Object Identifier 10.1109/TASE.2010.2050059

I. Introduction

DEADLOCK resolution is an important issue in the design and operation of flexible manufacturing systems (FMSs) [9]. Once a deadlock situation occurs, some manufacturing processes can never be finished until it is resolved. There are mainly three methods to deal with deadlocks: prevention [1], [7]–[12], [18]–[22], [24]–[29], [31]–[37], avoidance [1]–[3], [16], [17], [37]–[44], [47]–[49], and detection/recovery [6], [45]. Some methods can guarantee that a system is live, while others permit deadlocks to occur, but must ensure that they can be detected and proper recovery measures can be taken.

Petri nets [10], [13], [20], [43] are widely used to model and analyze FMS owing to their formal semantics, graphical nature, and analysis techniques. Some FMS can be modeled by subclasses of Petri nets, such as PPN [2], $\mathrm{S}^3\mathrm{PR}$ [8], [18], [33], $\mathrm{ES}^3\mathrm{PR}$ [12], [31], $\mathrm{S}^4\mathrm{R}$ [1], [3], RSNB [24], [25] and ERCN-merged net [14], [15], [46]. Based on these subclasses, a variety of policies are proposed to solve the deadlock problem. Most of these policies are based on siphons that are an important structural object of Petri nets and can be computed by many methods [5], [23].

In general, a supervisor is constructed in order to prevent a system from deadlocks [13], [20]. It may be seen as a Petri net consisting of a set of control places called *monitors*, a set of transitions by which the supervisor is synthesized with the Petri net modeling the original system, a set of related arcs, and an initial marking. Clearly, the simpler a supervisor structure as measured by the number of places, transitions, and arcs, the lower its implementation complexity and cost. The more legal behavior of the plant net model it preserves, the better system performance goals it can help achieve.

$\mathrm{S}^3\mathrm{PR}$ (system of simple sequential processes with resources) has attracted many scholars' attention since it was proposed by Ezpeleta *et al.* [8]. It can model a class of FMS in which a set of different types of products can be manufactured concurrently and each step in one manufacturing process only needs one resource such as a machine or robot. Due to the competition among manufacturing processes for the limited resources, deadlocks can occur. One policy, which is based on strict minimal siphons (SMSs), is proposed in [8] to prevent deadlocks. This policy is very simple and can successfully prevent deadlocks. However, too many control places and arcs are required, resulting in a complex supervisor and thus controlled system. To overcome such deficiency, Li and Zhou pioneer in the concept of elementary siphons [18]. They can control all SMS by controlling elementary SMS only, leading to a simple supervisor.

When initial tokens are allocated for each control place, however, a linear integer programming (LIP) test must be carried out to decide the liveness of the controlled system. Li and Zhou have obtained many additional results on the basis of elementary siphons [19], [20], [22].

In this paper, a new concept, KP, is proposed. Our idea is to control an SMS by controlling only one of its resource places. A *cover* of KP is defined as a set of KP that can "cover" all SMS. Based on this concept, two deadlock prevention policies are proposed. Compared with [18], the advantages of our policies are that:

1) fewer control places and arcs are added although more legal markings are lost at some cases.
2) The LIP test is eliminated when allocating initial tokens to control places.
3) Our policies are more adaptable to the change of the quantity of resources. When it increases or decreases, only the number of tokens in control places need to increase or decrease, and sometimes is left unchanged. Note that the policy in [18] requires LIP test for each such change.

Section II gives preliminaries used in this paper. Section III defines the KP and their cover. Sections IV and V propose two deadlock prevention policies based on KP. Section VI shows the relationship between the cover of KP and elementary siphons, and then develops an algorithm to generate a cover. Section VII applies the proposed policies to a typical FMS. Section VIII compares the proposed policies with others through some experiments. Section IX concludes this paper by discussing future work.

II. Basic Definitions and Notations

This section recalls some basic definitions and notations used in this paper.

A. Petri Nets

The following concepts, notations and properties are from [9], [10], and [20].

A *net* is a 3-tuple $N = (P, T, F)$, where P is the set of *places*, T is the set of *transitions*, $F \subseteq (P \times T) \cup (T \times P)$ is the set of *arcs*, $P \cap T = \phi$, and $P \cup T \neq \phi$. t is called an *input transition* of p and p is called an *output place* of t if $(t, p) \in F$. *Input place* and *output transition* can be defined similarly. $\forall x \in P \cup T, {}^\bullet x \in \{y | y \in P \cup T \wedge (y, x) \in F\}$ and $x^\bullet = \{y | y \in P \cup T \wedge (x, y) \in F\}$ are the *preset* and *postset* of x, respectively. This notation can be extended to a set of nodes as follows: $\forall X \subseteq P \cup T, {}^\bullet X = \cup_{x \in X} {}^\bullet x$ and $X^\bullet = \cup_{x \in X} x^\bullet$. A transition t is called a *sink* if $t^\bullet = \phi$.

Let $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$, $\mathbb{N} = \{0, 1, 2, \ldots\}$, $\mathbb{N}^+ = \{1, 2, \ldots\}$, and $\mathbb{N}_k = \{1, 2, \ldots, k\}, \forall k \in \mathbb{N}^+$.

A *marking* of N is a mapping $M : P \to \mathbb{N}$. $M(p)$ indicates the number of tokens in p at M. p is marked at M if $M(p) > 0$. Given $S \subseteq P$, $M(S) = \Sigma_{p \in S} M(p)$. S is marked at M if $M(S) > 0$. If each input place of a transition t is marked at a marking M, then t is said to be *enabled* at M, in symbols $M[t\rangle$. Otherwise, t is said to be *disabled* at M, in symbols $\neg M[t\rangle$. If an enabled transition t *fires*, a new marking M' is obtained, where $M'(p) = M(p) - 1$ if $p \in {}^\bullet t - t^\bullet$; $M'(p) = M(p) + 1$ if $p \in t^\bullet - {}^\bullet t$; and otherwise, $M'(p) = M(p)$. This fact is denoted by $M[t\rangle M'$. A marking M_k is said to be *reachable* from M if there exists a sequence $\sigma = t_1 t_2 \ldots t_k$ such that $M[t_1\rangle M_1[t_2\rangle \ldots M_{k-1}[t_k\rangle M_k$. This fact is denoted as $M[\sigma\rangle M_k$. The set of markings reachable from M is denoted as $R(N, M)$. (N, M_0) is generally called a *Petri net* or *net system*, where M_0 is called the *initial marking*.

The *incidence matrix* of a net N is a matrix $[N]_{|P| \times |T|}$ such that $[N](p, t) = 1$ if $(t, p) \in F$; $[N](p, t) = -1$ if $(p, t) \in F$; and otherwise $[N](p, t) = 0$. A *P-vector* is a column vector $I : P \to \mathbb{Z}$ indexed by the set of places and a *T-vector* is a column vector $J : T \to \mathbb{Z}$ indexed by the set of transitions. For convenience, a P-vector I is sometimes denoted by $\Sigma_{p \in P} I(p) \bullet p$. For example, $I = (1, 2, 0, 0)^T$ over $P = \{p_1, p_2, p_3, p_4\}$ is written as $I = p_1 + 2p_2$. The same applies to other vectors like J and M. A *P-invariant* is a P-vector I if $I^T \bullet [N] = \mathbf{0}^T$, $I \geq \mathbf{0}$, and $I \neq \mathbf{0}$. $||I|| = \{p \in P | I(p) > 0\}$ is called the *support* of I. A P-invariant is called *minimal* if its support is not a strict superset of the support of any other and the greatest common divisor of its elements is 1. It is well known that $\forall M \in R(N, M_0), I^T \bullet M = I^T \bullet M_0$ if I is a P-invariant of (N, M_0).

A nonempty subset of places S is called a *siphon* (*trap*) if ${}^\bullet S \subseteq S^\bullet$ ($S^\bullet \subseteq {}^\bullet S$). A siphon is said to be *minimal* if it does not contain any other siphon. A *strict minimal siphon* (SMS) is a minimal one that contains no trap.

Given a net $N = (P, T, F)$ and a subset of places S. $N_S = (S, T_S, F_S)$ is defined as the subnet generated by S, where $T_S = {}^\bullet S \cup S^\bullet$, and $F_S \subseteq ((S \times T_S) \cup (T_S \times S)) \cap F$.

In a Petri net (N, M_0), a transition t is *live* if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $M'[t\rangle$. (N, M_0) is *live* if $\forall t \in T$, t is live.

$N = (P, T, F)$ is called a *state machine* if $\forall t \in T$, $|{}^\bullet t| = |t^\bullet| = 1$. It is well known that if a state machine $N = (P, T, F)$ is strongly connected, then P is a minimal siphon and $\Sigma_{p \in P} p$ is a minimal P-invariant.

B. S^3PR and ES^3PR

The following definitions are from [8], [12], [18], and [20].

Definition 1: An $\mathrm{S^3PR}$ is defined as the union of a set of nets $N_1 - N_k$, $N_i = (P_{A_i} \cup P_{R_i} \cup \{p_{0i}\}, T_i, F_i), \forall i \in \mathbb{N}_k, k > 0$, which share common resource places in $\cup_{i=1}^k P_{R_i}$ and satisfy:

1) $\forall i \in \mathbb{N}_k, p_{0i}$ is the *process idle place* of $N_i, P_{A_i} \neq \phi$ (resp. $P_{R_i} \neq \phi$) is the set of *operation* (resp. *resource*) *places* of N_i, $p_{0i} \notin P_{A_i}$, and $(P_{A_i}) \cup \{p_{0i}\}) \cap P_{R_i} = \phi$.
2) $\forall i \in \mathbb{N}_k, \forall p \in P_{A_i}, \forall t_1 \in {}^\bullet p, \forall t_2 \in p^\bullet, \exists r \in P_{R_i}, \{r\} = {}^\bullet t_1 \cap P_{R_i} = t_2^\bullet \cap P_{R_i}; \forall r \in P_{R_i}, {}^{\bullet\bullet} r \cap P_{A_i} = r^{\bullet\bullet} \cap P_{A_i} \neq \phi$, ${}^\bullet r \cap r^\bullet = \phi$; and ${}^{\bullet\bullet} p_{0i} \cap P_{R_i} = p_{0i}^{\bullet\bullet} \cap P_{R_i} = \phi$.
3) $\forall i \neq j, T_i \neq \phi, T_j \neq \phi, T_i \cap T_j = \phi$, and $P_{A_i} \cap P_{A_j} = \phi$.
4) $\forall i \in \mathbb{N}_k, N_i' = (P_{A_i} \cup \{p_{0i}\}, T_i, F_i')$ is a strongly connected state machine (called an $\mathrm{S^2P}$), where N_i' is the subnet of N_i generated by $P_{A_i} \cup \{p_{0i}\}$, i.e., all places in P_{R_i} and the related arcs are removed from N_i.
5) Every circuit of N_i' contains p_{0i}.
6) N_i and N_j are composable when they share a set of common resource places. Every shared place must be a resource one.

For instance, Fig. 1(a) depicts an $\mathrm{S^3PR}$ that is composed of three subnets. $p_{01} - p_{03}$ are process idle places, $p_1 - p_{10}$ are operation places, and $r_1 - r_5$ are resource places in which r_3 and

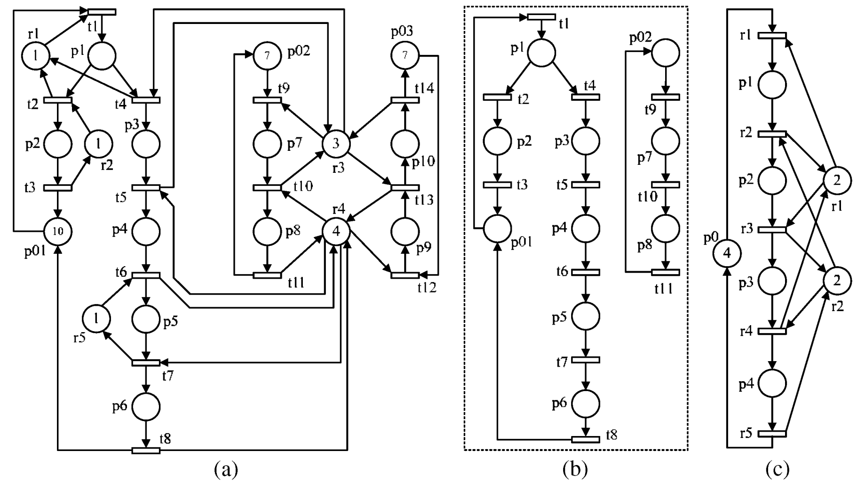

Fig. 1. (a) A marked $\mathrm{S^3PR}$, (b) N'_x, where $x = (r_3, \{p_3, p_7\})$, and (c) a simple marked $\mathrm{S^3PR}$.

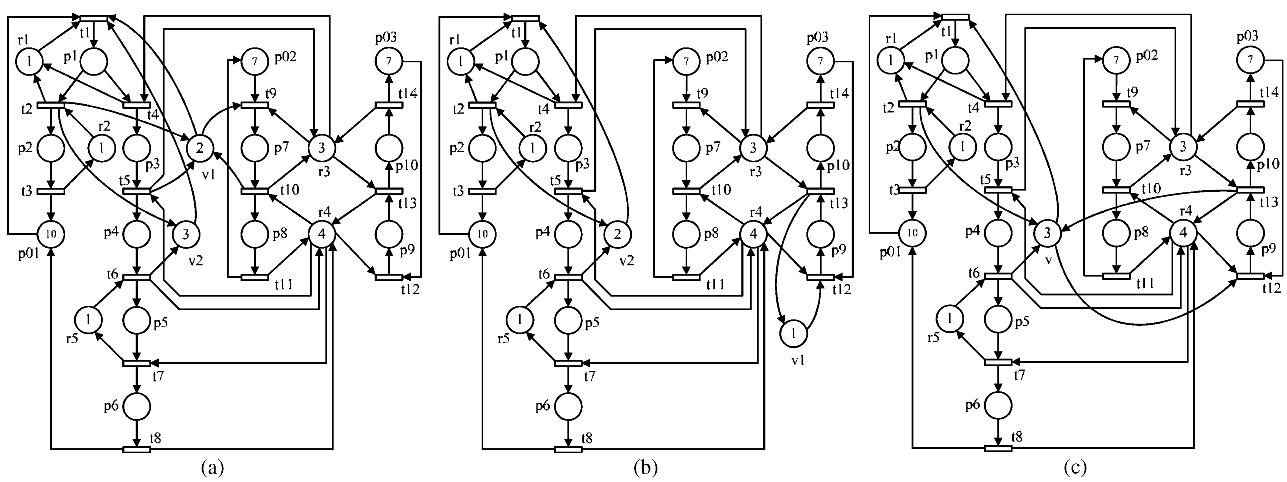

Fig. 2. (a)–(c) Three Type I controlled systems of the Petri net in Fig. 1(a) based on the covers C_1, C_2, and C_3, respectively, where C_1, C_2, and C_3 are in Table II.

r_4 are shared by the three subnets. Fig. 1(b) shows two $\mathrm{S^2P}$ and Fig. 1(c) is another $\mathrm{S^3PR}$. An $\mathrm{S^3PR}$ composed of k subnets is denoted by $N = \circ_{i=1}^{k} N_i = (P_A \cup P_R \cup P_0, T, F)$, where $P_A = \cup_{i=1}^{k} P_{A_i}$, $P_R = \cup_{i=1}^{k} P_{R_i}$, $P_0 = \cup_{i=1}^{k}\{p_{0i}\}$, $T = \cup_{i=1}^{k} T_i$, and $F = \cup_{i=1}^{k} F_i$.

Definition 2: Let $N = (P_A \cup P_R \cup P_0, T, F)$ be an $\mathrm{S^3PR}$. (N, M_0) is called a marked $\mathrm{S^3PR}$ if $\forall p \in P_A$, $M_0(p) = 0$, and $\forall p \in P_R \cup P_0$, $M_0(p) > 0$.

Definition 3: Let $N = (P_A \cup P_R \cup P_0, T, F)$ be an $\mathrm{S^3PR}$, and S be an SMS. $S = S^A \cup S^R$, $S^R = S \cap P_R$, and $S^A = S - S^R$. $\forall r \in P_R$, $H(r) = {}^{\bullet\bullet}r \cap P_A$ is called the set of *holders* of r (i.e., the set of operation places that use r). $\overline{S} = (\cup_{r \in S^R} H(r)) - S$ is called the *complementary set* of S.

For instance, $S_1 = \{r_4, r_5, p_6, p_8, p_9\}$ is an SMS of the net shown in Fig. 1(a). Then, $S_1^A = \{p_6, p_8, p_9\}$, $S_1^R = \{r_4, r_5\}$, $H(r_4) = \{p_4, p_6, p_8, p_9\}$, $H(r_5) = \{p_5\}$, and $\overline{S_1} = \{p_4, p_5\}$. Note that, some papers, such as [20] and [22], use $[S]$ to represent the complementary set of S.

An $\mathrm{S^3PR}$ has the following properties [8], [18].

1) $\forall i \in \mathbb{N}_k$, $p_{0i} + \Sigma_{p \in P_{A_i}} p$ is a minimal P-invariant, and $P_{A_i} \cup \{p_{0i}\}$ is a minimal siphon.
2) $\forall r \in P_R$, $r + \Sigma_{p \in H(r)} p$ is a minimal P-invariant, and $\{r\} \cup H(r)$ is a minimal siphon.
3) Let $S = S^A \cup S^R$ be an SMS. Then, $\forall p \in \overline{S}$, $\exists r \in S^R$, $p \in H(r)$, and $\forall r' \in S^R - \{r\}$, $p \notin H(r')$; $\forall p \in P_0$, $p \notin S$; $|S^R| > 1$; $\Sigma_{p \in S} p + \Sigma_{p \in \overline{S}} p$ is a P-invariant; and $S \cup \overline{S}$ is a siphon.

Definition 4: $N = \circ_{i=1}^{k} N_i = (P_A \cup P_R \cup P_0, T, F)$ is called an $\mathrm{ES^3PR}$ if it satisfies:

1) N is connected and self-loop-free.
2) $P_A = \cup_{i=1}^{k} P_{A_i}$ is the set of *operation places* of N, where $\forall i \neq j$, $P_{A_i} \neq \phi$, $P_{A_j} \neq \phi$, and $P_{A_i} \cap P_{A_j} = \phi$; P_R is the set of *resource places* of N; $P_0 = \cup_{i=1}^{k}\{p_{0i}\}$ is the set of *process idle places* of N; P_A, P_R, and P_0 are mutually disjoint.
3) $T = \cup_{i=1}^{k} T_i$ is the set of *transitions*, where $\forall i \neq j, T_i \neq \phi$, $T_j \neq \phi$, and $T_i \cap T_j = \phi$.
4) $\forall i \in \mathbb{N}_k$, subnet N'_i generated by $P_{A_i} \cup \{p_{0i}\}$ is a strongly connected state machine and every circuit of N'_i contains p_{0i}.
5) $\forall r \in P_R$, there exists a unique minimal P-invariant I_r such that $\{r\} = \|I_r\| \cap P_R$, $\|I_r\| \cap P_A \neq \phi$, $\|I_r\| \cap P_0 = \phi$, and $\forall p \in \|I_r\|$, $I_r(p) = 1$.
6) $P_A = \cup_{r \in P_R}(\|I_r\| - \{r\})$.

The net in Fig. 2(a) may be seen as an $\mathrm{ES^3PR}$. Only one resource can be used at a process stage of an $\mathrm{S^3PR}$, while an

TABLE I
SMS OF THE NET IN FIG. 1(a) AND KP

SMS	places	KR	KP	η_S
S_1	r_4, r_5, p_6, p_8, p_9	r_4	$(r_4, \{p_4\})$	$-t_5+t_7$
S_2	$r_3, r_4, p_4, p_6, p_8, p_{10}$	r_3, r_4	$(r_3, \{p_3, p_7\}), (r_4, \{p_9\})$	$-t_4+t_5-t_9+t_{10}-t_{12}+t_{13}$
S_3	$r_3, r_4, r_5, p_6, p_8, p_{10}$	r_3, r_4	$(r_3, \{p_3, p_7\}), (r_4, \{p_4, p_9\})$	$-t_4+t_7-t_9+t_{10}-t_{12}+t_{13}$

$\mathrm{ES^3PR}$ permits more than one resource to be used at a process stage.

Definition 5: Let $N = (P_A \cup P_R \cup P_0, T, F)$ be an $\mathrm{ES^3PR}$. (N, M_0) is called a marked $\mathrm{ES^3PR}$ if $\forall p \in P_A$, $M_0(p) = 0$, and $\forall p \in P_R \cup P_0$, $M_0(p) > 0$.

Theorem 1 [12]: A marked $\mathrm{ES^3PR}$ is live if no minimal siphons can be emptied.

III. KEY-RESOURCE/OPERATION-PLACE PAIR

This section presents the concept of KP and defines their cover.

Definition 6: Let N be an $\mathrm{S^3PR}$, S be an SMS of N, and N_S be the subnet generated by S. $r \in S$ is called a Key Resource-place (KR) of S if $\exists t \in r^\bullet$, t is a sink in N_S.

Definition 7: Let N be an $\mathrm{S^3PR}$, S be an SMS of N, and r be a KR of S. Then, $K = H(r) \cap \overline{S}$ is called the set of *Key Operation-places* (KO) w.r.t. r and S.

Definition 8: Let N be an $\mathrm{S^3PR}$, S be an SMS of N, r be a KR of S, and K be the set of KO w.r.t. r and S. Then, (r, K) is called a Key-resource/operation-place Pair (KP) of S.

Table I lists all SMS of the net in Fig. 1(a) as well as their KP. For instance, r_3 is a KR of the SMS $S_2 = \{r_3, r_4, p_4, p_6, p_8, p_{10}\}$ since $t_4 \in r_3^\bullet$ and $t_9 \in r_3^\bullet$ are two sink transitions in N_{S_2}. $\{p_3, p_7\}$ is the set of KO w.r.t. r_3 and S_2 since $\{p_3, p_7\} = H(r_3) \cap \overline{S_2}$. Hence, $(r_3, \{p_3, p_7\})$ is a KP of S_2. Similarly, $(r_4, \{p_9\})$ is also a KP of S_2. Note that, two different SMS possibly have the same KR, but the sets of related KO are probably different. For instance, r_4 is a KR of both S_1 and S_2, but the sets of KO are $\{p_4\}$ and $\{p_9\}$, respectively. η_S in Table I is defined in Section VI.

Definition 9: Let N be an $\mathrm{S^3PR}$, $\{S_1, S_2, \ldots, S_m\}$ be the set of SMS of N, and A_i be the set of KP of S_i, $\forall i \in \mathbb{N}_m$. Then, the set C is called a *cover* of KP if the following statements are true:

1) $C \subseteq \cup_{i=1}^m A_i$.
2) For each S_i, $i \in \mathbb{N}_m$, there exist a set of KP $x_1 - x_n \in C$ and a KP $y \in A_i$ such that $r_y = r_{x_1} = \ldots = r_{x_n}$ and $K_y \subseteq \cup_{j=1}^n K_{x_j}$, where $\forall j \in \mathbb{N}_n$, $x_j = (r_{x_j}, K_{x_j})$, and $y = (r_y, K_y)$.
3) For each subset $C' \subset C$, C' does not satisfy Condition 2.

Note that, Condition 3 in Definition 9 guarantees that C is minimal. For instance, $\{(r_4, \{p_4, p_9\})\}$ is a cover of KP for the net in Fig. 1(a). This is because for S_1, there is $(r_4, \{p_4\})$ such that $\{p_4\} \subseteq \{p_4, p_9\}$; for S_2, there is $(r_4, \{p_9\})$ such that $\{p_9\} \subseteq \{p_4, p_9\}$; and for S_3, there is $(r_4, \{p_4, p_9\})$ such that $\{p_4, p_9\} \subseteq \{p_4, p_9\}$. It is possible that an $\mathrm{S^3PR}$ has more than one cover and a resource place occurs in different KP in a cover. For instance, the net in Fig. 1(a) has three covers C_1, C_2, and C_3 that are listed in Table II. Define $\#_C(r)$ as the number of times the resource place r appears in C. For instance, $\#_{C_2}(r_4) = 2$, and $\#_{C_2}(r_1) = \#_{C_2}(r_2) = \#_{C_2}(r3) = \#_{C_2}(r5) = 0$. Identifying C based on Definition 9 is complex. Later we present a method to compute it. For convenience, a cover is generally denoted by $C = \{\alpha, \beta, \ldots, \gamma\}$, where $\forall x \in C$, $x = (r_x, K_x)$.

TABLE II
COVERS OF THE NET IN FIG. 1(a)

cover	KP
C_1	$(r_3, \{p_3, p_7\}), (r_4, \{p_4\})$
C_2	$(r_4, \{p_4\}), (r_4, \{p_9\})$
C_3	$(r_4, \{p_4, p_9\})$

IV. TYPE I CONTROL POLICY

This section presents a new deadlock prevention policy based on the concept of KP and control idea proposed in [8].

Definition 10: Let $N = \circ_{i=1}^k N_i$ be an $\mathrm{S^3PR}$, and N_i' be the $\mathrm{S^2P}$ corresponding to N_i. Given a KP $x = (r_x, K_x)$ of N, let the places in K_x belong to $N'_{x_1} - N'_{x_m}$, respectively. Denote N'_x as the net formed by $N'_{x_1} - N'_{x_m}$, and P_{0x} as the set of process idle places in N'_x. Let a and b be two nodes of N'_x. We say that a is previous to b in N'_x, in symbols $a \prec_{N'_x} b$, if there exists a path from a to b such that this path does not pass any process idle place. Let a and A be a node and a set of nodes of N'_x. We say that a is previous to A in N'_x, in symbols $a \prec_{N'_x} A$, if there is $b \in A$ such that $a \prec_{N'_x} b$. $P_x = \{p | p \prec_{N'_x} K_x^\bullet\}$, $T_x = \{t | t \in P_x^\bullet \wedge t \not\prec_{N'_x} K_x^\bullet\} \cup K_x^\bullet$, $T_x^- = T_x - \{t \in T_x | \exists t' \in T_x : t \neq t' \wedge t \prec_{N'_x} t'\}$, and $T_x^+ = \{t | t \in P_{0x}^\bullet\}$.

Clearly, $K_x \subseteq P_x$ and $a \prec_{N'_x} a$ by Definition 10. $x = (r_3, \{p_3, p_7\})$ is a KP of the net in Fig. 1(a) in which p_3 and p_7 belong to N_1' and N_2', respectively. N_1' and N_2' form N_x' that is in Fig. 1(b). Then, $K_x^\bullet = \{p_3, p_7\}^\bullet = \{t_5, t_{10}\}$; $P_x = \{p_1, p_3, p_7\}$ since $p_1 \prec_{N'_x} t_5$, $p_3 \prec_{N'_x} t_5$, and $p_7 \prec_{N'_x} t_{10}$; $T_x = \{t_2\} \cup \{t_5, t_{10}\} = \{t_2, t_5, t_{10}\}$, where $t_2 \in P_x^\bullet$ and $t_2 \not\prec_{N_x'} K_x^\bullet$ lead to $t_2 \in T_x$; $T_x^- = \{t_2, t_5, t_{10}\}$; and $T_x^+ = \{t_1, t_9\}$. In the net shown in Fig. 1(c), $y = (r_1, \{p_1, p_3\})$ is a KP of SMS $\{r_1, r_2, p_4\}$. Hence, $P_y = \{p_1, p_2, p_3\}$; $T_y = \{t_2, t_4\}$; $T_y^- = T_y - \{t_2\} = \{t_4\}$, where $t_2 \prec_{N'_y} t_4$ results in $t_2 \notin T_y^-$; and $T_y^+ = \{t_1\}$.

Definition 11: Let $N = (P, T, F)$ be an $\mathrm{S^3PR}$ and M_0 be the initial marking of N. Assume that there exists a cover of KP $C = \{\alpha, \beta, \ldots, \gamma\}$ such that $M_0(r) > \#_C(r)$ for each resource place r. Denote $x = (r_x, K_x)$, $\forall x \in C$. Then, (N_V, M_{0V}) is called a *Type I controlled system* of (N, M_0) if it satisfies:

1) $N_V = (P \cup P_V, T, F \cup F_V)$, where $P_V = \{v_\alpha, v_\beta, \ldots, v_\gamma\}$ is the set of *control places*, and $F_V = \cup_{x=\alpha}^{\gamma} \{(t, v_x), (v_x, t') | t \in T_x^-, t' \in T_x^+\}$ is the set of *control arcs*;
2) $\forall p \in P$, $M_{0V}(p) = M_0(p)$; $\forall v \in P_V$, $M_{0V}(v) > 0$; and for each resource place r in N, $\Sigma_{v \in P_V(r)} M_{0V}(v) < M_0(r)$, where $P_V(r) = \{v_x | x \in C \wedge r_x = r\}$.

Note that, in Definition 11 we do not definitely indicate how many tokens are allocated to every control place, but Condition 2 and $M_0(r) > \#_C(r)$ can guarantee that each control place is marked at M_{0V}. Clearly, $\forall x \in C$, ${}^\bullet v_x = T_x^-$, $v_x^\bullet = T_x^+$, and $|P_V(r)| = \#_C(r)$. Fig. 2(a) [resp. (b) and (c)] shows the Type I controlled system of the Petri net in Fig. 1(a) according to C_1 (resp. C_2 and C_3), where C_1, C_2, and C_3 are listed in Table II. For instance, in Fig. 2(b) v_1 and v_2 correspond to $\alpha = (r_4, \{p_9\})$ and $\beta = (r_4, \{p_4\})$, respectively. Because $T_\alpha^- = \{t_{13}\}$, $T_\alpha^+ = \{t_{12}\}$, $T_\beta^- = \{t_2, t_6\}$, and $T_\beta^+ = \{t_1\}$, we have that ${}^\bullet v_1 = \{t_{13}\}$, $v_1^\bullet = \{t_{12}\}$, ${}^\bullet v_2 = \{t_2, t_6\}$, and $v_2^\bullet = \{t_1\}$. $M_{0V}(v_1) = 1$ and $M_{0V}(v_2) = 2$ in Fig. 2(b). Certainly, there are also other markings such as $M_{0V}(v_1) = 1$ and $M_{0V}(v_2) = 1$. However, we should make the sum of tokens in these control places as great as possible such that the controlled system can preserve more legal behavior. Therefore, in Condition 2 of Definition 11, $\Sigma_{v \in P_V(r)} M_{0V}(v)$ should be $M_0(r) - 1$ in general. If $|P_V(r)| = 1$, i.e., there is only one KP in the cover corresponding to r, then $M_{0V}(v) = M_0(r) - 1$. When $|P_V(r)| > 1$ (for instance, cover C_2 in Table II), we may decide the marking of each control place by experiments, i.e., we should select a marking that can permit more legal behavior.

This control policy is similar to that in [8] and [18]. The difference is that it is based on KP, while those in [8] and [18] on (elementary) SMS. It can guarantee that each SMS in the original net is controlled and no new SMS is produced in the new net, thereby making the controlled system live. The proving process of liveness is similar to that of [8, Th. VI.1]. Here, we only need to prove that this policy can guarantee that each SMS of the original system is not emptied. First, we give the following property.

Property 1: Let the control place v_x correspond to KP $x = (r_x, K_x)$. P_x is defined in Definition 10. Then, in the Type I controlled system, $\{v_x\} \cup P_x$ is a minimal siphon, and I is a minimal P-invariant, where $\forall p \in \{v_x\} \cup P_x$, $I(p) = 1$, otherwise, $I(p) = 0$.

Proof: It is obvious since the subnet generated by $\{v_x\} \cup P_x$ is a strongly connected state machine. □

Property 2: Let (N, M_0) be a marked S^{3}PR, and (N_V, M_{0V}) be a Type I controlled system of (N, M_0). Then, each SMS of N is not emptied in (N_V, M_{0V}).

Proof: Let S be an SMS of N, and C be the cover resulting in (N_V, M_{0V}). Then, there are a KP (r, K) of S and a set of KP $(r_{x_1}, K_{x_1}) - (r_{x_n}, K_{x_n})$ in C such that $r = r_{x_1} = \ldots = r_{x_n}$ and $K \subseteq \cup_{i=1}^n K_{x_i}$. $\forall i \in \mathbb{N}_n$, let $x_i = (r_{x_i}, K_{x_i})$, and v_{x_i} be the control place corresponding to x_i. Since $r + \Sigma_{p \in H(r)} p$ is a P-invariant of N, it is a P-invariant of N_V. Hence, $\forall M \in R(N_V, M_{0V})$, $M_{0V}(r) = M(r) + M(H(r))$. Additionally, $H(r) \supseteq \cup_{i=1}^n K_{x_i}$. Hence, $M_{0V}(r) = M(r) + M(\cup_{i=1}^n K_{x_i}) + M(H(r) - \cup_{i=1}^n K_{x_i})$. Since $M_{0V}(r) = M_0(r)$, we can have

$$M\left(\cup_{i=1}^n K_{x_i}\right) = M_0(r) - M\left(H(r) - \cup_{i=1}^n K_{x_i}\right) - M(r). \quad (1)$$

By Property 1, $\forall i \in \mathbb{N}_n$, $v_{x_i} + \Sigma_{p \in P_{x_i}} p$ is a minimal P-invariant of N_V. Hence, $M(v_{x_i}) + M(P_{x_i}) = M_{0V}(v_{x_i})$. Hence, $\Sigma_{i=1}^n M(v_{x_i}) + \Sigma_{i=1}^n M(P_{x_i}) = \Sigma_{i=1}^n M_{0V}(v_{x_i})$. Because $M(\cup_{i=1}^n v_{x_i}) = \Sigma_{i=1}^n M(v_{x_i})$, $M(\cup_{i=1}^n P_{x_i}) \leq \Sigma_{i=1}^n M(P_{x_i})$, and $M_{0V}(\cup_{i=1}^n v_{x_i}) = \Sigma_{i=1}^n M_{0V}(v_{x_i})$, we have that $M(\cup_{i=1}^n v_{x_i}) + M(\cup_{i=1}^n P_{x_i}) \leq M_{0V}(\cup_{i=1}^n v_{x_i})$. In addition, $\cup_{i=1}^n K_{x_i} \subseteq \cup_{i=1}^n P_{x_i}$. We hence have $M(\cup_{i=1}^n v_{x_i}) + M(\cup_{i=1}^n K_{x_i}) \leq M_{0V}(\cup_{i=1}^n v_{x_i})$. Since $M_{0V}(\cup_{i=1}^n v_{x_i}) < M_0(r)$, we can obtain

$$M\left(\cup_{i=1}^n v_{x_i}\right) + M\left(\cup_{i=1}^n K_{x_i}\right) < M_0(r). \quad (2)$$

By (1) and (2), we have $M(\cup_{i=1}^n v_{x_i}) + M_0(r) - M(H(r) - \cup_{i=1}^n K_{x_i}) - M(r) < M_0(r)$, i.e., $M(r) + M(H(r) - \cup_{i=1}^n K_{x_i}) > M(\cup_{i=1}^n v_{x_i})$. Since $M(\cup_{i=1}^n v_{x_i}) \geq 0$, $M(r) + M(H(r) - \cup_{i=1}^n K_{x_i}) > 0$. Since $K \subseteq \cup_{i=1}^n K_{x_i}$, $M(r) + M(H(r) - K) > 0$. Since $(\{r\} \cup (H(r) - K)) \subseteq S$, $M(S) > 0$. □

Lemma 1: Let (N_V, M_{0V}) be a Type I controlled system of the marked S^{3}PR (N, M_0), and σ be a firing sequence of (N_V, M_{0V}). Then, σ is also a firing sequence of (N, M_0).

Proof: It is obvious. (Also see Lemma VI.2 in [8].) □

Lemma 2: Let (N_V, M_{0V}) be a Type I controlled system of the marked S^{3}PR (N, M_0). Then, $\forall t \in T$, $\forall M \in R(N_V, M_{0V})$, $\exists M' \in R(N_V, M)$, $M'[t\rangle$.

Proof: This lemma can be derived by Property 2 and Lemma 1. This proof is the same as that of [8, Lemma VI.4] and thus omitted here. □

By Lemma 2, we can draw the following conclusion

Theorem 2: A Type I controlled system is live.

The three controlled systems in Fig. 2, which are from three different covers, are all live by Theorem 2. Here, deadlocks can be prevented through one control place and five control arcs (see Fig. 2(c)). Please note, if there are more than one KP in a cover such that they have the same KR, then these KP may be combined into one "KP," i.e., if $(r_{x_1}, K_{x_1}) - (r_{x_n}, K_{x_n})$ in a cover satisfy that $r_{x_1} = \ldots = r_{x_n} = r$ and $n > 1$, then the new "KP" $(r, \cup_{i=1}^n K_{x_i})$ takes over these KP in the cover. Based on the new "cover," we carry out the same control with Definition 11, and then the controlled system is also live. Obviously, in this kind of "cover," the number of "KP" is no more than the number of resource types. In other words, we can guarantee that control places are not more than resource places.

V. Type II Control Policy

Just as the policy in [8] and [18], the Type I control policy always sets about "its control" from the first step of each manufacturing process, i.e., each output transition of control places is exactly an output transition of some process idle place. As a consequence, some legal markings are lost. In this section we give another control policy based on the idea of "local control."

Definition 12: Let $N = (P, T, F)$ be an S^{3}PR and M_0 be the initial marking of N. Assume that there exists a cover of KP $C = \{\alpha, \beta, \ldots, \gamma\}$ such that $M_0(r) > \#_C(r)$ for each resource place r. Denote $x = (r_x, K_x)$, $\forall x \in C$. Then, (N_V, M_{0V}) is called a *Type II controlled system* of (N, M_0) if it satisfies:

1) $N_V = (P \cup P_V, T, F \cup F_V)$, where $P_V = \{v_\alpha, v_\beta, \ldots, v_\gamma\}$ is the set of *control places*, and $F_V = \cup_{x=\alpha}^{\gamma} \{(t, v_x), (v_x, t') | t \in K_x^\bullet, t' \in {}^\bullet K_x\}$ is the set of *control arcs*; and
2) $\forall p \in P$, $M_{0V}(p) = M_0(p)$; $\forall v \in P_V$, $M_{0V}(v) > 0$; and for each resource place r in N, $\Sigma_{v \in P_V(r)} M_{0V}(v) < M_0(r)$, where $P_V(r) = \{v_x | x \in C \wedge r_x = r\}$.

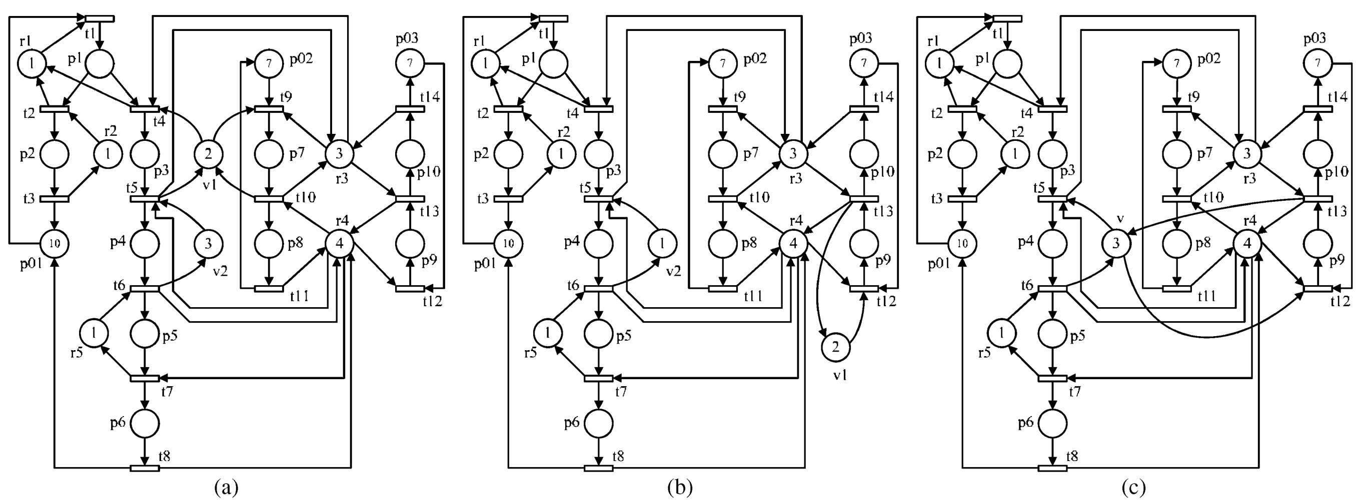

Fig. 3. (a)–(c) Three Type II controlled systems of the Petri net in Fig. 1(a) based on the covers C_1, C_2, and C_3, respectively, where C_1, C_2, and C_3 are in Table II.

Similar to Definition 11, Definition 12 does not definitely set the initial marking for every control place, but Condition 2 and $M_0(r) > \#_C(r)$ can guarantee that each one is marked at M_{0V}. Clearly, $\forall x \in C$, ${}^\bullet v_x = K_x^\bullet$ and $v_x^\bullet = {}^\bullet K_x$. Compared with Type I, Type II uses even fewer arcs. By Definition 12, we can control the system in Fig. 1(a). Fig. 3(a)–(c) shows the controlled systems on the basis of covers C_1, C_2, and C_3, respectively. However, the Petri net in Fig. 3(c) is not live, i.e., Type II cannot always guarantee that the controlled system is live. Fortunately, such system is live if there is no SMS containing any control place in it.

Property 3: Let v be the control place corresponding to (r, K) in a Type II controlled system. Then, $\{v\} \cup K$ is a minimal siphon, and I is a minimal P-invariant, where $\forall p \in \{v\} \cup K$, $I(p) = 1$, otherwise, $I(p) = 0$.

Proof: It is also obvious because the subnet generated by $\{v\} \cup K$ is a strongly connected state machine. □

Property 4: Let (N, M_0) be a marked $\mathrm{S^3PR}$, and (N_V, M_{0V}) be a Type II controlled system. Then, each SMS of N is not emptied in (N_V, M_{0V}).

Proof: It is similar to the proof of Property 2 and thus omitted here. □

Property 5: Let (N, M_0) be a marked $\mathrm{S^3PR}$, and (N_V, M_{0V}) be a Type II controlled system. Then, (N_V, M_{0V}) is a marked $\mathrm{ES^3PR}$.

Proof: Here, each control place may be seen as a special resource place. Therefore, this property is true according to Property 3 and Definitions 4, 5, and 12. □

Theorem 3: A Type II controlled system (N_V, M_{0V}) is live if there is no SMS containing any control place in N_V.

Proof: It is known that any SMS of an $\mathrm{ES^3PR}$ must contain resource places. Therefore, any SMS of a Type II controlled system is exactly the one of the original system if there is no SMS containing any control place in the controlled system. Hence, this theorem is true by Theorem 1 and Properties 4 and 5. □

The following property helps to decide whether a control place belongs to some SMS.

Property 6: Let v be the control place corresponding to KP (r, K). Then, no minimal siphon contains both v and r.

Proof: If a siphon S includes both v and r, then $S - \{v\}$ is also a siphon since ${}^\bullet v \subseteq {}^\bullet r$ and $v^\bullet \subseteq r^\bullet$. Therefore, v and r do not belong to the same minimal siphon. □

It can be verified that there is no SMS containing any control place in the controlled systems in Fig. 3(a) and (b). Hence, the two controlled systems are live. However, deadlocks can take place in Fig. 3(c) because $\{v, r_3, p_4, p_7, p_{10}\}$ is a new SMS. When a Type II controlled system is not live, we can make it live by an iterative control policy [13]. For example, Fig. 4(a) is the controlled system of the Petri net in Fig. 3(c) based on the cover $\{(r_3, \{p_3\})\}$. This Petri net is live. Additionally, Fig. 4(b) and (c) show the Petri nets resulted from the policies in [8] and [18], respectively. Table I lists all SMS. Let v_{S_1}, v_{S_2}, and v_{S_3} be the control places corresponding to S_1, S_2, and S_3, respectively. Note that, the initial marking in Fig. 4(c) is generated by adjusting the *control depth variables*. Before the adjustment, the number of tokens in v_{S_1} and v_{S_2} should be 4 and 6, respectively. Table III lists the number of reachable states, control places, and control arcs of each Petri net in Figs. 2–4. Clearly, Type II control policy is better than Type I.

VI. KPS and Elementary Siphons

In this section, we describe an algorithm that can generate a cover of KP. We first show the relationship between the KP of dependent SMS and those of elementary ones. Li and Zhou [18] define elementary siphons for arbitrary Petri nets, while here we are concerned about elementary SMS of $\mathrm{S^3PR}$ only.

Definition 13: Let S be an SMS of an $\mathrm{S^3PR}$. Then, P-vector λ_S is called the *characteristic P-vector* of S, where $\forall p \in S$, $\lambda_S(p) = 1$; otherwise, $\lambda_S(p) = 0$.

Definition 14: Let S be an SMS of an $\mathrm{S^3PR}$. Then, $\eta_S = [N]^T \bullet \lambda_S$ is called the *characteristic T-vector* of S.

Definition 15: Let N be an $\mathrm{S^3PR}$ with n transitions and m places, and $\Pi = \{S_1, S_2, \ldots, S_k\}$ be the set of SMS of N. Then, $[\lambda]_{k\times m} = [\lambda_{S_1}|\ldots|\lambda_{S_k}]^T$ and $[\eta]_{k\times n} = [\eta_{S_1}|\ldots|\eta_{S_k}]^T$

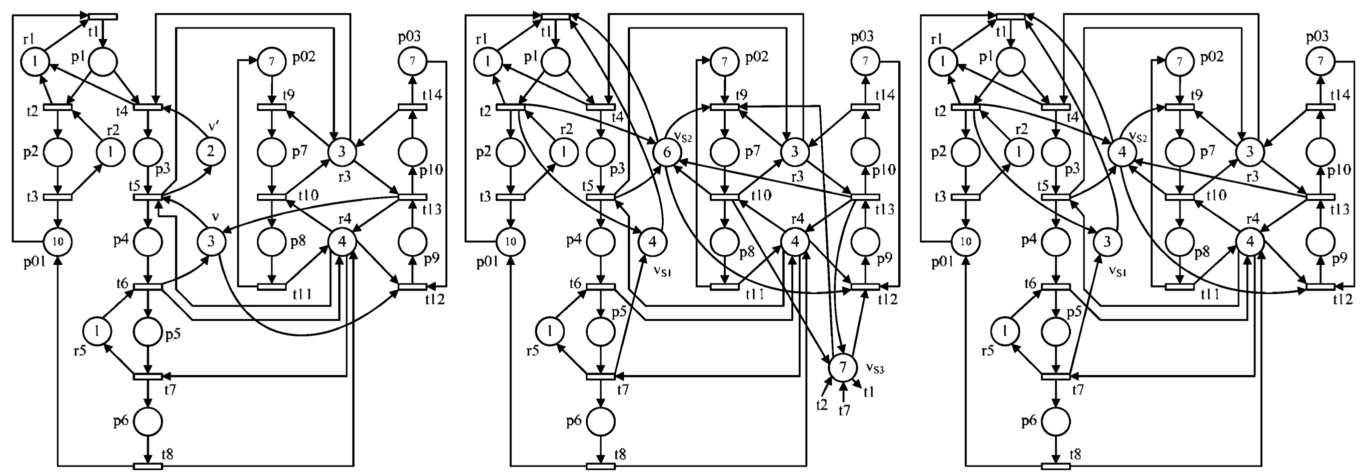

Fig. 4. (a) The Type II controlled system of the Petri net in Fig. 3(c). (b) The controlled system based on the policy in [8]. (c) The controlled system based on the policy in [18].

TABLE III
PERFORMANCE COMPARISON OF THE CONTROLLED SYSTEMS IN FIGS. 2–4

	Type I policy			Type II policy			Policy [8]	Policy [18]
	Fig. 2(a)	Fig. 2(b)	Fig. 2(c)	Fig. 3(a)	Fig. 3(b)	Fig. 4(a)	Fig. 4(b)	Fig. 4(c)
no. reachable states	6,590	5,406	7,346	8,700	7,800	9,724	9,718	7,320
no. control places	2	2	1	2	2	2	3	2
no. control arcs	8	5	5	6	4	6	17	10

are called the *characteristic* P- and T-*vector matrix* of N, respectively.

Clearly, $[\lambda]_{k\times n} = [\lambda]_{k\times m} \bullet [N]_{m\times n}$. Denote $\Pi = \{S_1, \ldots, S_k\}$ as the set of SMS of an S^{3}PR.

Definition 16: Let $\eta_{S_\alpha} - \eta_{S_\gamma}$ be a linearly independent maximal set of $[\eta]$. Then, $\Pi_E = \{S_\alpha, S_\beta, \ldots, S_\gamma\}$ is called the set of *elementary SMS*.

Definition 17: $S \in \Pi - \Pi_E$ is called a *strongly dependent SMS* if $\eta_S = \Sigma_{S_i\in\Pi_E} a_i \bullet \eta_{S_i}$, where $a_i \geq 0$.

Definition 18: $S \in \Pi - \Pi_E$ is called a *weakly dependent SMS* if there are two nonempty sets A and B such that $A \subseteq \Pi_E$, $B \subseteq \Pi_E$, $A \cap B = \phi$, and $\eta_S = \Sigma_{S_i\in A} a_i \bullet \eta_{S_i} - \Sigma_{S_i\in B} a_i \bullet \eta_{S_i}$, where $a_i > 0$.

Table I lists the characteristic T-vector of each SMS. Obviously, $\{S_1, S_2\}$ may be as the set of elementary SMS since $\eta_{S_3} = \eta_{S_1} + \eta_{S_2}$. Under this case, S_3 is a strongly dependent SMS. $\{S_1, S_3\}$ may also be as the set of elementary SMS since $\eta_{S_2} = \eta_{S_3} - \eta_{S_1}$. Under this case, S_2 is a weakly dependent SMS.

Definition 19: Let Π_W be the set of weakly dependent SMS. $\Pi_{EE} = \Pi_E \cup \Pi_W$ is called the set of *extended elementary SMS*.

Property 7 [18]: Let S be a strongly dependent SMS with $\eta_S = \Sigma_{S_i\in\Pi_E} a_i \bullet \eta_{S_i}$. Then, $a_i = 0$ or $a_i = 1$.

Property 8: $\forall S \in \Pi$, $\eta_S = \Sigma_{S_i\in\Pi_{EE}} a_i \bullet \eta_{S_i}$, where $a_i = 0$ or $a_i = 1$.

Proof: If S is a strongly dependent SMS, then this conclusion is true by Property 7. If S is a weakly dependent SMS or an elementary SMS, i.e., $S \in \Pi_{EE}$, then $\eta_S = \eta_S$. □

Definition 19 and Property 8 show that the characteristic T-vector of each SMS can be linearly represented by extended elementary SMS with nonnegative coefficients (0 and 1).

It is well known that in an S^{3}PR: 1) A transition that is an output one of a process idle place has two input arcs and one output arc; 2) A transition that is an input one of a process idle place has one input arc and two output arcs; and 3) any other transition has two input arcs and two output arcs. In addition, it is impossible that a transition in the subnet generated by an SMS has two input arcs. This is because if it had two input places in the subnet, i.e., a resource place and an operation place, then the set of places, which is generated by deleting the operation place from the SMS, would also be a siphon. This contradicts the minimality of the SMS. Therefore, for each transition in the subnet generated by an SMS, only one of the following three statements is true: 1) it has an input arc only, i.e., firing it decreases the number of tokens in the SMS by 1; 2) it has one input arc and an output arc, i.e., firing it does not change the number of tokens in the SMS; and 3) it has one input arc and two output arcs, i.e., firing it increases the number of tokens in the SMS by 1. Thus, we have the following conclusion.

Property 9: For each SMS S, each entry of η_S is either -1, or 0, or 1.

Property 10: $\forall S \in \Pi, \forall p \in \overline{S}, \exists S' \in \Pi_{EE}, p \in \overline{S'}$.

Proof: By Property 9, we have that $\forall p \in \overline{S}$, $\eta_S(t) = 0$ or $\eta_S(t) = -1$, where $\{t\} = {}^\bullet p$. In addition, $\exists S_1 - S_k \in \Pi_{EE}$, $\eta_S = \sum_{i=1}^{k} \eta_{S_i}$ (by Property 8). Hence, it is impossible that $\forall i \in \mathbb{N}_k$, $\eta_{S_i}(t) = 1$. Hence, $\exists i \in \mathbb{N}_k$, $\eta_{S_i}(t) = 0$ or $\eta_{S_i}(t) = -1$, i.e., $p \in \overline{S_i}$. □

Property 11: Let $S \in \Pi - \Pi_{EE}$. Then, for each KO p of S, there exists $S' \in \Pi_{EE}$ such that p is a KO of S'.

Proof: It is obvious by Property 10. □

Define Ω as the set of KP of extended elementary SMS.

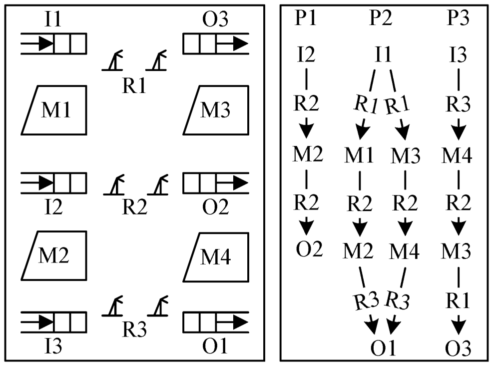

Fig. 5. An FMS.

Theorem 4: Let $S \in \Pi - \Pi_{EE}$. Then, for each KP (r, K) of S, there is a set of KP $x_1 - x_n \in \Omega$ such that $r_{x_1} = \ldots = r_{x_n} = r$ and $K \subseteq \cup_{i=1}^{n} K_{x_i}$, where $\forall i \in \mathbb{N}_n, x_i = (r_{x_i}, K_{x_i})$.

Proof: It is obvious by Property 11. □

Therefore, we can make use of Ω to generate a cover of KP. Assume that Ω has been produced before executing the following algorithm. Additionally, in the following algorithm, each KP x is denoted by (r_x, K_x).

Algorithm 1

Input: Ω

Output: a cover C of KP

Step 1: $C := \phi$

Step 2: **while** $\Omega \neq \phi$ **do**

2.1 Get a KP ω from Ω

2.2 **if** {for each $S \in \Pi$, there exist a set of KP $x_1 - x_n \in ((\Omega \cup C) - \{\omega\})$ and a KP y of S such that $r_y = r_{x_1} = \ldots = r_{x_n}$ and $K_y \subseteq \cup_{i=1}^{n} K_{x_i}$} **then** $\{\Omega := \Omega - \{\omega\}\}$ **else** $\{C := C \cup \{\omega\}; \Omega := \Omega - \{\omega\}\}$

Obviously, in this algorithm the number of times executing Step 2.2 equals $|\Omega|$, and the time complexity on checking the condition in Step 2.2 is $O(|\Pi|)$. Therefore, the time complexity of this algorithm is $O(|\Pi| \times |\Omega|)$.

VII. An Illustrative Example

In this section, we will illustrate our proposed policies by a well-known example.

The FMS in Fig. 5 is similar to [8] and [18]. There are four machines $M_1 - M_4$, three groups of robots $R_1 - R_3$, three input buffers $I_1 - I_3$, and three output buffers $O_1 - O_3$. We assume that there are enough raw parts in the input buffers and the parts in the output buffers are always moved away in time. In other words, these buffers are not considered in our models. Each machine may process two parts at the same time. The number of robots in each group is 2. Each robot can hold one part at a time. R_1 can move parts from I_1 to M_1, I_1 to M_3, and M_3 to O_3. R_2 is in charge of part movement from I_2 to M_2, M_2 to O_2, M_1 to M_2, M_3 to M_4, and M_4 to M_3. R_3 handles part movements from M_2 to O_1, M_4 to O_1, and I_3 to M_4. The FMS can manufacture three types of products and manufacturing processes are shown in Fig. 5.

The Petri net in Fig. 6(a) models this FMS and its SMS are listed in Table IV in which extended elementary SMS are marked by *. The Petri net is not live due to these SMS. The KP of each SMS are also listed in Table IV. We can generate a cover: $\{(R_1, \{p_4\}), (R_2, \{p_1, p_6, p_9\}), (R_3, \{p_{12}\}), (M_3, \{p_8\})\}$. By this cover, we can obtain types I and II controlled systems that are shown in Fig. 6(b) and (c), respectively. In Fig. 6(b) and (c), control places $v_1 - v_4$ correspond to $(R_1, \{p_4\})$, $(R_2, \{p_1, p_6, p_9\})$, $(R_3, \{p_{12}\})$, and $(M_3, \{p_8\})$, respectively. By Theorem 2, the Petri net in Fig. 6(b) is live. It can be verified that there is no SMS containing any control place in the net in Fig. 6(c). Thus, the Petri net in Fig. 6(c) is also live by Theorem 3. Hence, deadlocks can be prevented only by 4 control places and 13 control arcs.

VIII. Experiments

In this paper, we propose KP, and apply them to the deadlock prevention for S^3PR. Our idea is to prevent deadlocks through restricting the use of some resources. This can limit the number of control places up to the number of resource ones. We adopt a conservative measure: if a type of resources is used by a number of manufacturing processes, we only allow a part of these processes to use a part of these resources. That is, we always leave a part of these resources for a (or several) fixed process (-es). As a result, we propose the concept of KP. Policy II reflects this idea on the whole. Policy II can guarantee that all the original SMS are controlled, but sometimes it leads to some new SMS that include control places. In other words, it does not always result in a live controlled system. Thus, we give Policy I that combines Policy II with the method proposed in [8] and makes more legal behavior lost consequently. Indeed, we can also utilize the iterative control to prevent deadlocks on the basis of Policy II [Fig. 4(a) shows such an example]. This kind of policy is possibly better than Policy I. We believe that the number of new SMS is relatively small because the arcs are relatively few (it is possible that the fewer arcs, the fewer circular waits), i.e., the supervisor is not large either. At the same time, the iterative control possibly obtains more legal behavior than Policy I.

In what follows, we will compare the two policies with others through some experimental results. [22] and [27] summarize many Petri-net-based deadlock prevention policies. Here, some of them are applied to the example in Section VII. The related experimental results are summarized in Table V. Note that these experiments are done with the aid of INA [30]. The Petri net in Fig. 6(a) has 449 160 reachable states including 309 deadlock states. In fact, it has 34 631 bad states and 414 529 good ones. Once a system reaches a bad state, it will enter a deadlock finally. In Table V, each performance measure is obtained by $x/414529$, where x represents the number of reachable markings of each controlled system.

For the control policy in [8], eighteen control places have to be added and their output arcs must point to the source transitions of the original net. This supervisor is shown in Table VI in which $v_1 - v_{18}$ correspond to $S_1 - S_{18}$, respectively.

For the control policy in [18], six control places are needed since there are only six elementary SMS. Table VII shows the corresponding supervisor. In Table VII, $v_1 - v_6$ correspond to S_1, S_4, S_{10}, S_{16}, S_{17}, and S_{18}, respectively. Under the initial

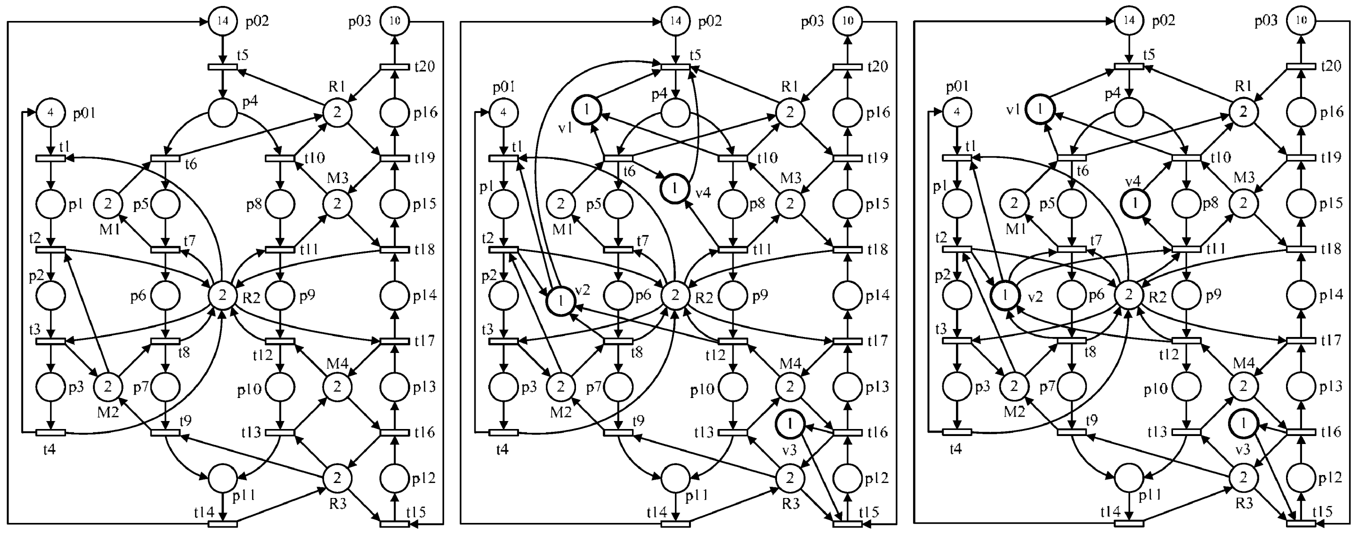

Fig. 6. (a) The Petri net modeling the FMS in Fig. 5. (b) Type I controlled system. (c) Type II controlled system.

TABLE IV
SMS OF THE NET IN FIG. 6(a) AND KPS

SMS	places	KP
S_1^*	R_3, M_4, p_{11}, p_{13}	**$(R_3, \{p_{12}\}), (M_4, \{p_{10}\})$**
S_2	$R_1, R_2, R_3, M_1, M_2, M_3, M_4, p_3, p_{11}, p_{16}$	$(R_1, \{p_4\}), (R_2, \{p_1, p_6, p_9, p_{14}\}), (R_3, \{p_{12}\})$
S_3	$R_2, R_3, M_2, M_3, M_4, p_3, p_{11}, p_{15}$	$(R_2, \{p_1, p_6, p_9, p_{14}\}), (R_3, \{p_{12}\}), (M_3, \{p_8\})$
S_4^*	$R_2, R_3, M_2, M_4, p_3, p_{11}, p_{14}$	**$(R_2, \{p_1, p_6, p_9\}), (R_3, \{p_{12}\})$**
S_5	$R_1, R_2, M_1, M_2, M_3, M_4, p_3, p_7, p_{10}, p_{16}$	$(R_1, \{p_4\}), (R_2, \{p_1, p_6, p_9, p_{14}\}), (M_4, \{p_{13}\})$
S_6	$R_2, M_2, M_3, M_4, p_3, p_7, p_{10}, p_{15}$	$(R_2, \{p_1, p_6, p_9, p_{14}\}), (M_3, \{p_8\}), (M_4, \{p_{13}\})$
S_7	$R_2, M_2, M_4, p_3, p_7, p_{10}, p_{14}$	$(R_2, \{p_1, p_6, p_9\}), (M_4, \{p_{13}\})$
S_8	$R_1, R_2, M_1, M_2, M_3, p_3, p_7, p_9, p_{16}$	$(R_1, \{p_4\}), (R_2, \{p_1, p_6, p_{14}\})$
S_9	$R_2, M_2, M_3, p_3, p_7, p_9, p_{15}$	$(R_2, \{p_1, p_6, p_{14}\}), (M_3, \{p_8\})$
S_{10}^*	$R_2, M_2, p_3, p_7, p_9, p_{14}$	**$(R_2, \{p_1, p_6\})$**
S_{11}	$R_1, R_2, R_3, M_1, M_3, M_4, p_1, p_3, p_6, p_{11}, p_{16}$	$(R_1, \{p_4\}), (R_3, \{p_{12}\})$
S_{12}	$R_1, R_2, M_1, M_3, M_4, p_1, p_3, p_6, p_{10}, p_{16}$	$(R_1, \{p_4\}), (M_4, \{p_{13}\})$
S_{13}	$R_2, R_3, M_3, M_4, p_1, p_3, p_6, p_{11}, p_{15}$	$(R_3, \{p_{12}\}), (M_3, \{p_8\})$
S_{14}	$R_2, M_3, M_4, p_1, p_3, p_6, p_{10}, p_{15}$	$(M_3, \{p_8\}), (M_4, \{p_{13}\})$
S_{15}	$R_2, R_3, M_4, p_1, p_3, p_6, p_{11}, p_{14}$	$(R_2, \{p_9\}), (R_3, \{p_{12}\})$
S_{16}^*	$R_2, M_4, p_1, p_3, p_6, p_{10}, p_{14}$	**$(R_2, \{p_9\}), (M_4, \{p_{13}\})$**
S_{17}^*	$R_1, R_2, M_1, M_3, p_1, p_3, p_6, p_9, p_{16}$	**$(R_1, \{p_4\}), (R_2, \{p_{14}\})$**
S_{18}^*	$R_2, M_3, p_1, p_3, p_6, p_9, p_{15}$	**$(R_2, \{p_{14}\}), (M_3, \{p_8\})$**

marking, the number of tokens in $v_1 - v_6$ is 3, 7, 2, 3, 7, and 3, respectively. Note that this initial marking is generated by adjusting the *control depth variables*, otherwise, the controlled system is not live (before the adjustment, the number of tokens in $v_1 - v_6$ should be 3, 7, 3, 3, 7, and 3, respectively). This supervisor is smaller than that generated by the policy in [8], but the number of reachable states decreases due to the adjustment of the initial marking.

To avoid the adjustment of control depth variables, Li and Zhou [20] give an improved deadlock prevention policy on the basis of the policies in [8] and [18]. First of all, all elementary SMSs are controlled through adding monitors for them. Next, check whether other dependent SMS are controlled and obtain the dependent SMS that are not controlled. Finally, add control places for those uncontrolled SMSs. For instance, after six control places are added for S_1, S_4, S_{10}, S_{16}, S_{17}, and S_{18}, two dependent SMS, S_7 and S_9, are not controlled. Hence, two additional control places are needed. This policy has a good effect: it can permit much more legal behavior with a not-too-large supervisor. Table VIII shows all control places and arcs in which $v_1 - v_8$ correspond to S_1, S_4, S_{10}, S_{16}, S_{17}, S_{18}, S_7, and S_9, respectively.

From Table V we can see that reachable states produced by the above three policies are more than those in Fig. 6(b) but less than those in Fig. 6(c). However, their supervisors are obviously larger than those in Figs. 6(b) and (c).

The policy in [4] is to prevent deadlocks by decreasing the number of tokens in process idle places, and needs no additional control places and arcs. According to this policy, the number of tokens in p_{01}, p_{02}, and p_{03} in the initial marking becomes 2, 1, and 2, respectively. It thus maintains only a small portion of legal behavior. Park and Reveliotis [26] present a polynomial-time algorithm to prevent the deadlock. This policy is based on the resource ordering. The number of control places is equal to the number of resource ones. However, much legal behavior is lost. For instance, there are only 5,141 reachable marking under

TABLE V
PERFORMANCE COMPARISON

	Policy [8]	Policy [18]	Policy [20]	Policy [26]	Policy [4]	Policy [28]	Policy [1, 48]	Policy [34, 36]	Type I Fig. 6(b)	Type II Fig. 6(c)
no. reachable states	133,350	88,846	133,350	5,141	1,495	195,842	414,529	414,529	41,834	188,955
no. control places	18	6	8	7	0	2	18	23	4	4
no. control arcs	106	32	46	33	0	8	104	149	13	13
performance (%)	32.17	21.43	32.17	1.24	0.36	47.24	100	100	10.09	45.58

TABLE VI
SUPERVISOR GENERATED BY POLICY IN [8]

v	$M_0(v)$	$^{\bullet}v$	$v^{\bullet}$
v_1	3	$\{t_6, t_{13}, t_{16}\}$	$\{t_5, t_{15}\}$
v_2	13	$\{t_3, t_9, t_{13}, t_{19}\}$	$\{t_1, t_5, t_{15}\}$
v_3	9	$\{t_3, t_9, t_{13}, t_{18}\}$	$\{t_1, t_5, t_{15}\}$
v_4	7	$\{t_3, t_9, t_{13}, t_{17}\}$	$\{t_1, t_5, t_{15}\}$
v_5	11	$\{t_3, t_8, t_{12}, t_{19}\}$	$\{t_1, t_5, t_{15}\}$
v_6	7	$\{t_3, t_8, t_{12}, t_{18}\}$	$\{t_1, t_5, t_{15}\}$
v_7	5	$\{t_3, t_8, t_{12}, t_{17}\}$	$\{t_1, t_5, t_{15}\}$
v_8	9	$\{t_3, t_8, t_{11}, t_{19}\}$	$\{t_1, t_5, t_{15}\}$
v_9	5	$\{t_3, t_8, t_{11}, t_{18}\}$	$\{t_1, t_5, t_{15}\}$
v_{10}	3	$\{t_3, t_8, t_{10}\}$	$\{t_1, t_5\}$
v_{11}	11	$\{t_7, t_{13}, t_{19}\}$	$\{t_5, t_{15}\}$
v_{12}	9	$\{t_7, t_{12}, t_{19}\}$	$\{t_5, t_{15}\}$
v_{13}	7	$\{t_6, t_{13}, t_{18}\}$	$\{t_5, t_{15}\}$
v_{14}	5	$\{t_6, t_{12}, t_{18}\}$	$\{t_5, t_{15}\}$
v_{15}	5	$\{t_6, t_{13}, t_{17}\}$	$\{t_5, t_{15}\}$
v_{16}	3	$\{t_6, t_{12}, t_{17}\}$	$\{t_5, t_{15}\}$
v_{17}	7	$\{t_7, t_{11}, t_{19}\}$	$\{t_5, t_{15}\}$
v_{18}	3	$\{t_6, t_{11}, t_{18}\}$	$\{t_5, t_{15}\}$

TABLE VII
SUPERVISOR GENERATED BY POLICY IN [18]

v	$M_0(v)$	$^{\bullet}v$	$v^{\bullet}$
v_1	3	$\{t_6, t_{13}, t_{16}\}$	$\{t_5, t_{15}\}$
v_2	7	$\{t_3, t_9, t_{13}, t_{17}\}$	$\{t_1, t_5, t_{15}\}$
v_3	2	$\{t_3, t_8, t_{10}\}$	$\{t_1, t_5\}$
v_4	3	$\{t_6, t_{12}, t_{17}\}$	$\{t_5, t_{15}\}$
v_5	7	$\{t_7, t_{11}, t_{19}\}$	$\{t_5, t_{15}\}$
v_6	3	$\{t_6, t_{11}, t_{18}\}$	$\{t_5, t_{15}\}$

TABLE VIII
SUPERVISOR GENERATED BY POLICY IN [20]

v	$M_0(v)$	$^{\bullet}v$	$v^{\bullet}$
v_1	3	$\{t_6, t_{13}, t_{16}\}$	$\{t_5, t_{15}\}$
v_2	7	$\{t_3, t_9, t_{13}, t_{17}\}$	$\{t_1, t_5, t_{15}\}$
v_3	3	$\{t_3, t_8, t_{10}\}$	$\{t_1, t_5\}$
v_4	3	$\{t_6, t_{12}, t_{17}\}$	$\{t_5, t_{15}\}$
v_5	7	$\{t_7, t_{11}, t_{19}\}$	$\{t_5, t_{15}\}$
v_6	3	$\{t_6, t_{11}, t_{18}\}$	$\{t_5, t_{15}\}$
v_7	5	$\{t_3, t_8, t_{12}, t_{17}\}$	$\{t_1, t_5, t_{15}\}$
v_8	5	$\{t_3, t_8, t_{11}, t_{18}\}$	$\{t_1, t_5, t_{15}\}$

the following resource ordering: $o(R_1) = o(R_3) = o(M_1) = o(M_4) = 2$ and $o(R_2) = o(M_2) = o(M_3) = 3$. Table IX shows the supervisor computed according to this resource ordering.

The policy in [28] appears to achieve a better result. However, this policy needs to preallocate some resources. A modified Petri net can reflect the preallocation. For instance, in Fig. 6(a), the postset of M_3 (resp. M_4) is changed into $\{t_{10}, t_{17}\}$ (resp. $\{t_{11}, t_{15}\}$). Table X is the related supervisor. It is not easy to decide which resources need to be preallocated, however.

TABLE IX
SUPERVISOR GENERATED BY POLICY IN [26]

v	$M_0(v)$	$^{\bullet}v$	$v^{\bullet}$
v_1	2	$\{t_6, t_{10}, t_{20}\}$	$\{t_5, t_{19}\}$
v_2	2	$\{t_4, t_8, t_{12}, t_{18}\}$	$\{t_1, t_5, t_{15}\}$
v_3	2	$\{t_{14}, t_{16}\}$	$\{t_9, t_{12}, t_{15}\}$
v_4	2	$\{t_7, t_{10}\}$	$\{t_5\}$
v_5	2	$\{t_3, t_9, t_{10}\}$	$\{t_1, t_5\}$
v_6	2	$\{t_{11}, t_{19}\}$	$\{t_{10}, t_{15}\}$
v_7	2	$\{t_{13}, t_{17}\}$	$\{t_{12}, t_{15}\}$

TABLE X
SUPERVISOR GENERATED BY POLICY IN [28]

v	$M_0(v)$	$^{\bullet}v$	$v^{\bullet}$
v_1	3	$\{t_3, t_8\}$	$\{t_1, t_7\}$
v_2	3	$\{t_{11}, t_{17}\}$	$\{t_{10}, t_{15}\}$
M_3	2	$\{t_{11}, t_{19}\}$	$\{t_{10}, t_{17}\}$
M_4	2	$\{t_{13}, t_{17}\}$	$\{t_{11}, t_{15}\}$

TABLE XI
SUPERVISOR GENERATED BY POLICIES IN [1] AND [48]

v	$M_0(v)$	$^{\bullet}v$	$v^{\bullet}$
v_1	3	$\{t_{13}, t_{16}\}$	$\{t_{12}, t_{15}\}$
v_2	13	$\{t_3, t_9, t_{13}, t_{19}\}$	$\{t_1, t_5, t_{15}\}$
v_3	9	$\{t_3, t_9, t_{13}, t_{18}\}$	$\{t_1, t_7, t_{10}, t_{15}\}$
v_4	7	$\{t_3, t_9, t_{13}, t_{17}\}$	$\{t_1, t_7, t_{11}, t_{15}\}$
v_5	11	$\{t_3, t_8, t_{12}, t_{19}\}$	$\{t_1, t_5, t_{16}\}$
v_6	7	$\{t_3, t_8, t_{12}, t_{18}\}$	$\{t_1, t_7, t_{10}, t_{16}\}$
v_7	5	$\{t_3, t_8, t_{12}, t_{17}\}$	$\{t_1, t_7, t_{11}, t_{16}\}$
v_8	9	$\{t_3, t_8, t_{11}, t_{19}\}$	$\{t_1, t_5, t_{17}\}$
v_9	5	$\{t_3, t_8, t_{11}, t_{18}\}$	$\{t_1, t_7, t_{10}, t_{17}\}$
v_{10}	3	$\{t_3, t_8\}$	$\{t_1, t_7\}$
v_{11}	11	$\{t_7, t_{13}, t_{19}\}$	$\{t_5, t_{15}\}$
v_{12}	9	$\{t_7, t_{12}, t_{19}\}$	$\{t_5, t_{16}\}$
v_{13}	7	$\{t_{13}, t_{18}\}$	$\{t_{10}, t_{15}\}$
v_{14}	5	$\{t_{12}, t_{18}\}$	$\{t_{10}, t_{16}\}$
v_{15}	5	$\{t_{13}, t_{17}\}$	$\{t_{11}, t_{15}\}$
v_{16}	3	$\{t_{12}, t_{17}\}$	$\{t_{11}, t_{16}\}$
v_{17}	7	$\{t_7, t_{11}, t_{19}\}$	$\{t_5, t_{17}\}$
v_{18}	3	$\{t_{11}, t_{18}\}$	$\{t_{10}, t_{17}\}$

The policy in [48], based on maximal perfect-resource-transition circuit (MPC), preserves much more legal behavior than others. Generally, this policy is suboptimal. Just like SMS, however, there are many MPC in a Petri net. They can potentially make the supervisor very complex. It happens that the supervisor produced by this policy is optimal for this example. Table XI shows this supervisor. The policy in [1] also needs to control all SMS. It also happens that the policies in [1] and [48] result in the same liveness-enforcing supervisor for this example.

TABLE XII
SUPERVISOR GENERATED BY POLICIES IN [34] AND [36]

v	$M_0(v)$	$^\bullet v$	$v^\bullet$
v_1	3	$\{t_3, t_8\}$	$\{t_1, t_7\}$
v_2	5	$\{t_{13}, t_{17}\}$	$\{t_{11}, t_{15}\}$
v_3	5	$\{t_{12}, t_{18}\}$	$\{t_{10}, t_{16}\}$
v_4	7	$\{t_3, t_9, t_{13}, t_{17}\}$	$\{t_2, t_7, t_{11}, t_{15}\}$
v_5	9	$\{t_3, t_9, t_{11}, t_{13}, t_{18}\}$	$\{t_1, t_8, t_{10}, t_{12}, t_{15}\}$
v_6	11	$\{t_7, t_{13}, t_{19}\}$	$\{t_5, t_{15}\}$
v_7	13	$\{t_3, t_7, t_9, t_{11}, t_{13}, t_{19}\}$	$\{t_1, t_5, t_8, t_{12}, t_{15}\}$
v_8	3	$\{t_{12}, t_{17}\}$	$\{t_{11}, t_{16}\}$
v_9	5	$\{t_3, t_8, t_{12}, t_{17}\}$	$\{t_2, t_7, t_{11}, t_{16}\}$
v_{10}	7	$\{t_7, t_{11}, t_{19}\}$	$\{t_5, t_{17}\}$
v_{11}	9	$\{t_3, t_9, t_{11}, t_{13}, t_{18}\}$	$\{t_2, t_7, t_{10}, t_{12}, t_{15}\}$
v_{12}	13	$\{t_3, t_9, t_{11}, t_{13}, t_{19}\}$	$\{t_2, t_5, t_{12}, t_{15}\}$
v_{13}	3	$\{t_{11}, t_{18}\}$	$\{t_{10}, t_{17}\}$
v_{14}	5	$\{t_3, t_{12}, t_{17}\}$	$\{t_1, t_{11}, t_{16}\}$
v_{15}	7	$\{t_3, t_9, t_{13}, t_{17}\}$	$\{t_1, t_8, t_{11}, t_{15}\}$
v_{16}	9	$\{t_3, t_8, t_{11}, t_{19}\}$	$\{t_2, t_5, t_{17}\}$
v_{17}	3	$\{t_{13}, t_{16}\}$	$\{t_{12}, t_{15}\}$
v_{18}	5	$\{t_3, t_8, t_{11}, t_{18}\}$	$\{t_2, t_7, t_{10}, t_{17}\}$
v_{19}	7	$\{t_3, t_9, t_{13}, t_{17}\}$	$\{t_1, t_7, t_{12}, t_{15}\}$
v_{20}	9	$\{t_3, t_7, t_{11}, t_{19}\}$	$\{t_1, t_5, t_{17}\}$
v_{21}	5	$\{t_3, t_{11}, t_{18}\}$	$\{t_1, t_{10}, t_{17}\}$
v_{22}	7	$\{t_{13}, t_{18}\}$	$\{t_{10}, t_{15}\}$
v_{23}	9	$\{t_7, t_{12}, t_{19}\}$	$\{t_5, t_{16}\}$

Uzam obtains many results on optimal control policies [32]–[36]. These policies are mainly based on the reachability graph of a Petri net. In general, an optimal control policy leads to a complex supervisor. Table XII is the supervisor resulting from the policies in [34] and [36] that makes use of the iterative control and reachability graph.

From the experimental results we can see that Policy II has advantages over Policy I. Policy II can relatively permit more legal behavior while they make the supervisor smaller than some famous ones in [8] and [18]. Although some policies [4], [26] can generate a smaller supervisor or utilize a more efficient polynomial-time algorithm to produce the supervisor than our policies do, they lose much more legal behavior. Optimized policies [34], [36], [48] can preserve most of legal behavior, but they usually result in a large supervisor and need the reachability graph technique and integer programming method that are computationally prohibitive. Some experiments of S^3PR show that reachability set grows much larger than the number of SMS. Note that the latter is related to only net structure while the former relates to both net structure and initial marking. For instance, we add a new manufacturing process: $R_2 \rightarrow M_1 \rightarrow R_3$. The new Petri net has 30 SMS and more than 2 000 000 states, and the Petri net [Fig. 6(a)] has only 18 SMS and 449 160 states. By comparison, state explosion seems much more severe than SMS explosion does.

One important index to evaluate a supervisor is the number of good states that are preserved in a controlled system. Another is the size of a supervisor. Places may be seen as the variables in the control programming (they consume memory space) and transitions as events. Then, input arcs of a transition represent the deciding conditions before the corresponding event is executed. At the same time, input and output arcs of a transition represent the operations that modify the variable values when the corresponding event is executed. Hence, the more arcs, the more runtime. It is obvious that the policies proposed in this paper have advantages in this aspect.

IX. CONCLUSION

This paper presents KP, and two deadlock prevention policies are proposed based on this new structural concept. Because they can guarantee that the supervisor is small and efficient, they are better suited to complex S^3PR that often have more job types and resource types. However, there are several drawbacks in the two policies. First, they are restricted by the number of resources. When $M_0(r) \leq \#_C(r)$, i.e., the quantity of resource type r is not greater than the number of times the resource place r appears in a cover, our policies are not suitable. Note that if a resource place appears more than once in the cover, then the corresponding KP can be merged into one, i.e., our policies must require that $M_0(r) > 1$ if r appears in the cover. Second, our policies are not optimal. That is to say, some legal markings are lost in the controlled systems. Third, the computational complexity is high although it is comparable to those in [8] and [18]. All SMS must be computed for generating a cover of KP.

Future work should focus on applying this proposed new concept to more general Petri nets such as ES^3PR [12] and improve the method by overcoming the above mentioned drawbacks. The relationship between KP and other important concepts, e.g., MPC [48], should be explored as well.

ACKNOWLEDGMENT

The authors would like to thank Prof. Z. Li and H. Hu who offer help on the computation.

REFERENCES

[1] I. B. Abdallah and H. ElMaraghy, "Deadlock prevention and avoidance in FMS: A Petri net based approach," *Int. J. Adv. Manuf. Tech.*, vol. 14, pp. 704–715, 1998.
[2] Z. A. Banaszak and B. H. Krogh, "Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows," *IEEE Trans. Robot. Automat.*, vol. 6, pp. 724–734, 1990.
[3] K. Barkaoui and I. B. Abdallah, "An efficient deadlock avoidance control policy in FMS using structural analysis of Petri nets," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, San Antonio, TX, Oct. 2–5, 1994, pp. 525–530.
[4] K. Barkaoui, A. Chaoui, and B. Zouari, "Supervisory control of discrete event systems based on structure theory of Petri nets," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Orlando, FL, 1997, pp. 3750–3755.
[5] D. Y. Chao, "Incremental approach to computation of elementary siphons for arbitrary simple sequential processes with resources," *IET Control Theory Appl.*, vol. 2, pp. 168–179, 2008.
[6] H. Cho, T. K. Kumaran, and R. A. Wysk, "Graph-theoretic deadlock detection and resolution for flexible manufacturing systems," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 413–421, 1995.
[7] F. Chu and X. L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. Robot. Automat.*, vol. 13, pp. 793–840, 1997.
[8] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 173–184, 1995.
[9] M. P. Fanti and M. C. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern., A.*, vol. 34, no. 1, pp. 5–22, Jan. 2004.
[10] B. Hrúz and M. C. Zhou, *Modeling and Control and Discrete-Event Dynamic Systems: With Petri Nets and Other Tool*. New York: Springer, 2007.
[11] Y. S. Huang, "Design of deadlock prevention supervisors using Petri nets," *Int. J. Adv. Manuf. Technol.*, vol. 35, pp. 349–362, 2007.

[12] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, "A deadlock prevention policy for flexible manufacturing systems using siphons," in *Proc. IEEE Int. Conf. Robot. Automat.*, Seoul, Korea, May 21–26, 2001, pp. 541–546.

[13] M. V. Iordache and P. J. Antsaklis*, Supervisory Control of Concurrent Systems: A Petri Net Structural Approach*. New York: Springer, 2006.

[14] M. D. Jeng, X. L. 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.*, vol. 34, no. 1, pp. 102–112, Jan. 2004.

[15] M. D. Jeng, X. L. Xie, and M. Y. Peng, "Process nets with resources for manufacturing modeling and their analysis," *IEEE Trans. Robot. Automat.*, vol. 18, pp. 875–889, 2002.

[16] T. K. Kumaran, W. Chang, H. Cho, and R. A. Wysk, "A structured approach to deadlock detection, avoidance, and resolution in flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 32, pp. 2361–2379, 1994.

[17] M. Lawley and S. Reveliotis, "Deadlock avoidance for sequential resource allocation systems: Hard and easy cases," *Int. J. Flex. Manuf. Syst.*, vol. 13, pp. 385–404, 2001.

[18] Z. W. Li and M. C. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern., A.*, vol. 34, no. 1, pp. 38–51, Jan. 2004.

[19] 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.*, vol. 38, no. 1, pp. 133–148, Jan. 2008.

[20] Z. W. Li and M. C. Zhou*, Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. New York: Springer, 2009.

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

[22] Z. W. Li, M. C. Zhou, and N. Q. Wu, "A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern., C.*, vol. 38, no. 1, pp. 173–188, Jan. 2008.

[23] G. J. Liu and C. J. Jiang, "Incidence matrix based methods for computing repetitive vectors and siphons of Petri net," *J. Inf. Sci. Eng.*, vol. 25, pp. 121–136, 2009.

[24] G. J. Liu, C. J. Jiang, L. J. Chen, and Z. H. Wu, "Two types of extended RSNBs and their application in modeling flexible manufacturing systems," *Int. J. Adv. Manuf. Tech.*, vol. 45, pp. 573–582, 2009.

[25] G. J. Liu, C. J. Jiang, Z. H. Wu, and L. J. Chen, "A live subclass of petri net and their application in modeling flexible manufacturing systems," *Int. J. Adv. Manuf. Tech.*, vol. 41, pp. 66–74, 2009.

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

[27] L. Piroddi, R. Cordone, and I. Fumagalli, "Selective siphon control for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern., A.*, vol. 38, pp. 1337–1348, 2008.

[28] L. Piroddi, M. Cossalter, and L. Ferrarini, "A resource decoupling approach for deadlock prevention in FMS," *Int. J. Adv. Manuf. Tech.*, vol. 40, pp. 157–170, 2009.

[29] S. A. Reveliotis, "Implicit siphon control and its role in the liveness-enforcing supervision of sequential resource allocation systems," *IEEE Trans. Syst., Man, Cybern., A.*, vol. 37, no. 3, pp. 319–328, May 2007.

[30] P. H. Starke, INA: Integrated Net Analyzer. [Online]. Available: http://www2.informatik.hu-berlin.de/~starke/ina.html 2003

[31] F. Tricas, F. García-Vallès, J. M. Colom, and J. Ezpeleta, "A structural approach to the problem of deadlock prevention in processes with resources," in *Proc. WODES'98*, Cagliari, Italy, Aug. 26–28, 1998, pp. 273–278.

[32] M. Uzam, "The use of petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 23, pp. 204–219, 2004.

[33] 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, pp. 192–208, 2002.

[34] 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, pp. 150–168, 2007.

[35] 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.*, vol. 37, no. 3, pp. 362–371, May 2007.

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

[37] N. Viswanadham, Y. Narahari, and T. L. Johnson, "Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models," *IEEE Trans. Robot. Automat.*, vol. 6, pp. 713–723, 1990.

[38] N. Q. 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.*, vol. 29, pp. 182–204, 1999.

[39] N. Q. Wu and M. C. Zhou, "Modeling and deadlock control of automated guided vehicle systems," *IEEE/ASME Trans. Mechatronics*, vol. 9, pp. 50–57, 2004.

[40] N. Q. Wu and M. C. Zhou, "Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles," *IEEE Trans. Syst., Man, Cybern., B.*, vol. 35, pp. 1193–1202, 2005.

[41] N. Q. Wu and M. C. Zhou, "Deadlock resolution in automated manufacturing systems with robots," *IEEE Trans. Automat. Sci. Eng.*, vol. 4, no. 3, pp. 474–480, Jul. 2007.

[42] N. Q. Wu and M. C. Zhou, "Deadlock modeling and control of semiconductor track systems using resource-oriented Petri nets," *Int. J. Prod. Res.*, vol. 45, no. 15, pp. 3439–3456, 2007.

[43] N. Q. Wu and M. C. Zhou*, System Modeling and Control With Resource-Oriented Petri Nets*. New York: CRC Press, 2010.

[44] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Resource-oriented Petri net for deadlock avoidance in flexible assembly systems," *IEEE Trans. Syst., Man, Cybern., A.*, vol. 38, no. 1, pp. 56–69, Jan. 2008.

[45] R. A. Wysk, N. S. Yang, and S. Joshi, "Detection of deadlocks in flexible manufacturing systems," *IEEE Trans. Robot. Automat.*, vol. 7, pp. 853–859, 1991.

[46] X. L. Xie and M. D. Jeng, "ERCN-merged nets and their analysis using siphons," *IEEE Trans. Robot. Automat.*, vol. 15, pp. 692–703, 1999.

[47] K. Y. Xing, B. S. Hu, and H. X. Chen, "Deadlock avoidance policy for Petri net modeling of flexible manufacturing systems with shared resources," *IEEE Trans. Automat. Contr.*, vol. 41, no. 2, pp. 289–295, Feb. 1996.

[48] K. Y. 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.*, vol. 39, no. 1, pp. 188–199, Jan. 2009.

[49] M. C. Zhou, F. DiCesare, and A. A. Desrochers, "A hybrid methodology for synthesis of Petri nets for manufacturing systems," *IEEE Trans. Robot. Automat.*, vol. 8, pp. 350–361, 1992.

GuanJun Liu received the B.S. degree from Qufu Normal Univerdity, Qufu, China, in 2001, and the M.S. degree from Shandong University of Science and Technology, Qingdao, China, in 2006. He is currently working towards the Ph.D. degree at the Department of Computer Science and Technology, Tongji University, Shanghai, China.

His research interests include discrete-event systems, formal language and automata theory, Petri nets, and service-oriented computing.

ChangJun Jiang received the Ph.D. degree from the Institute of Automation, Chinese Academy of Science, Beijing, China, in 1995.

He is currently a Professor in the Department of Computer Science and Engineering, Tongji University, Shanghai, China. He is also a Research Fellow at City University of Hong Kong, and a Specialist of Information Area of Shanghai. His current research interests include concurrency theory, Petri nets, formal verification of software, cluster, grid technology, program testing, intelligent transportation systems, and service-oriented computing. He has been engaged in concurrency theory and concurrency processing. He has published more than 100 papers in domestic and international academic publications including *Chinese Science*, the IEEE Transactions on Robotics and Automation, the IEEE Transactions on Fuzzy Systems, *International Journal of Computer Mathematics*, the *International Journal of Computer Systems Science and Engineering*, the *International Journal of Studies in Informatics and Control*, and the *International Journal of Advanced System Science and Application*. Furthermore, he has published four books (Science Publishing Foundation of the Chinese Academy of Science). He has taken in over 30 projects supported by the National Nature Science Foundation, National High Technology R&D Program, National Basic Research Developing Program, and other important and key projects at provincial or ministerial levels.

Prof. Jiang received one international prize and seven prizes in the field of science and technology. He is the recipient of the Advancement Awards of the Ministry of Education and Province, National Excellent Doctoral Dissertation, and Excellent Youth Teacher. He was selected in the Encouragement Program for Teaching and Researching of Excellent Young Teachers by the Ministry of Education and National Preeminent Youth Foundation.

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

He joined the New Jersey Institute of Technology (NJIT), Newark, NJ, in 1990, and is currently a Professor of Electrical and Computer Engineering and the Director of Discrete-Event Systems Laboratory. He also serves as the Director of the M.S. in Computer Engineering Program and Area Coordinator of Intelligent Systems at NJIT. He has over 350 publications including 10 books, 150+ journal papers (majority in IEEE transactions), and 17 book chapters. He has coauthored *Modeling and Control of Discrete Event Dynamic Systems*, Springer, London, 2007 (with B. Hruz), *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*, Springer, New York, 2009 (with Z. Li), and *System Modeling and Control with Resource-Oriented Petri Nets*, CRC Press, New York, 2010 (with N. Wu). His research interests are in intelligent automation, lifecycle engineering and sustainability evaluation, Petri nets, wireless ad hoc and sensor networks, system security, semiconductor manufacturing, and energy systems.

Dr. Zhou was the recipient of NSF's Research Initiation Award, CIM University-LEAD Award by the Society of Manufacturing Engineers, the Perlis Research Award by NJIT, the Humboldt Research Award for U.S. Senior Scientists, Leadership Award and Academic Achievement Award by the Chinese Association for Science and Technology-USA, the Asian American Achievement Award by the Asian American Heritage Council of New Jersey, and the Distinguished Lecturership of the IEEE SMC Society. He was invited to lecture in Australia, Canada, China, France, Germany, Hong Kong, Italy, Japan, Korea, Mexico, Singapore, Taiwan, and the U.S. He served as a plenary speaker for several conferences. He served as Associate Editor of the IEEE Transactions on Robotics and Automation from 1997 to 2000, and the IEEE Transactions on Automation Science and Engineering from 2004 to 2007, and currently Editor of the IEEE Transactions on Automation Science and Engineering, and Associate Editor of the IEEE Transactions on Systems, Man and Cybernetics: Part A and the IEEE Transactions on Industrial Informatics. He served as Guest-Editor for many journals including the IEEE Transactions on Industrial Electronics and the IEEE Transactions on Semiconductor Manufacturing. He was General Chair of the IEEE Conference on Automation Science and Engineering, Washington DC, August 23–26, 2008, General Co-Chair of the 2003 IEEE International Conference on System, Man and Cybernetics (SMC), Washington, DC, October 5–8, 2003, Founding General Co-Chair of the 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, March 21–23, 2004, and General Chair of the 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, April 23–25, 2006. He was Program Chair of the 1998 and 2001 IEEE International Conference on SMC and the 1997 IEEE International Conference on Emerging Technologies and Factory Automation. He organized and chaired over 80 technical sessions and served on program committees for many conferences. He has led or participated in 40 research and education projects with a total budget over $10M, funded by National Science Foundation, Department of Defense, Engineering Foundation, New Jersey Science and Technology Commission, and industry. He is Program Committee Chair of the 2010 IEEE International Conference on Mechatronics and Automation, August 4–7, 2010, Xi'an, China. He was the founding Chair of the Discrete-Event Systems Technical Committee and founding Co-Chair of the Enterprise Information Systems Technical Committee of the IEEE SMC Society, and Chair (founding) of the Semiconductor Manufacturing Automation Technical Committee of the IEEE Robotics and Automation Society. He is a Life Member of the Chinese Association for Science and Technology-USA and served as its President in 1999.