Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 579623, 15 pages
http://dx.doi.org/10.1155/2015/579623

Research Article

Deadlock Prevention Policy with Behavioral Optimality or Suboptimality Achieved by the Redundancy Identification of Constraints and the Rearrangement of Monitors

Liang Hong,[1] YiFan Hou,[2] JunFeng Jing,[1] AnRong Wang,[2] and Dmitry A. Litvin[3]

[1] *College of Electronics and Information, Xi'an Polytechnic University, No. 19 South Jinhua Road, Xi'an 710048, China*
[2] *School of Electro-Mechanical Engineering, Xidian University, No. 2 South Taibai Road, Xi'an 710071, China*
[3] *Faculty of Telecommunication Networks, Odessa National Academy of Telecommunications Named after A.S. Popov, Koval'ska Street 1, Odessa 65029, Ukraine*

Correspondence should be addressed to Liang Hong; chelseagoon@gmail.com

Received 22 May 2015; Accepted 20 September 2015

Academic Editor: Kamel Barkaoui

This work develops an iterative deadlock prevention method for a special class of Petri nets that can well model a variety of flexible manufacturing systems. A deadlock detection technique, called mixed integer programming (MIP), is used to find a strict minimal siphon (SMS) in a plant model without a complete enumeration of siphons. The policy consists of two phases. At the first phase, SMSs are obtained by MIP technique iteratively and monitors are added to the complementary sets of the SMSs. For the possible existence of new siphons generated after the first phase, we add monitors with their output arcs first pointed to source transitions at the second phase to avoid new siphons generating and then rearrange the output arcs step by step on condition that liveness is preserved. In addition, an algorithm is proposed to remove the redundant constraints of the MIP problem in this paper. The policy improves the behavioral permissiveness of the resulting net and greatly enhances the structural simplicity of the supervisor. Theoretical analysis and experimental results verify the effectiveness of the proposed method.

1. Introduction

Deadlocks [1] always appear in the operations of a flexible manufacturing system (FMS). Hence, deciding how to reduce the impact of deadlocks is a very tricky problem that we have to cope with. Digraphs, automata, and Petri nets are three major mathematical tools to deal with deadlock problems in resource allocation systems.

Digraphs are used to deal with the deadlock detection and avoidance in [2, 3]. Researchers [4, 5] adopt automata as model tools to dispose the deadlock problem while Petri nets [6] are applied to model and analyze FMS. The main Petri net strategies are deadlock detection and recovery [7, 8], deadlock avoidance [9–11], and deadlock prevention [12–19].

This paper focuses our attention on deadlock prevention problems. A Petri net based deadlock prevention approach utilizes an off-line computational mechanism to impose constraints on a system to prevent the system from reaching deadlock states. Monitors (control places) and their related arcs are used to achieve such purposes and collectively called a supervisor of the plant model.

The theory of regions, as a technique to design supervisors for a Petri net, is adopted in [13, 20, 21]. Generally, the theory can lead to an optimal supervisor [22–28] if it exists. However, the theory is based on a reachability graph, which may cause state explosion with a net size increasing.

McMillan and Probst propose the concept of unfolding nets in [29] to describe the behavior of an FMS. A prefix of an unfolding net is adequate to completely describe the properties of a net, which is a concise but efficient method compared with the theory of regions. However, deciding how to find a complete and simple prefix is still worth consideration.

Siphons are special sets of places of a Petri net, by controlling which one can effectively prevent deadlocks. Ezpeleta et al. [12] propose a policy by enumerating siphons and impose constraints for the siphons to solve deadlock prevention problems. However, it is a time-consuming task with nets scale expansion [30]. In addition, behavioral permissiveness and structural complexity are tough issues that we have to face. Li and Zhou [14] propose elementary siphon theory, where enumerated siphons are divided into two parts: elementary and dependent. Monitors are only needed for the elementary siphons as long as the dependent siphons are controllable, which greatly reduces the structural complexity. Moreover, the computational complexity is reduced and the behavioral permissiveness is enhanced in the subsequent research [15, 31–38].

Huang et al. [39] propose a two-stage deadlock prevention policy for System of Simple Sequential Processes with Resources (S^3PR), a class of Petri nets that was proposed in [12]. The policy explores SMS based on the MIP technique, developed by Chu and Xie [40] for structurally bounded nets. For S^3PR, at the first stage, by the MIP technique, maximal unmarked siphon is obtained first if there exist deadlocks. SMS can be derived from the maximal unmarked siphon and a corresponding constraint, imposed on the complementary set of the SMSs to prevent it from being unmarked, is obtained enforcing the constraint to the MIP problem to check the liveness of the plant net under the constraint. If there still exists a maximal unmarked siphon, repeat the above process till the plant net is live under the derived constraints. Thus, we can obtain one or more constraints and add corresponding monitors (including their related arcs) to the complementary sets of the SMSs. Hence, the resulting net obtained after the first stage is a net consisting of the plant net and the corresponding monitors. Control-induced siphons (composed by operation places, resource places, and the monitors) can possibly be generated due to the addition of the monitors. The second stage, similar to the first one, is still an iterative process in finding siphons and the difference is that the monitors of this stage are added with their output arcs pointed to source transitions, which makes the control-induced siphons controlled and no problematic siphons were generated. The policy can usually lead to a more permissive supervisor than that proposed in [12].

However, there exist the following defects in [39]. First, the SMS obtained after an iteration is nondeterministic due to the following two reasons. The first is that the solution (corresponds to a maximal unmarked siphon) of the MIP problem is not unique and the second is that we can find different SMSs from the same maximal unmarked siphon via different place selection sequences. Thus, the sequence of the generation of SMSs is uncertain. Sometimes, SMS can be controlled if the subsequent SMSs are controlled. Hence, adding a monitor for the SMS is redundant and makes the control structure complex.

Second, at the second stage in [39], the constraint, mentioned at Step 17 of Algorithm 2 of [39] and used for exploring the condition of liveness in the considered MIP problem, is imposed on the complementary set of new generated SMSs (derived from a control-induced siphon) while the output arcs of the constraint corresponding monitor are pointed to source transitions. It causes the fact that the constraints obtained at the second stage are always more than the necessary monitors. However, the number of added monitors for SMSs is requested consistent with that of the obtained constraints in terms of the policy, which leads to the fact that redundant monitors are generated and the behavioral permissiveness is generally restricted.

This work improves [39] in terms of structural complexity and behavioral permissiveness by the following three points. First, for removing the redundant constraints, an algorithm is developed that checks the redundancy of a constraint by deciding whether the liveness is preserved after its removal. This operation markedly reduces the structural complexity. Second, a new type of constraints is constructed to replace the one that emerged at Step 17 of Algorithm 2 in [39] to reduce structural complexity and enhance behavioral permissiveness. The generation of redundant monitors is avoided by imposing the new constraint on the complementary set of newly generated SMSs and the set of its upstream places (defined in Definition 14), which makes the constraint have the same effect with the monitors added at the second stage on preventing the newly generated SMS from being unmarked. Third, for each output arc of the monitors added at the second stage, it is led from the source transition step by step to release more legal states on condition that the liveness is preserved. In summary, the improvements largely enhance the performance of a supervisor for a plant net.

The rest of this paper is organized as follows. Preliminaries used in this paper are presented in the next section. In Section 3, we introduce a deadlock prevention policy that mainly contains two phases: siphon control phase and extended siphon control phase. The specific method is shaped to an algorithm in Section 4. By experimental analysis, the performance of the proposed method is shown in Section 5. Finally, Section 6 concludes this paper.

2. Preliminaries

2.1. Basics of Petri Nets. A Petri net is a four-tuple $N = (P, T, F, W)$, where P and T are the sets of places and transitions, respectively. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation, represented by arcs with arrows from transitions to places or from places to transitions. $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ if $(x, y) \in F$, where $x \in P$, $y \in T$ (or $x \in T$, $y \in P$), and $\mathbb{N} = \{0, 1, 2, \ldots\}$. If $W(x, y) \leq 1$, $\forall (x, y) \in F$, the net is called an ordinary Petri net; otherwise, it is called a generalized Petri net.

A marking (also called a state) M is a mapping from P to $\mathbb{N}$. The number of tokens in place p is denoted by $M(p)$. A place p is marked at a marking M if $M(p) > 0$. $M(S)$ denotes the sum of tokens of all places in S; that is, $M(S) = \sum_{p \in S} M(p)$, where $S \subseteq P$. S is marked at M if $M(S) > 0$. S is unmarked at M if $M(S) = 0$. (N, M_0) is called a net system and M_0 is called an initial marking of N.

Let $x \in P \cup T$ be a node of $N = (P, T, F, W)$. ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ is called the preset of x and $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$ is called the postset of x. Similar notation extended to a set of nodes is as follows: given $X \subseteq P \cup T$, ${}^\bullet X = \cup_{x \in X} {}^\bullet x$, $X^\bullet = \cup_{x \in X} x^\bullet$, ${}^{\bullet\bullet} X = \cup_{x \in X} {}^{\bullet\bullet} x$, and $X^{\bullet\bullet} = \cup_{x \in X} x^{\bullet\bullet}$.

A nonempty set $S \subseteq P$ is called a siphon if $^\bullet S \subseteq S^\bullet$. S is minimal if there is no siphon in S as a proper subset. S is strict if $^\bullet S \subsetneq S^\bullet$.

A transition $t \in T$ is enabled at a marking M if, $\forall p \in {}^\bullet t$, $M(p) \geq W(p,t)$ and denoted as $M[t\rangle$. Firing t yields a new marking M' that can be denoted by $M[t\rangle M'$ and M' is called an immediately reachable marking from M. A marking M'' is said to be reachable from M if there exists a sequence of transitions $\sigma = t_0 t_1 \cdots t_n$ and markings $M_1, M_2, \ldots$, and M_n such that $M[t_0\rangle M_1[t_1\rangle M_2 \cdots M_n[t_n\rangle M''$ holds. The set of markings reachable from M in N is called the reachability set of Petri net (N, M) and denoted as $R(N, M)$. The set of legal states of net (N, M_0) is defined as follows: $\mathscr{M} = \{M \mid M \in R(N, M_0) \wedge M_0 \in R(N, M)\}$.

A transition $t \in T$ is live at M_0 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 at M_0. It is dead at M_0 if $\nexists t \in T$, $M_0[t\rangle$. It is deadlock-free if $\forall M \in R(N, M_0)$, $\exists t \in T$, $M[t\rangle$.

A P-vector is a column vector $I : P \rightarrow \mathbb{Z}$ indexed by P and a T-vector is a column vector $J : T \rightarrow \mathbb{Z}$ indexed by T, where $\mathbb{Z}$ is the set of integers. The column vectors where every entry equals 0(1) are denoted by $\mathbf{0}(\mathbf{1})$. $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p,t) = W(t,p) - W(p,t)$. I^T is the transposed versions of vector I. P-vector I is called a P-invariant if $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$ and $\|I\| = \{p \mid I(p) \neq 0\}$ is the support of I. I is minimal if its support is not contained in the support of any other and its components are mutually prime. P-invariant I is a P-semiflow if its every element is nonnegative.

2.2. S^{3}PR.

In this subsection, we introduce a class of Petri nets, called S^{3}PR, first defined in [12], which stands for *Systems of Simple Sequential Processes with Resources* and can model real-life automated FMSs.

Definition 1 (see [12]). An ordinary Petri net $N = (P, T, F)$ is called a state machine if, $\forall t \in T$, $|{}^\bullet t| = |t^\bullet| = 1$. It is strongly connected if, $\forall x, y \in P \cup T$, there is a sequence of nodes $x, a, b, \ldots, c$, and y such that $(x, a), (a, b), \ldots$, and $(c, y) \in F$, where $\{a, b, \ldots, c\} \subseteq P \cup T$.

Definition 2 (see [12]). A simple sequential process (S^{2}P) is a Petri net $N = (\{p^0\} \cup P_A, T, F)$, where $P_A \neq \emptyset$ is the set of operation places, p^0 is the idle process place, N is a strongly connected state machine, and the circuit of N contains p^0.

Definition 3 (see [12]). A simple sequential process with resources (S^{2}PR) is a Petri net $N = (\{p^0\} \cup P_A \cup P_R, T, F)$ such that

(1) the subnet generated by $X = P_A \cup \{p^0\} \cup T$ is S^{2}P,

(2) $P_R \neq \emptyset$ and $(P_A \cup \{p^0\}) \cap P_R = \emptyset$, where P_R is called the set of resource places,

(3) $\forall p \in P_A$, $\forall t \in {}^\bullet p$, $\forall t' \in p^\bullet$, $\exists r_p \in P_R$, ${}^\bullet t \cap P_R = t'^\bullet \cap P_R = \{r_p\}$,

(4) (a) $\forall r \in P_R$, ${}^{\bullet\bullet} r \cap P_A = r^{\bullet\bullet} \cap P_A \neq \emptyset$ and (b), $\forall r \in P_R$, ${}^\bullet r \cap r^\bullet = \emptyset$,

(5) ${}^{\bullet\bullet}(p^0) \cap P_R = (p^0)^{\bullet\bullet} \cap P_R = \emptyset$.

Definition 4 (see [12]). S^{3}PR is a system of S^{2}PR, defined as follows:

(1) An S^{2}PR is an S^{3}PR.

(2) Let $N_i = (\{p_i^0\} \cup P_{A_i} \cup P_{R_i}, T_i, F_i)$, $i \in \{1, 2\}$, be two S^{3}PR such that $(P_{A_1} \cup \{p_1^0\}) \cap (P_{A_2} \cup \{p_2^0\}) = \emptyset$, $P_{R_1} \cap P_{R_2} = P_C \neq \emptyset$, and $T_1 \cap T_2 = \emptyset$. Then, combine N_1 and N_2 via P_C into a net $N = (P^0 \cup P_A \cup P_R, T, F)$. N is also an S^{3}PR, defined as follows: (1) $P_A = P_{A_1} \cup P_{A_2}$, (2) $P^0 = \{p_1^0\} \cup \{p_2^0\}$, (3) $P_R = P_{R_1} \cup P_{R_2}$, (4) $T = T_1 \cup T_2$, and (5) $F = F_1 \cup F_2$.

Transitions in $(P^0)^\bullet$ are called *source transitions*, which represent the entry of raw materials when FMS is modeled with an S^{3}PR.

Definition 5 (see [12]). For $r \in P_R$, $H(r) = {}^{\bullet\bullet} r \cap P_A$ is called the set of holders of r, which represent the operation places that use r. Let S be s siphon and $S^R = S \cap P_R$; $[S] = (\cup_{r \in S^R} H(r)) \setminus S$ is called the *complementary set* of S.

Theorem 6 (see [12]). *An S^{3}PR (N, M_0) is live if, $\forall M \in R(N, M_0)$, $\forall S \in \Pi$, $M(S) > 0$.*

Let Π be the set of SMSs in an S^{3}PR. The above theorem indicates that an S^{3}PR is live if there is no siphon that can be emptied.

2.3. MIP Technique.

Let (N, M) be an ordinary net with $N = (P, T, F)$ and let S be the maximal unmarked siphon at M; that is, $\forall p \notin S$, $M(p) > 0$. In the sequel, we introduce a technique, first proposed in [40], to find S in N by exploring the solution of a mixed integer programming (MIP) problem. $\forall p \notin S$, let $v_p = 1$ and, $\forall t \notin S^\bullet$, let $z_t = 1$.

Since S is a siphon and, $\forall p \notin S$, $M(p) > 0$, any p with $v_p = 1$ and any t with $z_t = 1$ should be removed from the solution. Furthermore, $\forall t \in p^\bullet$, $v_p = 0$ implies $z_t = 0$ and, $\forall p \in t^\bullet$, $z_t = 1$ implies the truth of $v_p = 1$. This leads to

$$\begin{aligned} & z_t \geq \sum_{p \in {}^\bullet t} v_p - |{}^\bullet t| + 1, \quad \forall t \in T, \\ & v_p \geq z_t, \quad \forall (t, p) \in F, \\ & \qquad\qquad v_p, z_t \in \{0, 1\}. \end{aligned} \tag{1}$$

For a structurally bounded net, we have

$$v_p \geq \frac{M(p)}{\mathrm{SB}(p)}, \quad \forall p \in P, \tag{2}$$

where $\mathrm{SB}(p) = \max\{M(p) \mid M = M_0 + [N]Y,\ M \geq 0,\ Y \geq 0\}$ denotes the structural bound of place p. Therefore, the maximal unmarked siphon can be determined by the following MIP problem and there exist siphons unmarked in (N, M_0) if $G^{\mathrm{MIP}}(M_0) < |P|$:

$$G^{\mathrm{MIP}}(M_0) = \text{Minimize} \sum_{p \in P} v_p \tag{3}$$

subject to constraints (1)-(2) and

$$M = M_0 + [N]\,Y, \quad M \geq 0,\ Y \geq 0. \tag{4}$$

Although MIP problems are NP-hard, it is shown in [40] that its computational efficiency is relatively insensitive to the initial marking and more efficient than state enumeration methods.

Theorem 7 (see [40]). *Let (N, M_0) be an ordinary Petri net with $N = (P, T, F)$. (N, M_0) is live if $G^{MIP}(M_0) = |P|$.*

Theorem 7 is used to check $\mathrm{S}^3\mathrm{PR}$ (N, M_0) whether liveness is enforced or not. If $G^{\mathrm{MIP}}(M_0)$ equals the cardinality of the place set of N, it implies that (N, M_0) is live.

However, the classical MIP problem to determine a maximal unmarked siphon in [40] only applies to ordinary Petri nets. As for a generalized Petri net, deadlocks may occur due to insufficiently marked siphons. Hence, the new conditions for checking whether there exist a maximal unmarked siphon in a generalized Petri net (N, M_0) with $N = (P, T, F, W)$ are presented in [41], where a new constraint (5) is used to replace constraint (2):

$$v_p \geq \frac{(M(p) - W(p,t) + 1)}{\mathrm{SB}(p)}, \qquad \forall p \in P,\ \forall W(p,t) > 0. \tag{5}$$

The maximal unmarked siphon of a generalized Petri net (N, M_0) can be determined by the following MIP problem and there exist unmarked siphons if $G^{\mathrm{MIP}}_{\mathrm{gen}}(M_0) < |P|$:

$$G^{\mathrm{MIP}}_{\mathrm{gen}}(M_0) = \mathrm{Minimize} \sum_{p \in P} v_p \tag{6}$$

subject to constraints (1) and (5) and

$$M = M_0 + [N]\,Y, \quad M \geq 0,\ Y \geq 0. \tag{7}$$

Theorem 8 (see [41]). *Let (N, M_0) be a generalized Petri net with $N = (P, T, F, W)$. (N, M_0) is live if $G^{MIP}_{gen}(M_0) = |P|$.*

For a generalized Petri net (N, M_0), if $G^{\mathrm{MIP}}_{\mathrm{gen}}(M_0)$ equals the cardinality of the place set of N, we can conclude that (N, M_0) is live.

3. Siphon Control Approach

In general, the presented method mainly contains two phases: siphon control phase and extended siphon control phase. The two phases are similar in finding siphons and the difference between them is the fashion of adding monitors for the siphons.

First, we apply the MIP technique to a plant net to obtain a maximal unmarked siphon (if there exists one), derive a minimal siphon from the maximal one by the minimal siphon extraction algorithm in [42], and check liveness by solving an MIP problem of the plant net after enforcing a constraint imposed on the complementary set of the minimal siphons. Repeat the above process until the plant net is live under a set of constraints. After removing redundant constraints by a proposed algorithm, we add monitors to the complementary sets of the minimal siphons according to the remaining constraints. If in the resulting net there still exists a deadlock, we conduct the next phase.

At the second phase, a newly presented constraint, different from the one claimed at Step 17 of Algorithm 2 in [39], is enforced to the MIP problem of the resulting net of the first phase for preventing the system from reaching deadlock states. After iterations, we obtain a set of constraints and implement the constraints by adding monitors with the output arcs (with weights) pointing to the source transitions to avoid the generation of new problematic siphons. Then the output arcs of the monitors are rearranged to obtain a more permissive supervisor.

3.1. Siphon Control Phase

3.1. Siphon Control Phase. At this phase, for modeling a maximally permissive supervisor as much as possible, a monitor for an SMS is designed to be imposed on the complementary set of the SMSs.

Definition 9 (see [39]). Let S be an SMS in an $\mathrm{S}^3\mathrm{PR}$ (N, M_0). A monitor V_S for S is added to N to prevent S from being unmarked such that

(1) ${}^\bullet V_S = \{t \in p^\bullet \mid p \in [S],\ p^{\bullet\bullet} \notin [S]\}$; $V_S^\bullet = \{t \in {}^\bullet p \mid p \in [S],\ {}^{\bullet\bullet}p \notin [S]\}$,

(2) $\forall t \in {}^\bullet V_S,\ W(t, V_S) = 1$; $\forall t \in V_S^\bullet, W(V_S, t) = 1$,

(3) $M(V_S) = M_0(S) - 1$,

where $W(V_S, t)$ and $W(t, V_S)$ denote the weights of related arcs of V_S and $M(V_S)$ denotes the initial number of tokens in V_S.

Theorem 10 (see [43]). *The addition of V_S for S minimally restricts the behavior of a plant net.*

Definition 9 shows the fashion of adding a monitor for an SMS at the first phase. In the sequel, we should identify SMSs that need to be controlled.

In this study, the MIP technique is applied to iteratively find SMSs to avoid a complete siphon enumeration. For an $\mathrm{S}^3\mathrm{PR}$ (N, M_0), an SMS S can be found in N by the MIP technique if it is not live. A constraint $\sum_{p \in [S]} M(p) \leq M_0(S) - 1$ is enforced to the MIP problem of N, which implies that S cannot be unmarked any more in the MIP problem while the constraint minimally restricts the behavior of the plant net, where $M(p)$ is a variable of the MIP problem and denotes the number of tokens in p and $M_0(S)$ is the number of tokens in S at the initial marking. If N is live under the constraint, we just need to enforce the constraint by adding a corresponding monitor to N and end the first phase. Otherwise, we need to iterate and will obtain a set of constraints in the MIP problem, under which N can never reach deadlock states.

However, if the number of constraints is more than one, there may be the case that not all the constraints are necessary. As mentioned in Introduction, some constraints can be replaced by the combination of other constraints and

```
Input: A Petri net $(N, M)$ and a set of constraints $C = \{c_1, \ldots, c_n\}$.
Output: A set of necessary constraints $C_N$.
(1) begin
(2) $C_N := \emptyset$. \* $C_N$ denotes a set of necessary constraints *\
(3) while $\{C \neq \emptyset\}$
    (3.1) Choose a constraint $c_i$ from $C$; $C := C \setminus \{c_i\}$.
    (3.2) Enforce $C \cup C_N$ to the MIP problem of $N$.
    (3.3) if there exists a maximal unmarked siphon
        (3.3.1) $C_N := C_N \cup \{c_i\}$.
(4) Output $C_N$.
(5) end
```

Algorithm 1: Extraction of necessary constraints.

therefore they are redundant. Here, we propose an algorithm to exclude redundant constraints and the remaining ones are necessary.

Algorithm 1 eliminates redundancy by excluding a constraint each time. If the liveness is preserved after removing the constraint, we can conclude that the constraint is redundant and should be removed. Otherwise, it should be considered as necessary.

A necessary constraint corresponds to an SMS that needs to be controlled. The constraint is achieved by adding a monitor and related arcs to the plant net to prevent the corresponding SMS from being unmarked. Hence, a necessary constraint requires a monitor.

Given an S^{3}PR (N, M_0) with $N = (P^0 \cup P_A \cup P_R, T, F, W)$, we enforce the necessary constraints by adding monitors according to Definition 9 and denote the resulting net as (N^1, M^1), where $N^1 = (P^0 \cup P_A \cup P_R \cup P_V, T, F \cup F_1, W_1)$ and P_V denotes the set of monitors V_S. Then,

(1) $\forall p \in P^0 \cup P_A \cup P_R$, $\forall t \in T$, $W_1(p,t) = W(p,t)$, $W_1(t,p) = W(t,p)$ and

(2) $\forall t \in {}^{\bullet}V_S$, $\forall V_S \in P_V$, $W_1(t, V_S) = 1$; $\forall t \in V_S^{\bullet}$, $\forall V_S \in P_V$, $W_1(V_S, t) = 1$,

(3) $\forall p \in P_A \cup P^0 \cup P_R$, $M^1(p) = M_0(p)$,

(4) $\forall V_S \in P_V$, $M^1(V_S)$ is the same as $M(V_S)$ in Definition 9.

Theorem 11 (see [39]). *(N^1, M^1) is live if no siphon in N^1 can be emptied.*

Now, the S^{3}PR net depicted in Figure 1(a) is taken as an example. It has 11 places, where $\{p_1, p_5\}$ is the set of idle places, $\{p_2, p_3, p_4, p_6, p_7, p_8\}$ is the set of operation places, and $\{p_9, p_{10}, p_{11}\}$ is the set of resource places. Applying the MIP technique to the net, a maximal unmarked siphon $\{p_4, p_7, p_8, p_9, p_{10}, p_{11}\}$ is found. An SMS can be derived from the maximal one, denoted as $S_1 = \{p_4, p_7, p_{10}, p_{11}\}$, where p_{10} and p_{11} are resource places and $\{p_3, p_7\}$ and $\{p_4, p_6\}$ denote their holder sets, respectively. According to the definition of complementary sets, we have $[S_1] = \{p_3, p_6\}$ and $M_0(S_1) - 1 = 3 - 1 = 2$. Constraint $M(p_3) + M(p_6) \leq 2$ is enforced to the MIP problem of the plant net to check whether the plant net is live under the constraint. If it is not live, we need to continue to find an SMS and enforce a corresponding constraint to the MIP problem of the plant net. After three iterations, the plant net is live under three derived constraints. We find that there exists no redundant constraint by applying Algorithm 1. Three monitors are added by Definition 9 and shown in Figure 1(b).

We find that all of the above three constraints are necessary. However, if we change the initial marking of the net in Figure 1(a) to the one shown in Figure 2(a), we may find three constraints in turn:

$$M(p_2) + M(p_3) + M(p_6) + M(p_7) \leq 4, \tag{8}$$

$$M(p_2) + M(p_7) \leq 2, \tag{9}$$

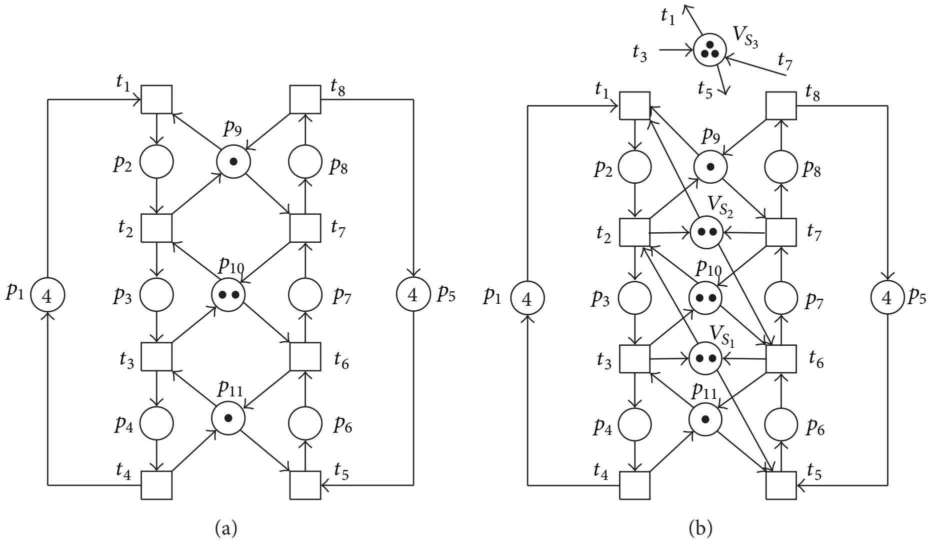
$$M(p_3) + M(p_6) \leq 2. \tag{10}$$

If constraint (8) can be found first, we still require constraints (9) and (10) to guarantee liveness, which makes constraint (8) redundant. Consequently, Algorithm 1 becomes an essential tool to deal with this condition. The controlled net is shown in Figure 2(b).

In fact, the condition of Theorem 11 is not always met. Newly added monitors and resource places may coproduce new siphons. Therefore, the second phase is proposed to solve this problem.

3.2. Extended Siphon Control Phase. In this subsection, we still utilize the MIP-based deadlock detection method to find siphons (if there exist ones). In order to avoid the case that the added monitors take part in generating new siphons (i.e., control-induced siphons), the output arcs of monitors (with weighted arcs) added at this phase are pointed to source transitions first. Then the output arcs are rearranged to be far away from the source transitions to release legal states. The following definitions are presented to introduce the fashion of adding a monitor for an SMS at the second phase.

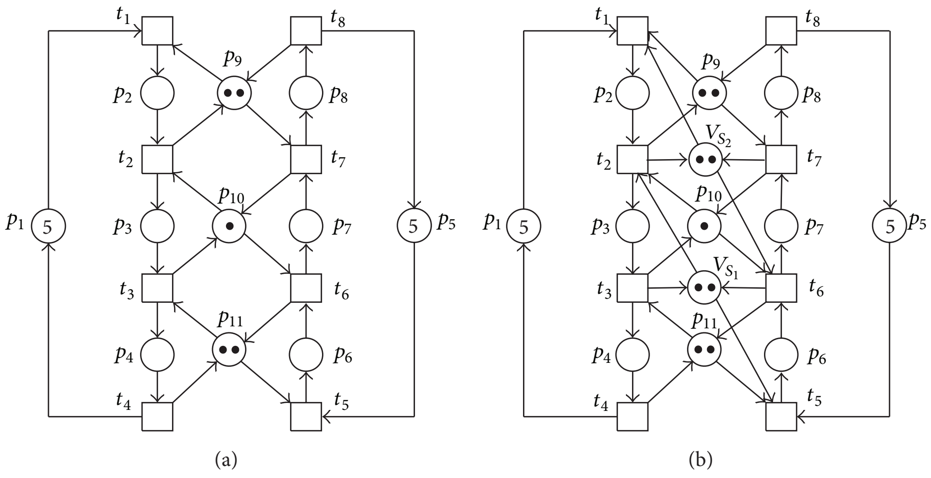
Definition 12. Let (N, M_0) be an S^{3}PR, let p^0 be the idle place of a process, and let p_a and p_b be different operation places



(a)

(b)

Figure 1: (a) An S^{3}PR (N, M_0) and (b) a controlled system of N.



(a)

(b)

Figure 2: (a) An S^{3}PR (N, M_0) and (b) a controlled system of N.

in the process. If p_b can be found in the process path (in accordance with the direction of the arrows of the process flow) from p_a to p^0 (p_a and p^0 are not included), p_b is called a downstream place of p_a, denoted as $p_b \prec p_a$, and p_a is called an upstream place of p_b, denoted as $p_a \succ p_b$.

Definition 13. Let p be an operation place of an S^{3}PR. $P_D(p) = \{q \mid q \prec p\}$ is called the set of downstream places of p and $P_U(p) = \{q \mid q \succ p\}$ is called the set of upstream places of p.

Definition 14. Let S be an SMS of an S^{3}PR, let $[S]$ be the complementary set of S, and let p be an operation place in $[S]$. $P_U([S]) = \bigcup_{p\in[S]}\{q \mid q \in P_U(p) \wedge q \notin [S]\}$ is called the set of upstream places of $[S]$ and $P_{U_i}([S]) = P_U([S]) \cap P_{A_i}$ is called the set of upstream places of $[S]$ in process i, where P_{A_i} denotes the set of operation places in process i.

Take the S^{3}PR net (N, M_0) depicted in Figure 1(a) as an example. p_4 is a downstream place of p_3, denoted as $p_4 \prec p_3$. p_2 is an upstream place of p_3, denoted as $p_2 \succ p_3$. Moreover, p_3 and p_4 are the downstream places of p_2, denoted as $P_D(p_2) = \{p_3, p_4\}$. p_2 and p_3 are the upstream places of p_4, denoted as $P_U(p_4) = \{p_2, p_3\}$. It is supposed that $P_{A_1} = \{p_2, p_3, p_4\}$ and $P_{A_2} = \{p_6, p_7, p_8\}$. $S_1 = \{p_4, p_7, p_{10}, p_{11}\}$ and $S_2 = \{p_3, p_8, p_9, p_{10}\}$ are two SMSs in N. $P_U([S_1]) = \{p_2\}$, $P_{U_1}([S_1]) = \{p_2\}$, and $P_{U_2}([S_1]) = \emptyset$. Similarly, $P_U([S_2]) = \{p_6\}$, $P_{U_1}([S_1]) = \emptyset$, and $P_{U_2}([S_1]) = \{p_6\}$.

Definition 15. Let S be an SMS of (N^1, M^1) with $N^1 = (P^0 \cup P_A \cup P_R \cup P_V, T, F \cup F_1)$. The complementary set of S is defined as $[S] = \bigcup_{V_S\in S} H(V_S) \bigcup_{r\in S} H(r) \backslash S$, where $V_S \in P_V$ and $r \in P_R$. $[S]$ can be written in the multiset form $Th_S = \sum_{p\in H(V_S)} p + \sum_{p\in H(r)} p - \sum_{p\in S} p$ and $Th_S(p)$ represents the coefficient of p.

Suppose that $Th_S = a \cdot p_l + b \cdot p_m + c \cdot p_n$, and we have $Th_S(p_l) = a$, $Th_S(p_m) = b$, and $Th_S(p_n) = c$.

Definition 16. Let S be an SMS of (N^1, M^1). $Q_S(p) = \max\{\max_{q \in P_D(p)} Th_S(q), Th_S(p)\}$ is called the maximum demand for resources of p on S.

Next, we illustrate the definition with the example $Th_S = a \cdot p_l + b \cdot p_m + c \cdot p_n + \cdots + r \cdot p_x + s \cdot p_y + t \cdot p_z$. Suppose that $\{p_l, p_m, p_n\}$ is a set of operation places in a process with $p_l \succ p_m \succ p_n$ while $\{p_x, p_y, p_z\}$ is a set of operation places in another process, and we have $P_D(p_l) = \{p_m, p_n\}$, $Q_S(p_l) = \max\{Th_S(p_l), Th_S(p_m), Th_S(p_n)\}$, $P_D(p_m) = \{p_n\}$, and $Q_S(p_m) = \max\{Th_S(p_m), Th_S(p_n)\}$.

Definition 17 (see [39]). Let S be an SMS in (N^1, M^1). A monitor EV_S for S is added to N^1 to prevent S from being unmarked such that

(1) $\forall t \in P^{0\bullet}, W(EV_S, t) = Q_S(p)\ (p \in t^\bullet \cap P_A)$,

(2) $\forall t \in T \setminus P^{0\bullet}, W(t, EV_S) = Q_S(p) - Q_S(p')\ (p \in {}^\bullet t \cap P_A, p' \in t^\bullet \cap P_A)$,

(3) $M(EV_S) = M^1(S) - 1$,

where $W(EV_S, t)$ and $W(t, EV_S)$ denote the weights of related arcs of EV_S and $M(EV_S)$ denotes the initial number of tokens in EV_S.

If (N^1, M^1) is not live, we can find an SMS S containing monitors added at the first phase. In [39], at the second stage, a constraint $\sum_{p \in [S]} M(p) \leq M^1(S) - 1$ is enforced to the MIP problem of N^1 to prevent S from being unmarked in the MIP problem, where $M(p)$ is a variable of the MIP problem and denotes the number of tokens in p, $M^1(S)$ is the number of tokens in S at M^1. By exprimental analysis, it is found that the constraint is imposed on the complementary set of S while the output arcs of monitors are pointed to source transitions to prevent the generation of control-induced siphons, which makes redundant constraints emerged. Hence, in the following, a new type of constraints is proposed to solve the problem.

Based on Definitions 12–16, we propose a new constraint, $\sum_{p \in [S] \cup P_U([S])} Q_S(p) \cdot M(p) \leq M^1(S) - 1$, to replace $\sum_{p \in [S]} M(p) \leq M^1(S) - 1$, where $M(p)$ is a variable of the MIP problem and denotes the number of tokens in p and $M^1(S)$ is the number of tokens in S at M^1. Similar to the first phase, for controlling the derived SMSs in the MIP problem, we can obtain one or a set of such constraints, under which N^1 is live. Applying Algorithm 1 to the constraints, we find a set of necessary constraints and each of them corresponds to an SMS that needs to be controlled.

For the net (N^1, M^1) with $N^1 = (P^0 \cup P_A \cup P_R \cup P_V, T, F \cup F_1, W_1)$ obtained after the first phase, the necessary constraints are enforced by adding monitors to N^1 according to Definition 17 and the resulting net is denoted as (N^2, M^2), where $N^2 = (P^0 \cup P_A \cup P_R \cup P_V \cup P_{EV}, T, F \cup F_1 \cup F_2, W_2)$ and P_{EV} denotes the set of monitors EV_S. Then,

(1) $\forall p \in P^0 \cup P_A \cup P_R \cup P_V, \forall t \in T, W_2(p, t) = W_1(p, t), W_2(t, p) = W_1(t, p)$,

(2) $\forall t \in T$, $\forall EV_S \in P_{EV}$, $W_2(EV_S, t)$ and $W_2(t, EV_S)$ are the same as $W(EV_S, t)$ and $W(t, EV_S)$ in Definition 17, respectively,

(3) $\forall p \in P^0 \cup P_A \cup P_R \cup P_V, M^2(p) = M^1(p)$,

(4) $\forall EV_S \in P_{EV}$, $M^2(EV_S)$ is the same as $M(EV_S)$ in Definition 17.

Theorem 18 (see [44]). *(N^2, M^2) is live.*

The fashion of adding monitors in Definition 17 restricts the behavioral permissiveness while avoiding the generation of control-induced siphons. Hence, we utilize Algorithm 2 to release legal states.

Algorithm 2 aims to construct a more permissive supervisor, which releases most legal states. For each monitor added by Definition 17, move each of its output arcs that originally points to a source transition step by step away from the source transition. Note that all the movements are implemented on condition that the liveness is preserved and the liveness is checked by the MIP problem for generalized Petri nets.

Theorem 19. *$(N^{2'}, M^{2'})$ is live.*

Proof. $(N^{2'}, M^{2'})$ is obtained based on the rearrangements of the output arcs of monitors in (N^2, M^2). By Theorem 18, it is found that (N^2, M^2) is live. In addition, each movement of the output arcs is implemented on condition that $G_{\text{gen}}^{\text{MIP}}(M^{m2})$ ((N^{m2}, M^{m2}) denotes the resulting net of each movement) equals the cardinality of the place set of N^{m2}, which implies that there is no unmarked siphon. According to Theorem 8, $(N^{2'}, M^{2'})$ is live. □

For the S^3PR shown in Figure 3, the number of its maximally permissive states is 891. The first phase leads to 12 monitors, as shown in Table 1. The resulting net is denoted as (N^1, M^1). However, there still exist maximal unmarked siphons rendering deadlocks. Two minimal siphons $S_{13} = \{p_5, p_{10}, p_{16}, p_{21}, p_{22}, p_{24}, p_{25}, V_{S_4}, V_{S_6}, V_{S_8}\}$ and $S_{14} = \{p_5, p_{10}, p_{16}, p_{21}, p_{22}, p_{24}, p_{25}, V_{S_3}, V_{S_4}, V_{S_9}\}$ can be derived. The multisets of $[S_{13}]$ and $[S_{14}]$ are presented as below. N^1 is live under constraints (12) and (13).

Multisets are as follows:

$$\begin{aligned} Th_{S_{13}} &= 2p_2 + p_3 + p_4 + 2p_7 + 2p_8 + 2p_9 + 3p_{12} \\ &\quad + 2p_{13} + 3p_{14} + p_{15} + 2p_{18} + 2p_{19} + p_{20}, \\ Th_{S_{14}} &= 3p_2 + p_3 + p_4 + 2p_7 + 2p_8 + 3p_9 + 2p_{12} + p_{13} \\ &\quad + 2p_{14} + p_{15} + p_{18} + p_{19} + p_{20}. \end{aligned} \tag{11}$$

```
Input: Net $(N^2, M^2)$ with $N^2 = (P \cup P_V \cup P_{EV}, T, F \cup F_1 \cup F_2, W_2)$.
Output: A more permissive supervisor $(N^{2'}, M^{2'})$.
(1) begin
(2) $P_{EV}^{\text{fin}} := \emptyset$. \* $P_{EV}^{\text{fin}}$ denotes the set of monitors with rearranged output arcs *\
(3) while $\{P_{EV} \neq \emptyset\}$
    (3.1) Choose a monitor $EV_S$ from $P_{EV}$; $P_{EV} := P_{EV} \setminus \{EV_S\}$; $P^{m1} := P \cup P_V \cup P_{EV} \cup P_{EV}^{\text{fin}}$.
          The resulting net is denoted by $(N^{m1}, M^{m1})$, where $N^{m1} = (P^{m1}, T, F \cup F_1 \cup F_{m1}, W_{m1})$.
    (3.2) Find $P_U([S]) = \bigcup_{i \in K} P_{U_i}([S])$, where $K = \{1, \ldots, k\}$ and $k$ is the total number of processes.
          \* $S$ is the SMS controlled by $EV_S$ *\
    (3.3) foreach $P_{U_i}([S])$ $(0 < i < k + 1)$,
        (3.3.1) $P_{IR} := \{p_i^0\}^{\bullet\bullet}$; $P'_{IR} := \emptyset$; $P_U([S]) := P_U([S]) \setminus P_{U_i}([S])$.
                  \* $P_{IR}$ and $P'_{IR}$ denote two sets of operation places *\
        (3.3.2) while $\{P_{IR} \neq \emptyset\}$
            (3.3.2.1) Choose $p \in P_{IR}$; $P_{IR} := P_{IR} \setminus \{p\}$.
            (3.3.2.2) $P_{U_i}([S]) := P_{U_i}([S]) \setminus \{p\}$; $P_U([S]) := P_U([S]) \cup P_{U_i}([S])$.
            (3.3.2.3) $EV_S$ with rearranged output arcs is added to $(N^{m1}, M^{m1})$
                      such that $EV_S + \sum_{p \in P_U([S])} p + \sum_{p \in [S]} p$ is a $P$-invariant of the resulting net,
                      denoted by $(N^{m2}, M^{m2})$, where $N^{m2} = (P^{m2}, T, F \cup F_1 \cup F_{m2}, W_{m2})$ and $P^{m2} = P^{m1} \cup \{EV_S\}$.
            (3.3.2.4) if $\{G_{\text{gen}}^{\text{MIP}}(M^{m2}) = |P^{m2}|\}$
                (3.3.2.4.1) $P'_{IR} := P'_{IR} \cup \{p\}$.
            (3.3.2.5) else
                (3.3.2.5.1) $P_U([S]) := P_U([S]) \setminus P_{U_i}([S])$; $P_{U_i}([S]) := P_{U_i}([S]) \cup \{p\}$; $P_U([S]) := P_U([S]) \cup P_{U_i}([S])$.
            (3.3.2.6) Remove $EV_S$ from $(N^{m2}, M^{m2})$ and obtain $(N^{m1}, M^{m1})$.
        (3.3.3) if $\{P'_{IR} \neq \emptyset\}$
            (3.3.3.1) $P_{IR} := P'^{\bullet\bullet}_{IR}$; $P'_{IR} := \emptyset$; go to (3.3.2)
    (3.4) $EV_S$ with rearranged output arcs is added to $(N^{m1}, M^{m1})$ such that $EV_S + \sum_{p \in P_U([S])} p + \sum_{p \in [S]} p$ is a $P$-invariant
        of the resulting net, denoted by $(N^{2'}, M^{2'})$; $P_{EV}^{\text{fin}} := P_{EV}^{\text{fin}} \cup \{EV_S\}$.
(4) Output $(N^{2'}, M^{2'})$ with $N^{2'} = (P \cup P_V \cup P_{EV}^{\text{fin}}, T, F \cup F_1 \cup F_{2'}, W_{2'})$.
(5) end
```

ALGORITHM 2: Synthesis of a more permissive supervisor.

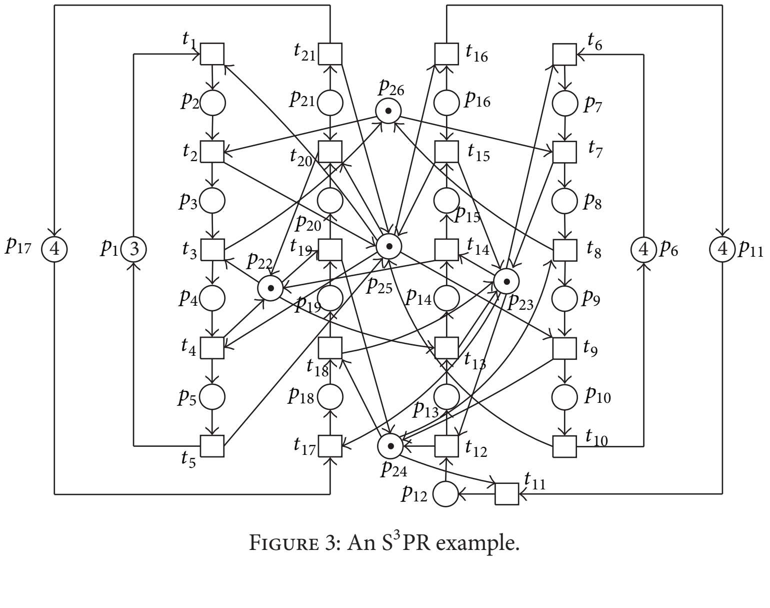
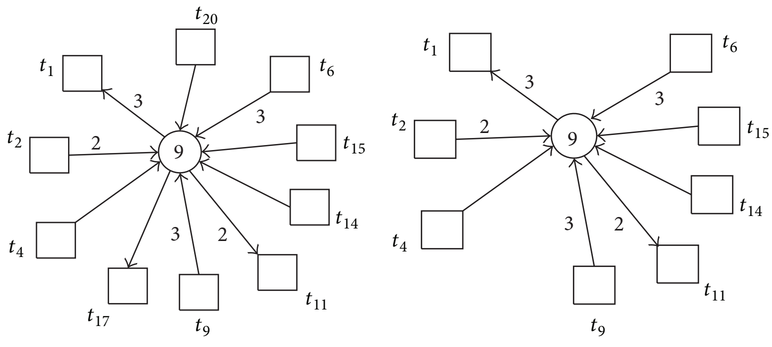


FIGURE 3: An S^3PR example.

Table 1: Monitors added at the first phase.

V_{S_i}	$M^1(V_{S_i})$	Preset	Postset	V_{S_i}	$M^1(V_{S_i})$	Preset	Postset
V_{S_1}	4	t_4, t_9, t_{15}, t_{20}	$t_1, t_6, t_{11}t_{17}$	V_{S_2}	3	t_4, t_7, t_{15}, t_{20}	t_1, t_6, t_{11}, t_{17}
V_{S_3}	3	t_3, t_8, t_{14}, t_{19}	t_1, t_6, t_{11}, t_{17}	V_{S_4}	3	t_4, t_9, t_{20}	t_1, t_6, t_{17}
V_{S_5}	2	t_4, t_{20}	t_1, t_{17}	V_{S_6}	3	t_2, t_9, t_{15}, t_{18}	t_1, t_6, t_{11}, t_{17}
V_{S_7}	1	t_{12}, t_{18}	t_{11}, t_{17}	V_{S_8}	2	t_3, t_7, t_{14}	t_1, t_6, t_{11}
V_{S_9}	2	t_{14}, t_{19}	t_{11}, t_{17}	$V_{S_{10}}$	2	t_2, t_9	t_1, t_6
$V_{S_{11}}$	2	t_8, t_{12}, t_{18}	t_6, t_{11}, t_{17}	$V_{S_{12}}$	1	t_{14}	t_{11}



(a)

(b)

Figure 4: Monitor $EV_{S_{13}}$ (a) with original output arcs and (b) with rearranged output arcs.

Constraints are as follows:

$$\begin{aligned} & 2M(p_2) + M(p_3) + M(p_4) + 2M(p_7) + 2M(p_8) \\ & \quad + 2M(p_9) + 3M(p_{12}) + 3M(p_{13}) + 3M(p_{14}) \\ & \quad + M(p_{15}) + 2M(p_{18}) + 2M(p_{19}) + M(p_{20}) \leq 9, \end{aligned} \tag{12}$$

$$\begin{aligned} & 3M(p_2) + M(p_3) + M(p_4) + 3M(p_7) + 3M(p_8) \\ & \quad + 3M(p_9) + 2M(p_{12}) + 2M(p_{13}) + 2M(p_{14}) \\ & \quad + M(p_{15}) + M(p_{18}) + M(p_{19}) + M(p_{20}) \leq 9. \end{aligned} \tag{13}$$

It is found that constraint (10) is redundant by applying Algorithm 1. According to constraint (12), $EV_{S_{13}}$ with its output arcs pointed to source transitions, as shown in Figure 4(a), is added to N^1 by Definition 17. The resulting net N^2 is live and it has 870 reachable states. Finally, we find $EV_{S_{13}}$ with rearranged output arcs by Algorithm 2, as depicted in Figure 4(b). We denote the resulting net as $N^{2'}$. It is live and has 878 reachable states that are very close to the number of maximally permissive states.

4. Deadlock Prevention Algorithm

In this section, the proposed method is shaped to an algorithm to show how to synthesize a liveness-enforcing supervisor and a supporting example is given in Algorithm 3.

The supervisor synthesized by Algorithm 3 preserves the legal states of a plant net to a large extent on a basis of low computational complexity. The MIP-based deadlock detection approach is iteratively used to find unmarked siphons, which avoids a complete siphon enumeration and saves computational time. The first phase is optimal or maximally permissive in the sense that no legal states are excluded since a monitor is added to the complementary set of SMSs. However, it always makes the generation of control-induced siphons unavoidable while minimally restricting the behavior of a plant net. To accelerate the convergence, the output arcs of the monitors added at the second phase are first pointed to the source transitions of the plant net and subsequently rearranged away from the source transitions to release legal states on condition that the liveness is preserved. It guarantees the liveness as well as largely enhances the behavioral permissiveness. Hence, the policy in Algorithm 3 can synthesize a liveness-enforcing supervisor with low computational complexity.

The Petri net shown in Figure 5 is a well-known model of FMS [12]. We can define this Petri net model as an S^3PR

```
Input: An $S^3PR$ $(N, M_0)$ with $N = (P, T, F)$.
Output: A liveness-enforcing net $(N^{2'}, M^{2'})$ with $N^{2'} = (P \cup P_V \cup P_{EV}, T, F \cup F_1 \cup F_{2'}, W_{2'})$.
(1) begin
(2) $C_{SC} := \emptyset$; $P_V := \emptyset$.    \* $C_{SC}$ and $P_V$ denote the sets of constraints in the MIP problem and monitors obtained
                       at the first phase, respectively *\
(3) Apply MIP to $N$ to obtain a maximal unmarked siphon.
(4) if there exists such a siphon then
        (4.1) Obtain a minimal siphon $S_i$ from the maximal one.
        (4.2) Enforce a constraint $c_i$, $\sum_{p\in[S_i]} M(p) \le M_0(S_i) - 1$, to the MIP problem of $N$; $C_{SC} := C_{SC} \cup \{c_i\}$; go to step (3).
(5) if $\{C_{SC} = \emptyset\}$ then $N^{2'} := N$; $M^{2'} := M_0$; go to step (17).
(6) Apply Algorithm 1 to $C_{SC}$, obtain a set of necessary constraints $C_N$.
(7) foreach $c_i \in C_N$,
        (7.1) Add a corresponding monitor $V_{S_i}$ to $N$ by Definition 9.
        (7.2) $P_V := P_V \cup \{V_{S_i}\}$.
(8) Obtain $(N^1, M^1)$ and $N^1 = (P \cup P_V, T, F \cup F_1)$.
(9) $C_{ESC} := \emptyset$; $P_{EV} := \emptyset$.    \* $C_{ESC}$ and $P_{EV}$ denote the sets of constraints in the MIP problem and monitors obtained
                   at the second phase, respectively *\
(10) Apply MIP to $N^1$ to obtain a maximal unmarked siphon.
(11) if there exists such a siphon then
        (11.1) Obtain a minimal siphon $S_{E_i}$ from the maximal one.
        (11.2) Enforce a constraint $c_{Ei}$, $\sum_{p\in[S_{E_i}]\cup P_U([S_{E_i}])} Q_S(p) \cdot M(p) \le M^1(S_i) - 1$, to the MIP problem of $N^1$;
            $C_{ESC} := C_{ESC} \cup \{c_{Ei}\}$; go to step (10).
(12) if $\{C_{ESC} = \emptyset\}$ then $N^{2'} := N^1$; $M^{2'} := M^1$; go to step (17).
(13) Apply Algorithm 1 to $C_{ESC}$ to obtain a set of necessary constraints $C_{EN}$.
(14) foreach $c_{Ei} \in C_{EN}$,
        (14.1) Add a corresponding monitor $EV_{S_i}$ to $N^1$ by Definition 17.
        (14.2) $P_{EV} := P_{EV} \cup \{EV_{S_i}\}$; $N^2 = (P \cup P_V \cup P_{EV}, T, F \cup F_1 \cup F_2, W_2)$.
(15) Obtain $(N^2, M^2)$ and $N^2 = (P \cup P_V \cup P_{EV}, T, F \cup F_1 \cup F_2, W_2)$.
(16) Apply Algorithm 2 to $(N^2, M^2)$ to find $(N^{2'}, M^{2'})$.
(17) Output $(N^{2'}, M^{2'})$.
(18) end
```

Algorithm 3: A liveness-enforcing policy.

(N, M_0) if $P^0 = \{p_1, p_5, p_{14}\}$, $P_A = P_1 \cup P_2 \cup P_3 = \{p_2, p_3, p_4\} \cup \{p_6, p_7, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{13}\} \cup \{p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$, and $P_R = \{p_{20}, p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}\}$.

The MIP technique is applied to N and a maximal unmarked siphon $S_{\max 1} = \{p_1, p_4, p_8, p_9, p_{10}, p_{11}, p_{12}, p_{15}, p_{17}, p_{18}, p_{20}, p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}\}$ can be obtained. A minimal siphon $S_1 = \{p_{10}, p_{18}, p_{22}, p_{26}\}$ can be derived from $S_{\max 1}$. Thus, we have $[S_1] = \{p_{13}, p_{19}\}$ and $M_0(S_1) - 1 = M_0(p_{22}) + M_0(p_{26}) - 1 = 2$. Constraint $M(p_{13}) + M(p_{19}) \le 2$ is enforced to the MIP problem of N. Unfortunately, there still exists a maximal unmarked siphon. The above process proceeds until we find seven constraints under which N is live. There is no redundant constraint acquired by Algorithm 1 implying that there are seven SMSs to be controlled. Then we add seven monitors $P_V = \{V_{S_i} \mid i \in \{1, \ldots, 7\}\}$ to N by Definition 9, as shown in Table 2, and denote the resulting net as (N^1, M^1).

Table 2: Monitors added at the first phase.

V_{S_i}	$M^1(V_{S_i})$	Preset	Postset
V_{S_1}	2	t_{10}, t_{16}	t_9, t_{15}
V_{S_2}	2	t_4, t_{13}	t_3, t_{11}
V_{S_3}	5	$t_5, t_{10}, t_{13}, t_{17}$	t_3, t_8, t_{11}, t_{15}
V_{S_4}	2	t_8, t_{18}	t_7, t_{17}
V_{S_5}	5	t_3, t_8, t_{19}	t_1, t_{17}
V_{S_6}	2	t_9, t_{17}	t_8, t_{16}
V_{S_7}	3	t_{10}, t_{17}	t_8, t_{15}

Applying the MIP technique to N^1, we can still find a maximal unmarked siphon $S_{\max 8} = \{p_2, p_3, p_4, p_8, p_{10}, p_{12}, p_{13}, p_{15}, p_{16}, p_{17}, p_{20}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}, V_{S_3}, V_{S_4}, V_{S_5}, V_{S_6}, V_{S_7}\}$. A minimal siphon $S_8 = \{p_{13}, p_{17}, p_{26}, V_{S_4}, V_{S_6}\}$ can be extracted from $S_{\max 8}$, and the multisets form of its complementary set are $Th_{S_8} = p_{11} + p_{12} + 2p_{18}$. We have $M^1(S_8) - 1 = M^1(p_{26}) + M^1(V_{S_4}) + M^1(V_{S_6}) - 1 = 2 + 2 + 2 - 1 = 5$. Then constraint $M(p_6) + M(p_{11}) + M(p_{12}) + 2M(p_{18}) + 2M(p_{19}) \le 5$ is enforced to the MIP problem of N^1. However, there still exists a maximal unmarked siphon and N^1 requires another constraint to guarantee liveness. It is found that both

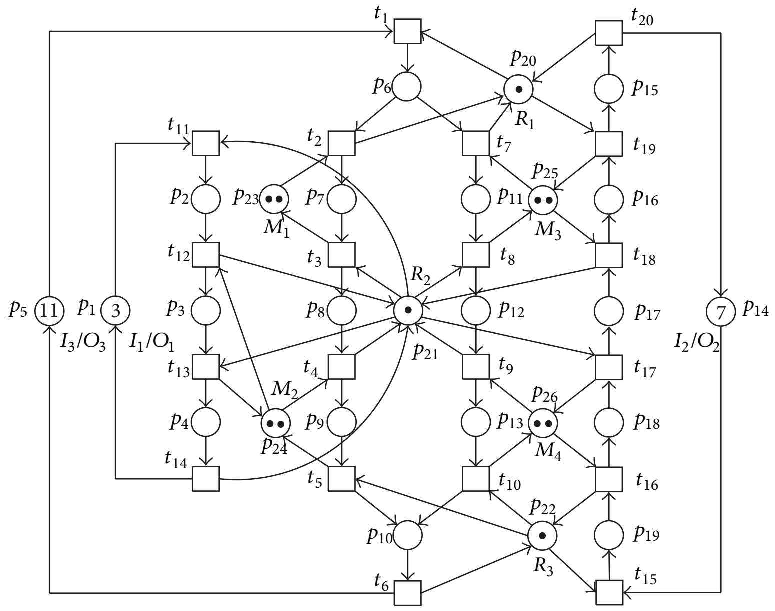
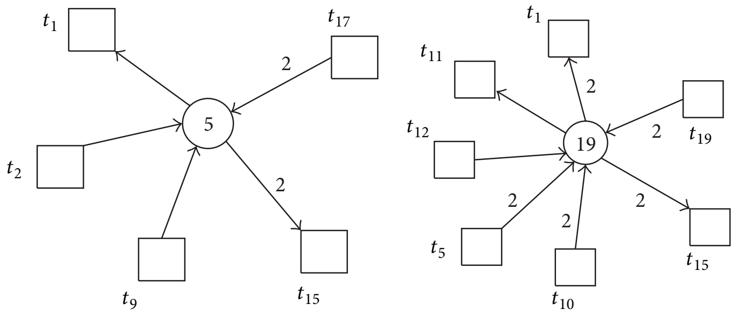


Figure 5: The Petri net model of an FMS.



(a)

(b)

Figure 6: (a) Monitors EV_{S_8} and (b) EV_{S_9}.

constraints are necessary by Algorithm 1 and we add two monitors, $P_{EV} = \{EV_{S_i} \mid i \in \{8, 9\}\}$, as shown in Figure 6, to N^1 by Definition 17. The resulting net is denoted as (N^2, M^2).

(N^2, M^2) excludes many legal states although it is live. The number of maximally permissive states of this plant model is 21581 while (N^2, M^2) preserves 19773 ones. Then we apply Algorithm 2 to (N^2, M^2) to rearrange the output arcs of monitors added at the second phase. The two monitors with rearranged output arcs are shown in Figure 7. The final net, denoted by $(N^{2\prime}, M^{2\prime})$, is live and it has 20444 reachable states.

5. Comparison and Discussion

The performance of the proposed policy is shown in Table 3 by calculating the example in Figure 5 compared with the existing methods. The first column represents the policies and the second column shows the number of reachable states. The third and fourth list the numbers of the added monitors and the related arcs, respectively. The fifth exhibits whether a complete siphon enumeration is necessary in a policy. Similarly, the sixth points out the necessity of a complete state enumeration. The seventh indicates the computational complexity.

The performance analysis of the policies is conducted by considering the following three criteria: behavioral permissiveness, computational complexity, and structural complexity. The advantage of the proposed policy can be analyzed in terms of the three criteria. The obtained supervisor by the proposed method of the net in Figure 5 is suboptimal. However, the number of its reachable states is nearly maximally permissive, only a bit smaller than those of the policies in

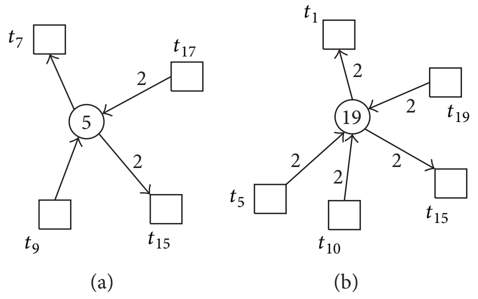


Figure 7: (a) Monitors EV_{S_8} with rearranged output arcs and (b) EV_{S_9} with rearranged output arcs.

Table 3: Performance comparison of policies applied to the example in Figure 5.

Policy referred to	Number of states	Number of monitors	Number of arcs	Complete siphon	Complete state	Computational complexity
Piroddi et al. [45]	21581	13	82	√	✗	Exponential
Ezpeleta et al. [12]	6287	18	106	√	✗	Exponential
Huang et al. [39]	12656	16	88	✗	✗	NP-hard
Huang [47]	16425	7	34	√	✗	Exponential
Li and Zhou [14]	6287	6	32	√	✗	Exponential
Li et al. [32]	6331	5	27	✗	✗	NP-hard
Li and Zhou [15]	15999	6	29	✗	✗	NP-hard
Park and Reveliotis [16]	2480	7	38	√	✗	Polynomial
Tricas et al. [17]	14850	8	40	✗	✗	NP-hard
Uzam and Zhou [46]	21562	19	112	✗	√	Exponential
Xing and Hu [48]	15098	4	17	√	✗	Exponential
The proposed policy	20444	9	45	✗	✗	NP-hard

[45, 46] while the computational complexity of these policies is exponential, their supervisors have more monitors and arcs, and they need either a complete siphon enumeration or a complete reachability graph. The proposed policy, compared with the policies in [15, 17, 32, 39], can provide more reachable states.

Among the policies listed in Table 3, we choose four representative ones to compare with the proposed policy in Table 4 through ten examples (obtained by an S^3PR-generating program of our work group except examples 7, 8, and 10), where number of RS and number of M denote the numbers of reachable states and the additonal monitors, respectively. The 12th and the last column indicate the maximally permissive behavior and the ratio of the reachable states of supervisors obtained by the proposed policy to the maximally permissive behavior, respectively. By the experimental study, we can find that the supervisor, synthesized by the proposed policy, has less monitors and is more permissive than the existing ones.

6. Conclusions

The deadlock prevention problem is always a tough and fussy task in FMSs. Behavioral permissiveness, computational complexity, and structural complexity are three criteria for checking the performance of supervisors. This work presents a deadlock prevention method for an S^3PR. A siphon is a special set of places that can lead a system to a deadlock state. The MIP technique is iteratively used to find maximal unmarked siphons. Due to the high computational efficiency of the MIP method, we can save computation time substantially. At the first phase, in order to find a more permissive supervisor, monitors are added to the complementary sets of the derived minimal siphons. Unfortunately, the added monitors are always involved in the generation of control-induced siphons. Hence, at the second phase, a new fashion of adding monitors such that the output arcs of the monitors are pointed to source transitions is adopted to avoid generating control-induced siphons. Then the output arcs are moved from the source transitions along the opposite direction to

Table 4: Performance analysis through ten examples.

Example	Ezpeleta et al. 1995 [12]		Huang et al. 2001 [39]		Li and Zhou 2006 [15]		Li et al. 2007 [32]		Proposed policy		Maximally permissive behavior	Optimality ratio
	Number of RS	Number of M	Number of RS	Number of M	Number of RS	Number of M	Number of RS	Number of M	Number of RS	Number of M		
1	48	5	196	5	97	1	97	1	196	4	205	95.61%
2	268	5	336	4	268	3	268	3	336	4	336	100%
3	250	5	516	5	250	2	250	2	516	5	516	100%
4	103	11	632	6	118	3	112	3	632	5	632	100%
5	642	12	827	17	830	8	696	8	878	13	891	98.54%
6	792	6	1746	3	1572	2	800	2	1746	3	1746	100%
7 [49]	1463	13	2108	12	1591	4	1591	4	2108	10	2108	100%
8 [50]	585	8	2328	9	1518	3	594	3	2328	7	2328	100%
9	8500	11	16116	11	9000	11	8500	11	16116	11	16116	100%
10 [12]	6287	18	12656	16	15999	6	6331	6	20444	9	21581	94.73%

the work flow for releasing legal states on condition that the liveness is preserved. In addition, this work proposes a redundant constraint removal algorithm that greatly reduces the structural complexity and enhances the behavioral permissiveness.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant nos. 61301276, 61403296, and 61202128, the Natural Science Basic Research Plan in Shaanxi Province of China under Grant no. 2015JQ6258, the Discipline Construction Funds of Xi'an Polytechnic University under Grant no. 107090811, and the Doctoral Research Startup Funds of Xi'an Polytechnic University.

References

[1] M. P. Fanti and M. C. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 34, no. 1, pp. 5–22, 2004.

[2] H. Cho, T. K. Kumaran, and R. A. Wysk, "Graph-theoretic deadlock detection and resolution for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 3, pp. 413–421, 1995.

[3] M. P. Fanti, B. Maione, S. Mascolo, and B. Turchiano, "Event-based feedback control for deadlock avoidance in flexible production systems," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 3, pp. 347–363, 1997.

[4] M. A. Lawley, S. A. Reveliotis, and P. M. Ferreira, "A correct and scalable dead-lock avoidance policy for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 5, pp. 796–809, 1998.

[5] S. A. Reveliotis and P. M. Ferreira, "Deadlock avoidance policies for automated manufacturing cells," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 6, pp. 845–857, 1996.

[6] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Short-term scheduling of crude-oil operations: enhancement of crude-oil operations scheduling using a Petri net-based control-theoretic approach," *IEEE Robotics & Automation Magazine*, vol. 22, no. 2, pp. 64–76, 2015.

[7] R. A. Wysk, N. S. Yang, and S. Joshi, "Detection of deadlocks in flexible manufacturing cells," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 853–859, 1991.

[8] Y. S. Huang, Y. L. Pan, and P. J. Su, "Deadlock detection and recovery policy for FMSs using graph technique," *ACM Transactions on Embedded Computing Systems*, vol. 12, no. 1, pp. 404–415, 2013.

[9] Z. A. Banaszak and B. H. Krogh, "Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 6, pp. 724–734, 1990.

[10] F.-S. Hsieh and S.-C. Chang, "Dispatching-driven deadlock avoidance controller synthesis for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 2, pp. 196–209, 1994.

[11] N. Q. Wu, M. Zhou, and Z. W. Li, "Resource-oriented Petri net for deadlock avoidance in flexible assembly systems," *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, vol. 38, no. 1, pp. 56–69, 2008.

[12] J. Ezpeleta, J. M. Colom, and J. Martinez, "Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 2, pp. 173–184, 1995.

[13] A. Ghaffari, N. Nidhal, and X. L. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 137–141, 2003.

[14] Z. W. Li and M. C. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 34, no. 1, pp. 38–51, 2004.

[15] Z. W. Li and M. C. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Transactions on Industrial Informatics*, vol. 2, no. 4, pp. 313–325, 2006.

[16] J. Park and S. A. Reveliotis, "Deadlock avoidance in sequential resource allocations systems with multiple resource acquisitions and flexible routings," *IEEE Transactions on Automatic Control*, vol. 46, no. 10, pp. 1572–1583, 2001.

[17] F. Tricas, F. G. Valls, J. M. Colom, and J. Ezpeleta, "An iterative method for deadlock prevention in FMSs," in *Proceedings of the 5th Workshop Discrete Event Systems*, pp. 139–148, Ghent, Belgium, August 2000.

[18] J. H. Ye, Z. W. Li, and A. Giua, "Decentralized supervision of Petri nets with a coordinator," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 6, pp. 955–966, 2015.

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

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

[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 Transactions on Automation Science and Engineering*, vol. 5, no. 1, pp. 182–188, 2008.

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

[23] Y. F. Chen, Z. W. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 2, pp. 374–393, 2011.

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

[25] Y. F. Chen, Z. W. Li, and M. C. Zhou, "Optimal supervisory control of flexible manufacturing systems by petri nets: a set classification approach," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 2, pp. 549–563, 2014.

[26] Y. F. Chen, Z. W. Li, K. Barkaoui, and M. Uzam, "New petri net structure and its application to optimal supervisory control: interval inhibitor arcs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 10, pp. 1384–1400, 2014.

[27] Y.-S. Huang, Y.-L. Pan, and M. C. Zhou, "Computationally improved optimal deadlock control policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 42, no. 2, pp. 404–415, 2012.

[28] Z. Y. Ma, Z. W. Li, and A. Giua, "Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 1774–1785, 2015.

[29] K. L. McMillan and D. K. Probst, "A technique of state space search based on unfolding," *Formal Methods in System Design*, vol. 6, no. 1, pp. 45–65, 1995.

[30] K. Lautenbach, "Linear algebraic calculation of deadlocks and traps," in *Concurrency and Nets*, pp. 315–336, Springer, New York, NY, USA, 1987.

[31] Z. W. Li and M. C. Zhou, "Clarifications on the definitions of elementary siphons in Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 36, no. 6, pp. 1227–1229, 2006.

[32] Z. W. Li, H. S. Hu, and A. R. Wang, "Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 37, no. 4, pp. 517–526, 2007.

[33] Z. W. Li and M. C. Zhou, "On siphon computation for deadlock control in a class of Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 38, no. 3, pp. 667–679, 2008.

[34] Z. W. Li and M. C. Zhou, "Control of elementary and dependent siphons in Petri nets and their application," *IEEE Transactions on Systems, Man, and Cybernetics A: Systems and Humans*, vol. 38, no. 1, pp. 133–148, 2008.

[35] 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 Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews*, vol. 38, no. 2, pp. 173–188, 2008.

[36] Z. W. Li and M. Zhao, "On controllability of dependent siphons for deadlock prevention in generalized Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 38, no. 2, pp. 369–384, 2008.

[37] Z. W. Li, G. Y. Liu, H.-M. Hanisch, and M. C. Zhou, "Deadlock prevention based on structure reuse of petri net supervisors for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 42, no. 1, pp. 178–191, 2012.

[38] Z. W. Li, N. Q. Wu, and M. C. Zhou, "Deadlock control of automated manufacturing systems based on Petri nets—a literature review," *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 42, no. 4, pp. 437–462, 2012.

[39] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, "Deadlock prevention policy based on Petri nets and siphons," *International Journal of Production Research*, vol. 39, no. 2, pp. 283–305, 2001.

[40] F. Chu and X.-L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Transactions on Robotics and Automation*, vol. 13, no. 6, pp. 793–804, 1997.

[41] M. Zhao, Z. W. Li, and H. S. Hu, "Suboptimal liveness-enforcing supervisor design for a class of generalised Petri nets using partial siphon enumeration and mathematical programming," *International Journal of Systems Science*, vol. 41, no. 9, pp. 1013–1026, 2010.

[42] Z. W. Li and D. Liu, "A correct minimal siphons extraction algorithm from a maximal unmarked siphon of a Petri net," *International Journal of Production Research*, vol. 45, no. 9, pp. 2161–2165, 2007.

[43] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion contraints on nets with uncontrollable transitions," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 974–979, IEEE, Chicago, Ill, USA, October 1992.

[44] F. Tricas, F. G. Valls, J. M. Colom, and J. Ezpeleta, "A structural approach to the problem of deadlock prevention in processes with shared resources," in *Proceedings of the 4th Workshop on Discrete Event Systems (WODES '98)*, pp. 273–278, Cagliari, Italy, 1998.

[45] L. Piroddi, R. Cordone, and I. Fumagalli, "Selective siphon control for deadlock prevention in Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 38, no. 6, pp. 1337–1348, 2008.

[46] M. Uzam and M. C. Zhou, "An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems," *International Journal of Production Research*, vol. 44, no. 10, pp. 1987–2030, 2006.

[47] Y. S. Huang, "Design of deadlock prevention supervisors using Petri nets," *The International Journal of Advanced Manufacturing Technology*, vol. 35, no. 3-4, pp. 349–362, 2007.

[48] K. Y. Xing and B. S. Hu, "Optimal liveness Petri net controllers with minimal structures for automated manufacturing systems," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 282–287, IEEE, Waikoloa, Hawaii, USA, October 2005.

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

[50] Z. W. Li and M. C. Zhou, *Modeling, Analysis, and Deadlock Control of Automated Manufacturing Systems*, Science Press, Beijing, China, 2009 (Chinese).

Advances in Operations Research
Advances in Decision Sciences
Journal of Applied Mathematics
Algebra
Journal of Probability and Statistics
The Scientific World Journal
International Journal of Differential Equations
International Journal of Combinatorics
Advances in Mathematical Physics
Journal of Complex Analysis
Journal of Mathematics
Mathematical Problems in Engineering
Abstract and Applied Analysis
Discrete Dynamics in Nature and Society
International Journal of Mathematics and Mathematical Sciences
Journal of Discrete Mathematics
Journal of Function Spaces
International Journal of Stochastic Analysis
Journal of Optimization