

Received August 1, 2020, accepted September 1, 2020, date of publication September 8, 2020, date of current version September 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022643

# On Multi-Step Look-Ahead Deadlock Prediction for Automated Manufacturing Systems Based on Petri Nets

RONGFENG LIN<sup>1</sup>, ZHENHUA YU<sup>2</sup>, (Member, IEEE), XIAONAN SHI<sup>2</sup>,  
LIHONG DONG<sup>2</sup>, AND EMAD ABOUEL NASR<sup>3,4</sup>

<sup>1</sup>School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China

<sup>2</sup>Institute of Systems Security and Control, College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China

<sup>3</sup>Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia

<sup>4</sup>Mechanical Engineering Department, Faculty of Engineering, Helwan University, Cairo 11732, Egypt

Corresponding author: Zhenhua Yu (zhenhua\_yu@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61873277 and 71571190 and the Key Research and Development Program of Shaanxi Province under Grant Nos. 2019GY-056 and 2018ZDXM-GY-036.

**ABSTRACT** Automated manufacturing systems (AMSs) are prone to be in deadlock states when resource allocation is unreasonable. Reasonably allocating system resources to achieve deadlock control is a primary task of the design of an AMS. This paper proposes a multi-step look-ahead deadlock prediction method to obtain an optimal deadlock avoidance policy for a class Petri nets, without calculating a complete reachability graph. As the reachability graph of a large-scale Petri net model is usually large, the analysis and calculation process to obtain the optimal deadlock avoidance policy is complicated. We first simplify a Petri net model by using the existing Petri net reduction methods or removing non-shared resources to simplify the model structure. Then we calculate the dead zone markings of the simplified model through a reverse generation method, and finally develop a multi-step look-ahead deadlock prediction method to obtain an optimal deadlock avoidance policy for a class of AMSs.

**INDEX TERMS** Petri net, reduction method, deadlock control, multi-step look-ahead deadlock prediction.

## I. INTRODUCTION

The vigorous development of science and technologies has profoundly changed our human society, especially the widespread application of computer technology, automation technology, information technology and related basic sciences in real life. In the face of fierce competition and uncertain changes in the market, traditional commodity production methods cannot meet market demand. For this reason, automated manufacturing systems (AMSs) [1]–[4] have emerged and have been continuously improved and developed. AMSs can be classified as a discrete event system because of its event-driven characteristics [5]–[8]. In the production practice, an AMSs is very large, and it is a computer control system composed of multiple interacting subsystems or modules. The interaction between the subsystems or modules in a system is manifested as a competition for limited system

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

resources [9], [10]. This will cause some undesirable situations, such as the occurrence of deadlocks, may cause serious consequences, and sometimes a system cannot run normally and the expected goal cannot be achieved, in the case of unreasonable resource allocation. Therefore, the deadlock control of a system is the focus of research [11], [12]. At present, extensive studies have been done on the problem of system deadlock [12]–[17].

As a mathematical model abstracted from an AMS, Petri nets [18]–[21] describe a system with simple and intuitive graphics, and are widely used in modeling, analysis and control of AMSs. Petri nets can be used to investigate a deadlock control policy of AMSs in a mathematical way [22], [23]. At present, the deadlock analysis and control of AMSs based on Petri net modeling mainly have the following two ways: (1) obtain a deadlock avoidance policy, based on the reachability graph analysis and the theory of regions [24]–[28]; (2) add control places to make a system deadlock-free [29]–[31], based on the structural analysis

[32], [33] of Petri nets in order to avoid clearing the strict minimum siphons [34].

For the early research on deadlock control of AMSs, the reachability graph analysis of Petri net model is widely used to obtain a system deadlock control method. Uzam [25] has studied a design method of the optimal active controller using the theory of regions. It divides the reachability graph of an uncontrolled system into a deadlock zone (DZ) and a non-deadlock zone (LZ), but this method is restricted by the state explosion problem [19]. After Uzam's research work, in 2003 Ghaffari *et al.* [24] designed an optimal deadlock controller using the theory of regions, which can obtain the maximum permissive behavior of a system. However, it is necessary to determine all legal markings of the Petri net to be controlled. Li *et al.* [35] propose a two-stage deadlock control policy combining the theory of regions and siphon control. The calculation time of this method is much shorter than that of using the theory of regions alone. Uzam and Zhou [36], [37] combine an iterative method with the theory of regions to obtain a system's deadlock control policy. In their research, the first bad marking (FBM) is defined in the DZ markings, which transforms the deadlock control into prohibiting dangerous markings in the LZ from reaching the FBM. This conversion requires multiple calculations of the reachability graph, which is not efficient. Chen and Li [38] use the vector covering method to find the minimum  $\mathcal{M}_{FBM}$  and the minimum dangerous marking set related to FBM, and design a deadlock controller to prohibit the transition from the dangerous markings to the first bad markings, but the calculation is complicated.

The Petri net's special structures, such as invariants [39], [40], siphons [30], resource transition circuits [45] are often closely related to the model activity. It is generally believed that there is a strict minimal siphon being cleared at a certain marking, which means that the system is deadlocked. Based on structural analysis of deadlock control policy such as siphon control, preventing strict minimum siphons from being emptied can achieve the purpose of deadlock control of a system. In 1995, Ezpeleta *et al.* [41] defined a simple sequential process system with resources ( $S^3PR$ ) as a type of Petri nets. Adding a control place to the siphon of the  $S^3PR$  net can prevent the siphon from being emptied. Although an  $S^3PR$  is more flexible in system modeling, in the face of complex and multi-process systems, the structural complexity of modeling with  $S^3PR$  is high. Furthermore, the model contains a large number of siphons.

The design of siphon-based deadlock controllers requires solving siphons. To solve the problem of high computational complexity of siphons, Chu and Xie [42] propose a Mixed Integer Programming (MIP) method for computing siphons. A maximal empty siphon of the model is solved by a mathematical planning method. Although the computational complexity of the MIP method for calculating siphons is still high, it is improved in computing efficiency compared with the method of traversing all siphons and state set methods. Huang *et al.* [33] propose to use MIP to iteratively calculate

the maximal empty siphon, and finally calculates all strict minimal siphons to add control places. Li and Zhou [43], [44] propose the concept of basic and dependent siphons in Petri nets to reduce the structural complexity, but the basic siphons can only be calculated after exhausting all strictly minimal siphons. Therefore this method does not improve the system deadlock controller design efficiency.

The reachability graph analysis method and structure analysis method of Petri net have their own advantages and disadvantages. Chao *et al.* [46], [47] proposes a multi-step forward deadlock prediction method based on structural analysis to obtain an optimal deadlock avoidance policy for the model. When determining the number of steps required for deadlock prediction, it is key to calculate the markings that no strict minimum siphons are cleared in the DZ. This requires the use of the reachability graph algorithm to calculate the complete reachability graph of the model, which limits the application of this method because the efficiency of calculating reachability graphs is often very low in the face of complex models. In this study, the reverse generation method is used to calculate the DZ markings, avoiding the calculation of the complete reachability graph. At the same time, the combination of model reduction rules and multi-step look-ahead deadlock prediction methods improves the efficiency of deadlock analysis methods in the literature [46], [47].

The remainder of the paper is arranged as follows. Section II reviews some basics of Petri net and  $S^3PR$ . Section III combines the multi-step look-ahead deadlock prediction method with model reduction rules, and uses the reverse generation method to calculate the simplified model dead zone markings, and finally determines the number of steps required for deadlock prediction. In Section IV, we use a multi-step look-ahead deadlock prediction method for a class of  $S^3PR$  to obtain the optimal deadlock avoidance policy. In Section IV, we reach some conclusions of this study and point out the work that should be done in the future.

## II. BASIC DEFINITIONS

### A. BASIC DEFINITIONS OF PETRI NETS

A Petri net model is a four-tuple  $N = (P, T, F, W)$ , the non-empty finite set  $P = \{p_1, p_2, \dots, p_n\}$  and  $T = \{t_1, t_2, \dots, t_m\}$  representing the collections of places and transitions, respectively, and  $P \cap T = \emptyset$ .  $F \subseteq (P \times T) \cup (T \times P)$  is a set of directed arcs from places to transitions or from transitions to places. Each directed arc has a weight. The mapping  $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$  is called weight function that assigns weights to directed arcs, that is, if  $f \in F$ ,  $W(f) > 0$ ; otherwise  $W(f) = 0$ , where  $\mathbb{N}$  is a set of non-negative integers. From the perspective of graph theory, a Petri net is a dichotomous directed graph.

A marking  $M$  of a Petri net  $N = (P, T, F, W)$  can be seen as a mapping:  $P \rightarrow \mathbb{N}$ .  $(N, M_0)$  represents a net system or an marked net with  $M_0$  as the initial marking. For simplicity, a Petri net  $(P, T, F, W, M_0)$  with initial marking  $M_0$  is denoted as  $(N, M_0)$ . Let  $p \in P$  be a place of  $(N, M_0)$ ,  $p$  is said marked at  $M$  if  $M(p) > 0$ . A set of places  $D \subseteq P$  is marked at

$M$  if exists a place in  $D$  is marked, that is,  $p \in D, M(p) > 0$ .  $M(D) = \sum_{p \in D} M(p)$  is the total number of tokens in  $D$  at  $M$ .

Let  $x \in P \cup T$  be a node of a Petri net  $N = (N, M_0)$ .  $\bullet x$  and  $x^\bullet$  represent the preset and postset of  $x$ , respectively, where  $\bullet x = \{y \in P \cup T | (y, x) \in F\}$  and  $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$ . When the node  $x$  be a place (transition),  $\bullet x$  and  $x^\bullet$  denotes the set of input and output transitions (place) of  $x$ . According to the structural characteristics of the Petri net, the preset and postset of arbitrary place are non-empty sets containing transitions, and the preset and postset of arbitrary transition are non-empty sets containing places.

Let  $N = (P, T, F, W)$  be a Petri net. A transition  $t \in T$  is enabled at  $M$  if for all  $p \in \bullet t$ ,  $M(p) \geq W(p, t)$ , denoted by  $M[t]$ . An enabled transition  $t$  can fire and its firing transfers the Petri net to a new marking  $M'$  such that for all  $p \in P$ ,  $M'(p) = M(p) - W(p, t) + W(t, p)$ , which is denoted as  $M[t]M'$ . A Petri net is said to be free of selfloop if there do not exist a place  $p$  and a transition  $t$  such that  $(p, t) \in F$  and  $(t, p) \in F$ . A self-loop-free Petri net can be represented by an incidence matrix  $[N](p, t) = W(t, p) - W(p, t)$  that is an integer matrix indexed by  $P$  and  $T$ .

Marking  $M'$  is reachable from  $M_1$  if there exist a feasible firing sequence of transitions (transition sequence for the sake of simplicity)  $\sigma = t_1 t_2 \dots t_n$  and markings  $M_2, \dots, M_n$  such that  $M_1[t_1]M_2[t_2] \dots M_n[t_n]M'$  holds. Given a Petri net  $(N, M_0)$ , the set of markings generated from  $M_0$  is called the reachability set of  $(N, M_0)$ , denoted by  $R(N, M_0)$ .

## B. S<sup>3</sup>PR MODELS

This section reviews the primary notions and properties of the system of simple sequential processes with resources, called S<sup>3</sup>PR, defined from the standpoint of Petri nets. It represents an important net type that can model a large class of automated manufacturing systems. This class of Petri nets has been extensively studied, due to its generality, perfect structural and behavioral properties.

**Definition 1:** A simple sequential process (S<sup>2</sup>P) is a Petri net  $N = (P_A \cup \{p^0\}, T, F)$  satisfying the following statements:

- (1)  $P_A \neq \emptyset$  is called the set of the activity (operation) places;
- (2)  $p^0 \notin P_A$  is called the process idle place or idle place;
- (3)  $N$  is a strongly connected state machine;
- (4) every circuit of  $N$  contains the place  $p^0$ ;

**Definition 2:** An S<sup>2</sup>P with resources (S<sup>2</sup>PR) is a Petri net  $N = (P_A \cup \{p^0\} \cup P_R, T, F, W)$ , satisfying

- (1) The subnet generated from  $X = P_A \cup \{p^0\} \cup T$  is an S<sup>2</sup>P;
- (2)  $P_R \neq \emptyset, P_R \cap (\{p^0\} \cup P_A) = \emptyset$ ;
- (3)  $\forall p \in P_A, \forall t \in \bullet p, \forall t' \in p^\bullet, \exists r_p \in P_R, \bullet t \cap P_R = t' \cap P_R = \{r_p\}$ ;
- (4)  $\forall r \in P_R, \bullet \bullet r \cap P_A = r \bullet \bullet \cap P_A \neq \emptyset; \forall r \in P_R, \bullet r \cap r^\bullet = \emptyset$ ;
- (5)  $\bullet \bullet (p^0) \cap P_R = (p^0) \bullet \bullet \cap P_R = \emptyset$ ;

**Definition 3:** Given an S<sup>2</sup>PR  $N = (P_A \cup \{p^0\} \cup P_R, T, F)$ , an initial marking  $M_0$  is said to be acceptable for  $N$  if:

- (1)  $M_0(p^0) \geq 1$ ;
- (2)  $M_0(p) = 0, \forall p \in P_A$ ;
- (3)  $M_0(r) \geq 1, \forall r \in P_R$ .

**Definition 4:** An S<sup>3</sup>PR i.e., a system of S<sup>2</sup>PR can be defined recursively as follows:

- (1) An S<sup>2</sup>PR is an S<sup>3</sup>PR.

(2) Let  $N_i = (P_{A_i} \cup \{p_i^0\} \cup P_{R_i}, T_i, F_i, W_i), i \in \{1, 2\}$ , be two S<sup>3</sup>PR, satisfying  $P_{R_1} \cap P_{R_2} = P_C \neq \emptyset, (P_{A_1} \cup \{p_1^0\}) \cap (P_{A_2} \cup \{p_2^0\}) = \emptyset, T_1 \cap T_2 = \emptyset$ .  $N_1$  and  $N_2$  can be combined into  $N = (P_A \cup \{p^0\}, T, F)$  through the shared resource  $P_C$ , and  $N$  is still an S<sup>3</sup>PR model, defined as  $P_A = P_{A_1} \cup P_{A_2}, \{p^0\} = \{p_1^0\} \cup \{p_2^0\}, P_R = P_{R_1} \cup P_{R_2}, T = T_1 \cup T_2$ , and  $F = F_1 \cup F_2$ .

Given a resource  $r \in P_R$  in an S<sup>3</sup>PR, the set of holders of  $r$  is denoted as  $H(r) = (\bullet \bullet r) \cap P_A$ . For a siphon  $S$  in an S<sup>3</sup>PR,  $S = S^R \cap S^A$ , where  $S^R = S \cap P_R$  and  $S^A = S \cap P_A$ .

## C. AUTOMATA THEORY

**Definition 5 ([49]):** The characters in a finite alphabet collection  $E$  can constitute a string, and the language is a collection of strings. The set of all strings fixed on the alphabet  $E$  is denoted as  $E^*$ .

**Definition 6 ([49]):** A deterministic finite automaton (DFA) is a 5-tuple.

$$G = (X, E, \delta, x_0, X_m)$$

where:

- $X$  is a finite set of states;
- $E$  is an alphabet;
- $\delta : X \times E \rightarrow X$  is a transition function;
- $x_0 \in X$  is an initial state;
- $X_m \subseteq X$  is a set of final states (or marked states).

**Definition 7 ([49]):** Let  $G_1 = (X_1, \delta_1, f_1, x_{01}, X_{m1})$  and  $G_2 = (X_2, \delta_2, f_2, x_{02}, X_{m2})$  be two DFA. Their synchronous system is a new DFA, denoted as  $G = (X, \delta, f, x_0, X_m)$ , where its generating language is  $L(G) = L(G_1) || L(G_2), L_m(G) = L_m(G_1) || L_m(G_2)$ , and  $\delta = \delta_1 \cup \delta_2$ .

## III. OPTIMAL DEADLOCK AVOIDANCE POLICY

### A. MODEL REDUCTION RULES OF PETRI NETS

A Petri net reduction rules ensure that some subnets or structures in the model retain relevant attributes, e.g., boundedness and reversibility, such that it is possible to derive the properties of the original Petri net model through the simplified model. In the deadlock analysis of Petri nets, using model reduction rules to convert a given model into a simple model is an important means to improve the efficiency of deadlock control. In this section, we introduce some Petri net reduction rules [49] to simplify a large Petri net model to obtain an optimal deadlock prevention policy for AMSs. These reduction rules are (a) fusion of series places, (b) fusion of series transitions, (c) fusion of parallel places, (d) fusion of parallel transitions, (e) elimination of self-loop places, and (f) elimination of self-loop transitions, which are shown in Figure 1.

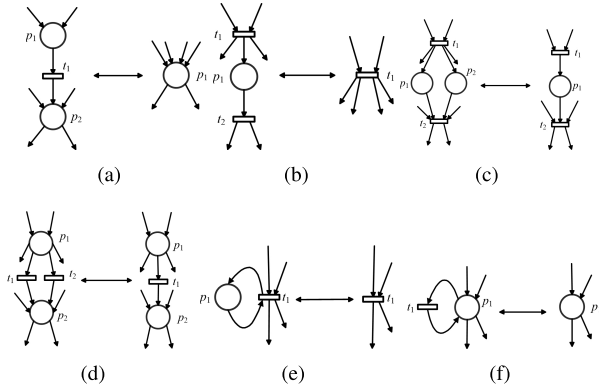


FIGURE 1. Petri net reduction rules.

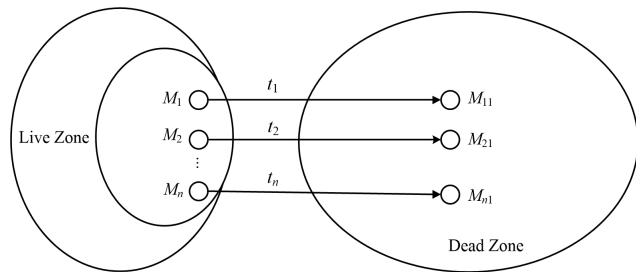


FIGURE 2. Principle analysis of one-step look-ahead deadlock prediction method.

**B. MULTI-STEP LOOK-AHEAD DEADLOCK PREDICTION METHOD**

A multi-step forward deadlock prediction method is a dynamic and online deadlock detection method. The main idea is to determine the safety of reachable states after the current state fire-enabled transition or transition sequence, that is, to detect in advance whether the system will enter deadlock.

If an  $S^3PR$  net structure does not contain the  $\xi$ -resource, there are only safe markings and deadlock markings in the reachable graph [51], the system can reach a deadlock marking by firing a transition at a dangerous marking. There are no strict minimal siphons cleared at the safe markings, and those markings can return to the initial marking. As shown in Figure 2, by firing transition  $t_i$  the dangerous marking  $M_i$  can reach a DZ marking  $M_{i1}$ , at which exists strict minimal siphon cleared, where  $i \in \{1, 2, \dots, n\}$ . This means that the dangerous marking  $M_i$  only needs to fire one transition can detect the deadlock.

When an  $S^3PR$  contains a  $\xi$ -resource, there are a certain number of markings, at which without strict minimal siphons being emptied, in the DZ. In this paper, this type of markings is defined as pseudo safe marking, to distinguish it from other markings.

*Definition 8* ([47]): Given an  $S^3PR(N, M_0)$  and a set of strict minimal siphons  $\Pi$ , a marking  $M \in DZ$  is said to be a pseudo safe marking, if for arbitrary strict minimal siphon  $S \in \Pi, M(S) > 0$ . The set of pseudo safe markings is denoted by  $M_{PSM}$ .

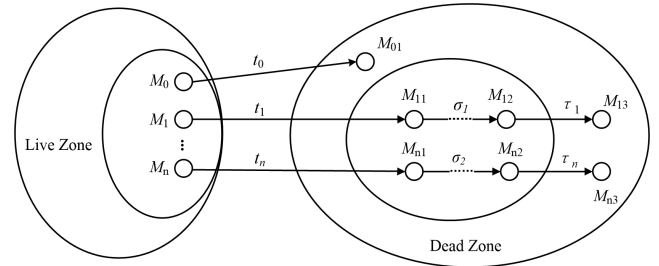


FIGURE 3. Principle analysis of multi-step look-ahead deadlock prediction method.

According to the relationship between siphon emptiness and deadlock, it is generally considered that if the system reaches a certain marking and there is a strict minimal siphon being emptied, then the system is said to be deadlocked. However, if the system reaches a pseudo safe marking, it is impossible to detect the deadlock based on whether the strict minimal siphon has been cleared. If the system reaches a pseudo safe marking, the system is actually deadlocked since the pseudo safe marking is a DZ marking.

To prevent a system from entering a deadlock without detecting it, this paper proposes a multi-step look-ahead deadlock prediction method. That is, a system arrives at a new marking, and fires a certain length of enable transition sequence to detect the security of the reachable marking. When the system arrives at a dangerous marking, through firing an enabled transition sequence in advance, it can reach a marking at which a strict minimal siphon is emptied in the DZ through the pseudo safe markings. This deadlock prediction method avoids the problem that the deadlock cannot be found. The following is a detailed analysis of the principle of multi-step forward deadlock prediction, to illustrate the principle and significance of the method of multi-step look-ahead deadlock prediction.

Figure 3 shows an  $S^3PR$ 's reachability markings divided into live and dead zones according to the theory of regions. In the figure, the dangerous marking  $M_0$  fires transition  $t_0$  can reach the dead zone marking  $M_{01}$ , at which exist strict minimal siphon being emptied. The dangerous marking  $M_i$  by firing transition  $t_i$ , can reach pseudo safe marking  $M_{i1}$ ,  $i \in \{1, 2, \dots, n\}$ . After  $M_{i1}$  fires transition sequence  $\sigma_i$  and transition  $\tau_i$  can reach the DZ marking  $M_{i3}$ , with a strict minimal siphon being cleared. Thus,  $M_{01}$  and  $M_{i3}$  are markings that can be used to determine whether a deadlock occurs in the system based on whether a strict minimal siphon has been cleared. This means that the markings  $M_0$  and  $M_i$  can reach the markings  $M_{01}$  and  $M_{i3}$  by firing transition sequences of lengths 1 and  $|\sigma_i| + 2$ , respectively. Under the markings  $M_{01}$  and  $M_{i3}$ , a deadlock can be detected based on whether the strict minimal siphon has been cleared.

According to the above analysis, in order to ensure that the deadlock can be detected under each dangerous marking, it is necessary to find the longest transition sequence in  $\{t_1\sigma_1\tau_1, t_2\sigma_2\tau_2, \dots, t_n\sigma_n\tau_n\}$ . The corresponding model uses a multi-step look-ahead deadlock prediction method, and the

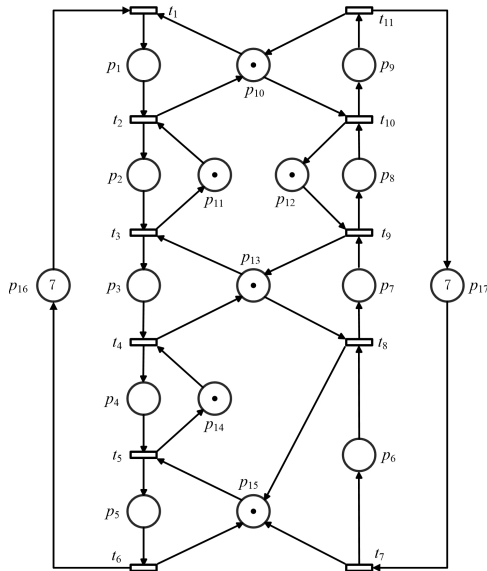


FIGURE 4. Petri net model (PNM) of an AMS (an  $S^3PR$  net).

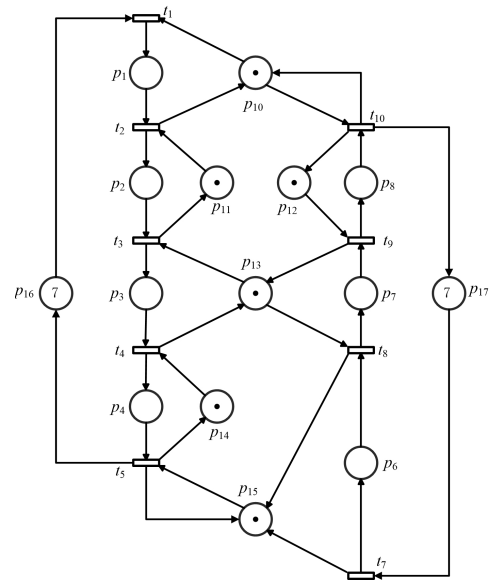


FIGURE 5. Reduced Petri net model (RPNM).

number of steps required to predict the deadlock is equal to the length of the longest transition prediction sequence.  $K$  represents the number of steps required for model deadlock prediction, and  $K = \max\{1, |t_1\sigma_1\tau_1|, |t_2\sigma_2\tau_2|, \dots, |t_n\sigma_n\tau_n|\}$ .

The above analysis shows that the number of steps required for model deadlock prediction is determined by the pseudo safe marking. Therefore, the calculation of false security signs is the key to the application of multi-step forward deadlock prediction methods. For a given Petri net model, in order to improve the efficiency of calculating pseudo safe markings, this paper uses model reduction rules to reduce the complexity of the model structure.

### C. ILLUSTRATIVE EXAMPLE

Figure 4 shows an  $S^3PR$  net. In this model there are 17 places,  $P = \{p_1, p_2, \dots, p_{17}\}$ , and 11 transitions,  $T = \{t_1, t_2, \dots, t_{11}\}$ , where  $p_{10} - p_{15}$  are resource places. Every resource place has only one token at the initial marking since this resource configuration does not affect the steps to look ahead.

The model can be simplified using the reduction rule (b) shown in Figure 1, and the simplified model shown in Figure 5 can be obtained through the following steps. First, we remove places  $p_5$  and  $p_9$ , and transitions  $t_6$  and  $t_{11}$ . Second, we remove the flow relationships  $f_1 = (t_5, p_5)$ ,  $f_2 = (p_5, t_6)$ ,  $f_3 = (t_6, p_{16})$ ,  $f_4 = (t_6, p_{15})$ ,  $f_5 = (t_{10}, p_9)$ ,  $f_6 = (p_9, t_{11})$ ,  $f_7 = (t_{11}, p_{10})$ , and  $f_8 = (t_{11}, p_{17})$ . Third, we increase the flow relationship  $f_9 = (t_5, p_{15})$ ,  $f_{10} = (t_5, p_{16})$ ,  $f_{11} = (t_{10}, p_{10})$ , and  $f_{12} = (t_{10}, p_{17})$ . The simplified model is shown in Figure 5.

When analyzing the Petri net model show in Figure 4 and its reduced version, we can see that there are 196 and 96 markings in their reachability graph, respectively, and they contain four identical deadlock markings. This shows that the

use of model simplification rules reduces the complexity of the structure and effectively reduces the number of states of the reachability graph. At the same time, the simplified model retains the original model's properties of interest. The efficiency of deadlock analysis can be improved if model reduction rules are used to simplify the model structure. By calculating the reachability graphs of the models shown in Figures 4 and 5, three identical pseudo safe markings can be obtained, namely  $M_1 = p_1 + p_2 + p_3 + p_6 + p_8 + p_{14} + 7p_{16} + 8p_{17}$ ,  $M_2 = p_1 + p_2 + p_4 + p_6 + p_8 + p_{13} + 7p_{16} + 8p_{17}$ , and  $M_3 = p_1 + p_2 + p_4 + p_6 + p_7 + p_{12} + 7p_{16} + 8p_{17}$ . At the same time, we have  $M_1[t_4]M_2$ ,  $M_3[t_9]M_2$ . The results show that the model  $(N, M_0)$  is transformed into  $(N', M'_0)$  by using the model reduction rules, and the reachable markings of  $(N, M_0)$  and  $(N', M'_0)$  have the same pseudo safe markings. If  $(N', M'_0)$  is used to calculate the pseudo safe markings of  $(N, M_0)$ , the calculation efficiency can be improved.

In this paper, in order to avoid calculating the complete reachability graph of a model, the reverse generation method is used to calculate the DZ markings of the simplified model, and the pseudo safe markings are obtained from the DZ markings.

In [48], the method of inverse calculation of the DZ markings of the model is studied. This method directly calculates the DZ markings to avoid calculating the complete reachability graph. In this paper, in order to avoid obtaining pseudo safe markings and calculate the complete reachability graph, this method is used to improve the efficiency of deadlock analysis. The following introduces the reverse generation method to calculate the DZ markings of the Petri net model.

### D. THE FIRST STEP

Given a Petri net model  $(N, M_0)$  that has  $n$  places, the marking  $M$  of  $N$  can be defined as a mapping from

$P \rightarrow \mathbb{N}$ . The token number held by place  $p_i$  at  $M_0$  is  $M_0(p_i)$ ,  $i \in \{1, 2, 3, \dots, n\}$ . By combining the branch and bound method and integer 0-1 programming, the  $M$  marking is decomposed into  $n$  parts. The  $i$ -th part of  $M$  represents the token number  $M(p_i)$ . According to binary coefficients and Boolean algebra,  $M(p_i)$  is expressed in the following form:

$$M(p_i) = 2^0 x_{i,1} + 2^1 x_{i,2} + 2^2 x_{i,3} + \dots + 2^{n_i-1} x_{i,m} \quad (1)$$

where  $x_{i,1}, x_{i,2}, \dots, x_{i,m} \in \{0, 1\}$ . This means that the token number held by each place under the  $M$  marking is expressed linearly by a binary number, and any reachable marking can be represented by this method.

Suppose that  $N$  has  $k$  P-invariants,  $v$  strict minimal siphons. Through the information of the model, an integer linear programming, namely DM, can be established to calculate all the deadlock markings of  $(N, M_0)$ .

The initial marking of  $N$  can be replaced by the above method as the objective function of linear programming,  $k$  P-invariants and  $v$  strict minimal siphons as constraints to establish integer 0-1 linear programming DM. During the iteration process, when the current strict minimal siphon is cleared and there is no feasible solution, we can replace other strict minimal siphons and continue to iterate until each strict minimal siphon is cleared and there is no feasible solution. After the above calculation process, all the deadlock markings can be obtained.

$$\min f = \sum_{M(p_i), p_i \in P} 2^0 x_{i,1} + 2^1 x_{i,2} + 2^2 x_{i,3} + \dots + 2^{m_i-1} x_{i,m} \quad (2)$$

where  $x_{i,1}, x_{i,2}, \dots, x_{i,m} \in \{0, 1\}$

$$\text{subject to : } \begin{cases} I_J^T \cdot M = I_J^T \cdot M_0 \\ M(S_j) = 0, M(S_h) \geq 0, h \neq j \\ \sum_{x_{i,k} \in SL_i} x_{i,k} + \sum_{x_{i,l} \in SL_i} (1 - x_{i,l}) \leq \\ \sum_{p_i \in P} n_i - 1, x_{i,k} = 1, x_{i,l} = 0 \\ J \in \{1, 2, \dots, k\} \\ j, h \in \{1, 2, \dots, v\} \\ i \in \{1, 2, \dots, n\} \end{cases} \quad (3)$$

where  $I_J$  is a P-invariant.

### E. THE SECOND STEP

The DZ markings will inevitably reach deadlock markings. According to this feature, other DZ markings can be calculated inversely through deadlock markings. Deadlock markings are used as the initial data of the DZ markings to reversely enable calculation of other dead zone markings. During the calculation process, if the set of post-marking of a certain marking is included in the known DZ markings, this marking is the newly calculated DZ markings; otherwise it is the pending DZ marking. In the next calculation, we first calculate the previous set of DZ markings for the last new

### Algorithm 1 Reverse Generation Algorithm of DZ Marking

**Input:** Deadlock marking set DL of a net system  $(N, M_0)$ ;  
**Output:** DZ markings

- 1:  $DZ = [], M_{new} = [], \mathcal{M}_{pre} = [], \mathcal{M}_{DD} = [];$
- 2: **if** DL == [] **then**
- 3:   Stop
- 4: **end if**
- 5:  $M_{new} = DL$
- 6: **while**  $M_{new} != []$  **do**
- 7:    $DZ = DZ \cup M_{new}, M'_{new} = []$
- 8:    $\mathcal{M}_{pre} = \{x | x = \bullet M, \forall M \in M_{new}\}$
- 9:   Calculate  $\mathcal{M}_{post} = \{x \bullet | x \in \mathcal{M}_{pre}\}$  for each marking  $M_{pre}$  in  $\mathcal{M}_{pre}$ , if  $\mathcal{M}_{post} \subseteq DZ, M'_{new} = M'_{new} \cup M_{pre}$ , otherwise  $\mathcal{M}_{DD} = \mathcal{M}_{DD} \cup M_{pre}$
- 10:    $M_{new} = M'_{new}$
- 11: **end while**  $M_{new} != []$
- 12:  $M''_{new} = [1]$
- 13: **while**  $M''_{new} != []$  **do**
- 14:    $M''_{new} = [], M_1 = []$
- 15:   Traverse the markings in  $\mathcal{M}_{DD}$ , and merge the DZ markings into  $M_1$
- 16:   Calculate  $\mathcal{M}_{pre} = \{x | x = \bullet M, \forall M \in M_1\}$  for all the markings in  $M_1$ .
- 17:    $\mathcal{M}_{DD} = \mathcal{M}_{DD} - M_1, \mathcal{M}_{DD} = \mathcal{M}_{DD} \cup \mathcal{M}_{pre}, DZ = DZ \cup M_1$
- 18:    $M''_{new} = M_1$
- 19: **end while**  $M''_{new} != []$
- 20: Stop = 0

calculation, determine the new DZ marking from the previous set of markings, and add the pending dead zone marking until there is no new DZ marking.

According to the deadlock makings calculated in the first step, the DZ making set of a model can be calculated by the following algorithm. The following is the symbol description in the algorithm:

$\mathcal{M}_{DD}$  indicates the set of pending DZ markings.

$M_{new}$  is the calculated DZ markings.

$\mathcal{M}_{pre}$  represents the set of preceding markings of  $M$ .

$\mathcal{M}_{post}$  is the postset marking set.

Through the above two calculation processes, by calculating the set of DZ markings of a given Petri net model, all DZ markings can be calculated without calculating the complete reachability graph.

## IV. DEADLOCK AVOIDANCE POLICY OF A CLASS OF LOOPED S<sup>3</sup>PR NETS

### A. CLASSIFICATION OF RESOURCES

A resource allocation system is composed of multiple interacting subsystems or execution units. The interaction of subsystems or execution units is mainly reflected as the competition for limited system resources. In the case of unreasonable system resource allocation, a system is prone to deadlock. In order to prevent the system from entering a

deadlock, it is necessary to reasonably allocate limited system resources to maximize system operation efficiency.

**Definition 9** ([46]): Given an  $S^3PR(N, M_0)$ , let  $S \in \Pi$  be a strict minimal siphon. If a resource  $r \in S^R$  of  $(N, M_0)$  satisfies arbitrary  $p \in H(r)$  and  $p \notin S_A$ , then  $r$  is called an independent resource. Otherwise  $r$  is called a dependent resource. Let  $S_{in}^R$  denote the set of independent resources in a siphon  $S$ .

In the field of deadlock control of Petri nets, according to whether they are related to the deadlock of a system, the independent resources of the system can be divided into weakly independent resources and strongly independent resources.

**Definition 10** ([46]): Given an  $S^3PR(N, M_0)$  with  $n$  strict minimum siphons  $S_1, S_2, \dots, S_n$ , let  $S_{in}^R$  denote the set of independent resources. Let  $S_{sin}^R = \{r \mid r \in (P_R \setminus \bigcup_{i=1}^n S_i^R)\}$

and  $S_{win}^R = S_{in}^R \setminus S_{sin}^R$  represent the sets of strongly independent resources and weakly independent resources of  $N$ , respectively.

**Definition 11** ([46]): A holder-resource circuit (HR-circuit) with respect to a resource  $r$  in an  $S^3PR$ , denoted by  $\mathcal{H}(r)$ , is a simple circuit if it contains only one resource  $r$  with  $|\mathcal{H}(r)| = 1$ , an activity place  $p \in \mathcal{H}(r)$ , and transitions.

If a resource corresponds to one holder-resource circuit only, the holder-resource circuit is said to be monoploid.

**Definition 12** ([46]): An  $S^3PR(N, M_0)$  is said to be unitary if there is only one  $\xi$ -resource. For each  $S \in \Pi$ ,  $r \in S_{in}^R$ , and  $r$  is associated with a monoploid holder-resource circuit.

A strongly monoploid HR-circuit  $\mathcal{H}(r)$  is denoted by  $\mathcal{H}^s(r)$  if  $r \in S_{sin}^R$ . This means that  $\mathcal{H}^s(r)$  contains strongly independent resources. Similarly,  $\mathcal{H}^w(r)$  denotes a weakly monoploid HR-circuit.

For example, Figure 6 demonstrates an  $S^3PR$ , where  $P_R = \{p_2, p_3, p_{11}, p_{12}, p_{13}\}$  is the set of resource place. The strict minimal siphon set of this model is denoted by  $\Pi = \{S_1, S_2, S_3\}$ , where  $S_1 = \{p_6, p_{10}, p_{11}, p_{12}\}$ ,  $S_2 = \{p_7, p_9, p_{12}, p_{13}\}$ , and  $S_3 = \{p_7, p_{10}, p_{11}, p_{12}, p_{13}\}$ . The activity places of the model in  $\Pi$  are  $p_6, p_7, p_9$  and  $p_{10}$ ; thus  $S_{in}^R = \{p_2, p_3\}$ . According to the definition of strongly independent resources,  $S_{sin}^R = \{p_2, p_3\}$ . Hence  $S_{win}^R = S_{in}^R \setminus S_{sin}^R = \emptyset$ . Moreover there are nine HR-circuits in this model, such as  $p_3t_2p_4t_3p_3$ ,  $p_2t_1p_1t_2p_2$ , and  $p_{13}t_8p_8t_9p_{13}$ . However, only  $p_2t_1p_1t_2p_2$ ,  $p_2t_1p_1t_4p_2$ , and  $p_3t_2p_4t_3p_3$  are three strongly monoploid HR-circuits, and there is no weakly monoploid HR-circuit in this model.

### B. OBTAINING THE DEADLOCK AVOIDANCE POLICY OF THE MODEL WITH A LOOP IN THE REACHABLE GRAPH

The model shown in Figure 6 uses the model simplification rules to remove the places  $p_7$  and  $p_{10}$ , while reducing flow relationships  $f_1 = (t_6, p_7), f_2 = (p_7, t_7), f_3 = (t_7, p_{13}), f_4 = (t_7, p_{14}), f_5 = (t_{10}, p_{10}), f_6 = (p_{10}, t_{11}), f_7 = (t_{11}, p_{11}),$  and  $f_8 = (t_{11}, p_{15})$ . In addition, the flow relationships  $f_9 = (t_6, p_{14}), f_{10} = (t_6, p_{13}), f_{11} = (t_{10}, p_{11}),$  and  $f_{12} = (t_{10}, p_{15})$  need to be added to the model structure. After the above

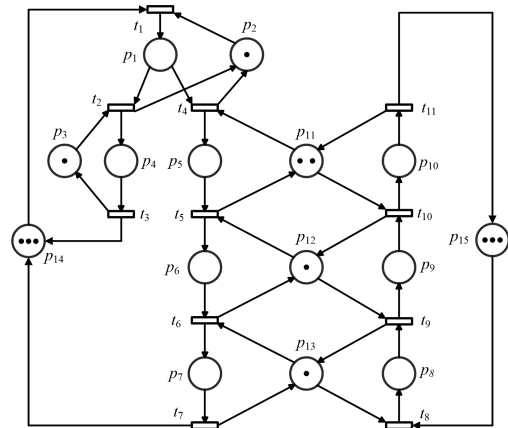


FIGURE 6. An  $S^3PR$  model.

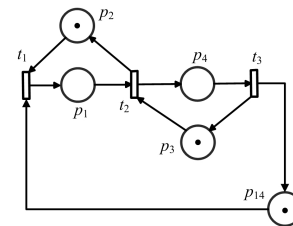


FIGURE 7. Subnet model A.

simplification process, this model can be transformed into a simple model, and the results are not given here.

In the model shown in Figure 6,  $p_2$  and  $p_3$  are strong independent resources that are not related to strict minimal siphons, and the subnet model that contains them, as shown in Figure 7, has nothing to do with the system's deadlock. Based on this fact, removing the subnet structure from the model during the deadlock analysis further reduces the structural complexity of the model and simplifies the necessary calculation process of the deadlock analysis to achieve the purpose of improving the calculation efficiency.

The simplified model of the Petri net shown in Figure 6 can be obtained by using the reduction rules of the model, and then the strongly independent resources can be removed from the simplified model to obtain the simplest model shown in Figure 8. For the convenience of discussion, the models shown in Figures 7 and 8 are called model A and model B, respectively.

Compared with the model shown in Figure 6, model B has a lower structural complexity. If the deadlock control policy of the former can be obtained through the deadlock analysis of the latter, this will improve the efficiency of obtaining the system's deadlock control policy.

**Definition 13:** Given an  $S^3PR$  net  $(N, M_0)$ ,  $r \in P$  is the  $\xi$ -resource of  $(N, M_0)$ , and  $\Pi$  is a strict minimal siphon set of  $(N, M_0)$ . Suppose a pseudo safe marking  $M \in \mathcal{M}_{PSM}$ , if  $M$  meets the following three conditions:

- (1) for arbitrary  $S \in \Pi$ ,  $p \in P \setminus \{r\}$ , there is  $M(p) = 0$ ,  $M(r) = 1$ .

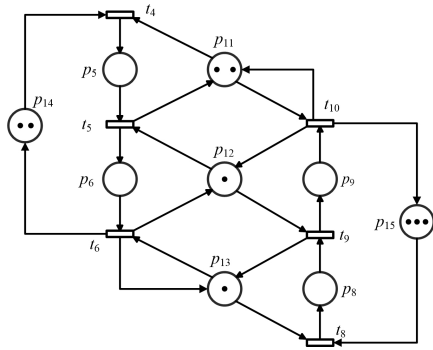


FIGURE 8. Subnet model B.

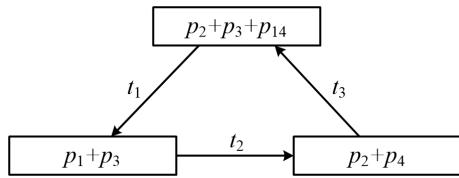


FIGURE 9. Reachability graph of subnet model A.

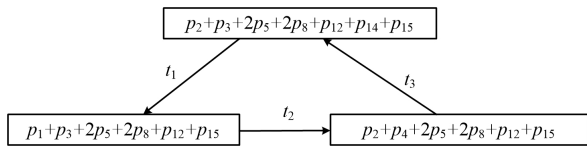


FIGURE 10. Pseudo safe marking.

(2) the set of post marking for  $M$  is  $M_{post}$ , for each marking  $M' \in M_{post}$ , there exists  $S \in \Pi$  makes  $M'(S) = 0$ .

(3) existing  $t \in T$  is enabled at  $M$ , if  $M[t]M'$ , at  $M'$  exists  $S \in \Pi$  makes  $M'(S) = 0$ .

$M$  is called a critical pseudo safe marking.

After the reverse generation method, it is calculated that model B has a pseudo safe marking  $M = 2p_5 + 2p_8 + p_{12} + p_{15}$ , which is a critical pseudo safe marking.

Model A has a simple structure and it is easy to calculate its three reachable markings as shown in Figure 9.

*Definition 14:* Let  $(N, M_0)$  be an  $S^3PR$  net, if the ring exists in the reachability graph of  $N$ , the constant  $\tau \in \mathbb{N}$  represents the number of cycles times in the ring when the net system runs, and  $\tau$  can be measured while the system is running.

Using these three markings and the pseudo safe marking of model B as synchronous operations, the pseudo safe markings of the model shown in Figure 6 can be calculated, as shown Figure 10.

There exists a ring among the three pseudo-security signs of the model shown in Figure 6. The worst case is that the system circulates integer times in the ring structure and traverses two signs when the ring structure is exited, so the number of steps required by the model for deadlock prediction is  $K = 1 + 3\tau + 2$ .

In the face of the model with the general structure shown in Figure 11, the above method can also be used to calculate

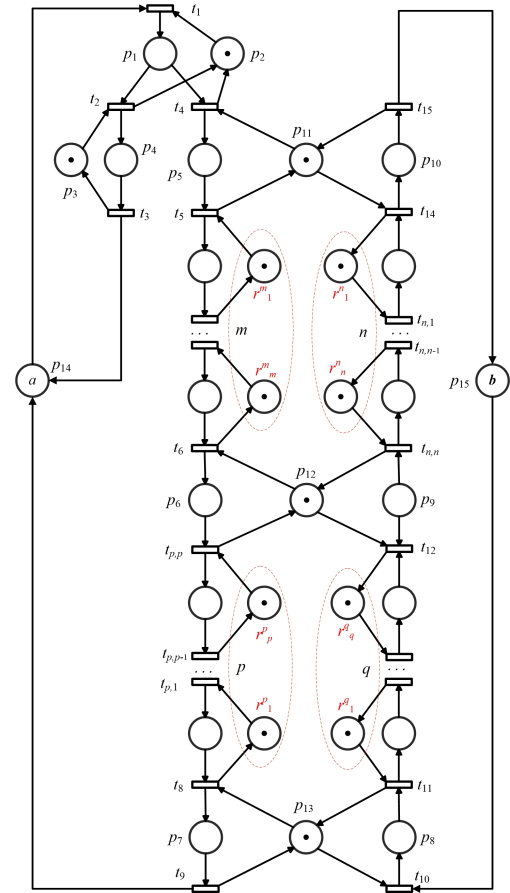


FIGURE 11. A Petri net model.

the number of steps required by the model for deadlock prediction. The structural parameters  $m, n, p,$  and  $q$  of this model can take any value,  $m, n, p, q \in \{0, 1, 2, \dots\}$ . For the model shown in Figure 6,  $m = n = p = q = 0$ .

When the structural parameters of the model take fixed values, the same method can also be used to obtain the number of steps required for deadlock prediction. First we use the rules to simplify the model, and then remove the strongly independent resources. Finally we can get the most simplified model of the model shown in Figure 12.

In fact, when the structure parameter takes a fixed value, the number of pseudo safe markings in the reachability graph can be determined, and the reverse generation method is easy to calculate those pseudo safe markings. For Figure 11, the following theorem is given.

*Theorem 1:* Let Petri net  $(N, M_0)$  be an  $S^3PR$  net with a  $\xi$ -resource, and the structural parameters be  $m, n, p,$  and  $q$  that represent the number of weakly independent resources. There are  $n + p + 1$  pseudo safe markings in the reachable markings of the model.

*Proof:* Suppose that its weakly independent resource set and pseudo safe markings set are  $\{r_1^m, r_2^m, \dots, r_m^m, r_1^n, r_2^n, \dots, r_n^n, r_1^p, r_2^p, \dots, r_p^p, r_1^q, r_2^q, \dots, r_q^q\}$  and  $\mathcal{M}_{PSM}$ , respectively.



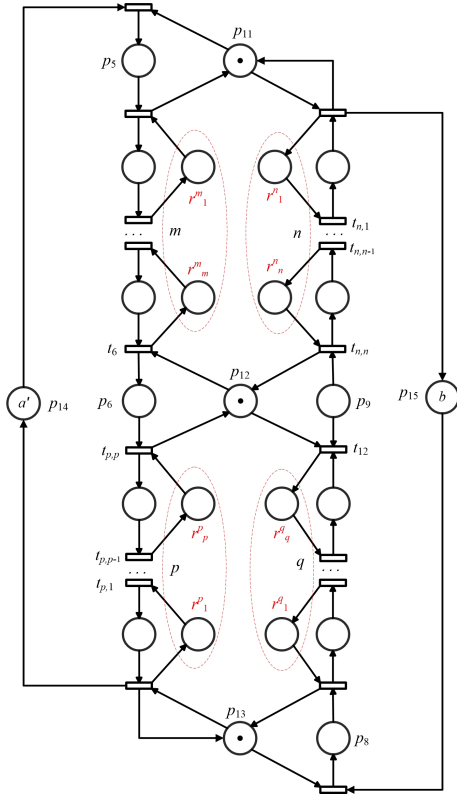


FIGURE 12. Model C.

According to the reachability graph analysis, the pseudo safe markings of this model are generated during the process of the strict minimal siphons  $S_1$  and  $S_2$  being emptied, where  $S_1 = \{p_6, p_{11}, r_1^m, r_2^m, \dots, r_m^m, r_1^n, r_2^n, \dots, r_n^n, p_{12}\}$ ,  $S_2 = \{p_9, p_{12}, r_1^p, r_2^p, \dots, r_p^p, r_1^q, r_2^q, \dots, r_q^q, p_{13}\}$ . There exist markings  $M_{n+1}^n$  and  $M_{p+1}^p$  that belong to  $\mathcal{M}_{PSM}$ , at marking  $M_{n+1}^n (M_{p+1}^p)$ , all other places except for  $r_1^n (r_1^p)$  in  $S_1 (S_2)$  are cleared, and  $M(r_1^n) = 1 (M(r_1^p) = 1)$ .

In order to clear  $S_1$ , at  $M_{n+1}^n$ , we fire the transition sequence  $t_{n,1} \dots t_{n,n-1} t_{n,n}$  to move the token of  $r_1^n$  to  $p_{12}$ , which can reach the critical pseudo safe marking  $M_1$ .  $S_1$  is cleared after transition  $t_{12}$  fires at  $M_1$ , that is, after transition sequence  $t_{n,1} \dots t_{n,n-1} t_{n,n} t_{12}$  fires at  $M_{n+1}^n$ ,  $S_1$  is cleared, and this process generates  $n+1$  pseudo safe markings. Similarly, at  $M_{p+1}^p$ ,  $t_{p,1} \dots t_{p,p-1} t_{p,p} t_6$  fires, corresponding to the process of  $S_2$  being emptied, which generates  $p+1$  pseudo safe markings, and reaches  $M_1$ . Therefore, the model has  $n+p+1$  pseudo safe markings. It should be noted that after  $M_{n+1}^n (M_{p+1}^p)$  evolves into  $M_1$ , at which there is no strict minimal siphon being cleared since the  $\xi$ -resource  $p_{12}$  is marked and fires  $t_6$  (or  $t_{12}$ ) and  $S_2$  (or  $S_1$ ) is emptied. In short, after  $M_1$  launches an enabling transition, there is a strict minimal siphon that is cleared of tokens. ■

Without loss of generality, the reachability graph corresponding to the pseudo safe markings is shown in Figure 13. If  $n = 0$  (or  $p = 0$ ), there are no pseudo safe markings  $M_2^n, M_3^n, \dots, M_{n+1}^n$  (or  $M_2^p, M_3^p, \dots, M_{p+1}^p$ ).

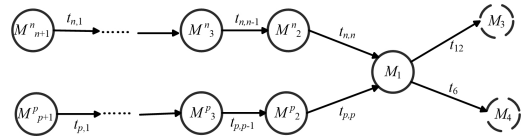


FIGURE 13. Pseudo safe markings of model C.

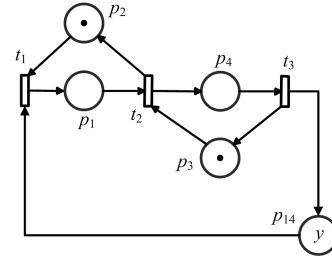


FIGURE 14. Model D.

Example 1: The model shown in Figure 5 with structural parameters  $m = 1, n = 1, p = 1, q = 0$  has three strict minimal siphons  $S_1 = \{p_3, p_{10}, p_{11}, p_{12}, p_{13}\}$ ,  $S_2 = \{p_7, p_{13}, p_{14}, p_{15}\}$ , and  $S_3 = \{p_{10}, p_{11}, p_{12}, p_{13}, p_{14}, p_{15}\}$ . For this model, we have calculated three pseudo safe markings  $M_1 = p_1 + p_2 + p_3 + p_6 + p_8 + p_{14} + 7p_{16} + 8p_{17}$ ,  $M_2 = p_1 + p_2 + p_4 + p_6 + p_8 + p_{13} + 7p_{16} + 8p_{17}$ , and  $M_3 = p_1 + p_2 + p_4 + p_6 + p_7 + p_{12} + 7p_{16} + 8p_{17}$ . At the same time, we have  $M_1[t_4]M_2, M_3[t_9]M_2$ . At marking  $M_1$ ,  $S_2$  is marked since  $M(p_{14}) = 1$ . After  $t_3$  fires at  $M_2$ , we reach a marking, at which  $S_2$  is cleared. That is, after the transition sequence  $t_4 t_3$  fires at  $M_1$ ,  $S_2$  is cleared. Similarly,  $S_1$  is cleared after the transition sequence  $t_9 t_8$  is transmitted at  $M_2$ . Therefore, there are three  $(n+p+1)$  pseudo safe markings in the reachable marking of this model.

Theorem 2: Let Petri net  $(N, M_0)$  be an  $S^3PR$  net with a  $\xi$ -resource, and  $\mathcal{M}_{PSM}$  be the pseudo safe markings set of  $N$ . There is only one critical pseudo safe marking in  $\mathcal{M}_{PSM}$ .

Proof: According to the proof of Theorem 1, we know that the two strict minimal siphons  $S_1 = \{p_6, p_{11}, r_1^m, r_2^m, \dots, r_m^m, r_1^n, r_2^n, \dots, r_n^n, p_{12}\}$  and  $S_2 = \{p_9, p_{12}, r_1^p, r_2^p, \dots, r_p^p, r_1^q, r_2^q, \dots, r_q^q, p_{13}\}$  of the model shown in Figure 12 generate the same critical false safety sign during the emptying process. Therefore, for a model containing only one  $\xi$ -resource, there is only one critical pseudo safe marking. ■

In the model shown in Figure 12, when the token number of an idle place ensures that  $m+p$  resource places are emptied,  $a'$  can be considered to be sufficiently large. When  $a'$  is sufficiently large, it will not affect the number of pseudo safe markings. It should be noted that  $a' + y = a$ .

For model D in Figure 14, when  $y = 1$  and  $y > 1$ , there are three and four reachable markings respectively, as shown in Figures 9 and 15. So far, we can calculate all pseudo safe markings of the original model shown in Figure 11 by using the synchronous calculation through the reachable markings of model D and the pseudo safe markings of model C.

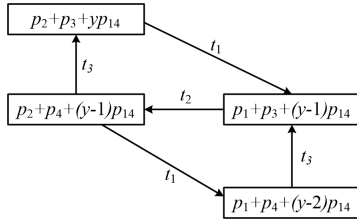


FIGURE 15. Reachability graph of model D.

Although there is a problem of state explosion when calculating a synchronous automation, the number of states involved in the calculation is small here. Compared with directly calculating the model shown in Figure. 11, the method adopted here effectively improves the calculation efficiency since it simplifies the model.

There is a ring structure between the reachable markings of this model, and the corresponding system operating cycle number in the ring structure is denoted by  $\tau$ , which is a measurable constant. For Figure 11, the following theorem is given.

*Theorem 3:* Let Petri net  $(N, M_0)$  be an  $S^3PR$  net with strongly independent resources, and the structural parameters be  $m, n, p$ , and  $q$  that represent the number of weakly independent resources. The number of steps required for deadlock prediction is  $K = 1 + k(3\tau + 2)$ , where  $k = \max\{n, p\}$ .

*Proof:* (1) We find that there are two paths between the pseudo safe markings, and the longest path is related to the structural parameters  $n$  and  $p$ . This conclusion can be obtained from the proof process of Theorem 1.

(2) A pseudo safe marking of the original model is calculated from the pseudo safe marking of the simplified model and the reachable identification of the subnet structure where a strong independent resource is located by using the synchronous product operation of the automata. There is a multi-layered ring structure between the pseudo safe markings of the original model, and each ring contains three markings.

It can be seen from (1) that the longest path existing between the pseudo safe markings of the original model has a  $k$ -layer ring structure, where  $k = \max\{n, p\}$ . The worst case is that the system arrives at the pseudo safe markings and visits  $\tau$  times at each layer of the ring, and then traverses two markings to leave the ring structure, that is, the number of steps required for each layer of the ring structure is  $3\tau + 2$ . A dangerous marking needs to fire a transition to reach a pseudo safe marking, so  $K = 1 + k(3\tau + 2)$ . ■

C. ILLUSTRATIVE EXAMPLE

When the structural parameters  $m, n, p, q$  of the model take fixed values, an optimal deadlock avoidance strategy of the corresponding model can be obtained by the above method.

In the model shown in Figure 16, the structural parameters are  $m = 1, n = 2, p = 1$ , and  $q = 0$ . To calculate the

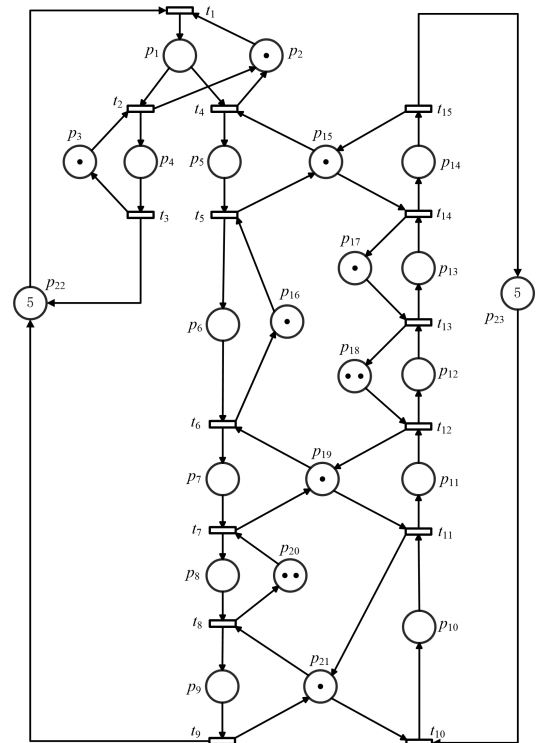


FIGURE 16. Instance model.

pseudo safe markings of this model, the following steps can be employed:

1. We use reduction rules to remove  $p_9, p_{14}, t_9, t_{15}$ , and then remove the subnet structure that contains strongly independent resources  $p_2, p_3$  to obtain its most simplified model. In the simplification process, let  $M(p_{22}) = 1$  in the subnet structure, where  $p_2$  and  $p_3$  are located.

2. According to the initial marking of this model, the P-invariants and the strict minimal siphons, an integer linear programming problem is established to obtain the dead marking set of the simplest model.

3. Taking the dead marking set of the simplest model as the initial data, the algorithm is called to calculate all the dead zone markings (DZ) of the simplest model. There are four pseudo safe markings to be obtained from DZ:  $M_1 = p_5 + p_6 + 2p_8 + p_{10} + 2p_{12} + p_{13} + p_{19} + p_{23}, M_2 = p_5 + p_6 + 2p_8 + p_{10} + p_{11} + p_{12} + p_{13} + p_{18} + p_{23}, M_3 = p_5 + p_6 + 2p_8 + p_{10} + p_{11} + 2p_{12} + p_{17} + p_{23}, M_4 = p_5 + p_6 + p_7 + p_8 + p_{10} + 2p_{12} + p_{13} + p_{20} + p_{23}$ , and we have  $M_3[t_{13}]M_2[t_{12}]M_1, M_4[t_4]M_1$ .

4. Obtain the pseudo safe markings of the simplest model from the DZ. These markings and the reachable markings of the subnet structure where  $p_2$  and  $p_3$  are located are calculated by synchronous product operations to calculate all pseudo safe markings of the model in Figure 16. The final calculation result is shown in Figure 17.

The subnet structure that contains strongly independent resources  $p_2$  and  $p_3$  can calculate three markings, namely  $M'_1 = p_2 + p_3 + p_{22}, M'_2 = p_1 + p_3, M'_3 = p_2 + p_4$ , and  $M'_1[t_1]M'_2[t_2]M'_3[t_1]M'_1$ . These three markings form a loop.

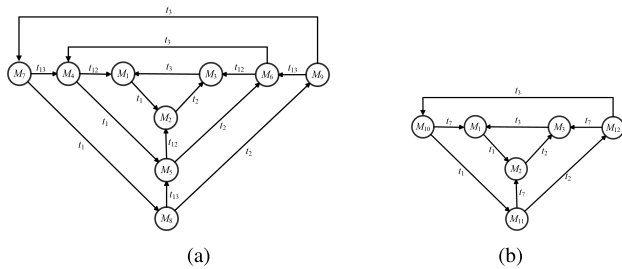


FIGURE 17. Pseudo safe markings of instance model.

It should be noted that the result is divided into two sub-graphs for convenience of drawing, and this graph can be calculated by INA. Therefore, the optimal deadlock avoidance policy of the model can be calculated, i.e.,  $K = 1 + 3(3\tau + 2)$ .

The above analysis results show that during the deadlock analysis of a given Petri net model, the model reduction rules can be used to simplify the model. On the basis of the simplified model, strongly independent resources, that is, the non-shared resources, can be further removed to fully simplify the model structure. In this paper, the method of obtaining the optimal deadlock avoidance strategy for a given Petri net model does not require direct analysis and calculation of the original model. The optimal deadlock avoidance strategy is obtained through its simplified model. It is easy to understand that simplifying the model can improve the efficiency of obtaining the optimal deadlock avoidance strategy. The multi-step look-ahead deadlock prediction method needs to analyze the model to obtain the number of steps required for the deadlock prediction.

Comparing with the existing methods, the above method of obtaining the optimal deadlock avoidance strategy of the model has the following advantages:

(1) The simplified rules of the model and the removal of strongly independent resources are used to reduce the number of reachability markings.

(2) Using the reverse generation method to directly calculate the DZ markings, we avoid the calculation of the complete reachability graph and improve the calculation efficiency.

(3) The optimal deadlock avoidance policy of the original model is calculated by the simplified model. The method is to use the synchronous product operation in the automata to calculate the marking related to the deadlock control, and the number of markings involved in the calculation is limited.

## V. CONCLUSION

In an  $S^3PR$  with an  $\xi$ -resource, there is a kind of markings belonging to the dead zone markings, at which do not exist the strict minimal siphons of been emptied. The number of steps required for the deadlock prediction method is determined by these markings, and the optimal deadlock avoidance policy of the model can be obtained by calculating these markings. In this paper, according to the model reduction rules, the inverse

generation method and the synchronous product operation, the multi-step look-ahead deadlock prediction method is proposed. The emergence of pseudo safe markings makes the multi-step look-ahead prediction method necessary. The method of calculating pseudo safe markings in this study is suitable for the existing one-step and multi-step look-ahead prediction methods. For a given Petri net model, if no pseudo safe marking is calculated, only one step look-ahead prediction method is needed to obtain the optimal deadlock avoidance policy. It is worth noting that other Petri net model simplification rules are combined with the multi-step look-ahead deadlock prediction method to make the multi-step look-ahead prediction method more effective. This method can be also applied in multi-agent systems [52] and cyber-physical systems [53], [54].

## ACKNOWLEDGMENT

The authors extend their appreciation to the Researchers Supporting Project number (RSP-2020/164), King Saud University, Riyadh, Saudi Arabia.

## REFERENCES

- [1] E. De Santis and S. Di Gennaro, "A representation of discrete event dynamical systems," in *Proc. 32nd IEEE Conf. Decis. Control*, Dec. 1993, pp. 1176–1181.
- [2] H. Chen, N. Wu, Z. Li, and T. Qu, "On a maximally permissive deadlock prevention policy for automated manufacturing systems by using resource-oriented Petri nets," *ISA Trans.*, vol. 89, pp. 67–76, Jun. 2019.
- [3] X. Zan, Z. P. Wu, C. Guo, and Z. H. Yu, "A Pareto-based genetic algorithm for multi-objective scheduling of automated manufacturing systems," *Adv. Mech. Eng.*, vol. 12, no. 1, pp. 1–15, 2020.
- [4] D. Sun, Y. Chen, M. A. El-Meligy, M. A. F. Sharaf, N. Wu, and Z. Li, "On algebraic identification of critical states for deadlock control in automated manufacturing systems modeled with Petri nets," *IEEE Access*, vol. 7, pp. 121332–121349, 2019.
- [5] Q. Chen, L. Yin, N. Wu, M. A. El-Meligy, M. A. F. Sharaf, and Z. Li, "Diagnosability of vector discrete-event systems using predicates," *IEEE Access*, vol. 7, pp. 147143–147155, 2019.
- [6] L. Yang, Z. Li, and A. Giua, "Containment of rumor spread in complex social networks," *Inf. Sci.*, vol. 506, pp. 113–130, Jan. 2020.
- [7] L. Yang, Z. Yu, M. A. El-Meligy, A. M. El-Sherbeenly, and N. Wu, "On multiplexity-aware influence spread in social networks," *IEEE Access*, vol. 8, pp. 106705–106713, 2020, doi: 10.1109/ACCESS.2020.2999312.
- [8] X. Li, E. S. Abouel Nasr, A. Mohammed El-Tamimi, and Z. Li, "Adaptive consensus of two coupled heterogeneous networked systems with bidirectional actions," *IEEE Access*, vol. 8, pp. 35832–35841, 2020.
- [9] E. G. Coffman, M. Elphick, and A. Shoshani, "System deadlocks," in *Proc. ACM Comput. Surv.*, vol. 3, no. 2, pp. 67–78, Jun. 1971.
- [10] M. Lawley and S. Reveliotis, "Deadlock avoidance for sequential resource allocation systems: Hard and easy cases," *Int. J. Flexible Manuf. Syst.*, vol. 13, no. 4, pp. 385–404, 2001.
- [11] M. Dotoli and M. P. Fanti, "Deadlock detection and avoidance strategies for automated storage and retrieval systems," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 4, pp. 541–552, Jul. 2007.
- [12] M. P. Fanti and M. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 1, pp. 5–22, Jan. 2004.
- [13] K. Barkaoui and I. Ben Abdallah, "A deadlock prevention method for a class of FMS," in *Proc. IEEE Int. Conf. Syst., Man Cybern. Intell. Syst. 21st Century*, Oct. 1995, pp. 4119–4124.
- [14] K. Barkaoui and I. B. Abdallah, "Deadlock avoidance in FMS based on structural theory of Petri nets," in *Proc. INRIA/IEEE Symp. Emerg. Technol. Factory Automat. (ETFA)*, Oct. 1995, pp. 499–510.
- [15] F. Basile, R. Cordone, and L. Piroddi, "Integrated design of optimal supervisors for the enforcement of static and behavioral specifications in Petri net models," *Automatica*, vol. 49, no. 11, pp. 3432–3439, Nov. 2013.

- [16] R. Cordone, A. Nazeem, L. Piroddi, and S. Reveliotis, "Designing optimal deadlock avoidance policies for sequential resource allocation systems through classification theory: Existence results and customized algorithms," *IEEE Trans. Autom. Control*, vol. 58, no. 11, pp. 2772–2787, Nov. 2013.
- [17] J. Ezpeleta and R. Valk, "A polynomial deadlock avoidance method for a class of nonsequential resource allocation systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 36, no. 6, pp. 1234–1243, Nov. 2006.
- [18] Z. Ma, Z. Li, and A. Giua, "Characterization of admissible marking sets in Petri nets with conflicts and synchronizations," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1329–1341, Mar. 2017.
- [19] Z. Ma, Y. Tong, Z. Li, and A. Giua, "Basis marking representation of Petri net reachability spaces and its application to the reachability problem," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1078–1093, Mar. 2017.
- [20] S. G. Wang, D. You, and C. Seatzu, "A novel approach for constraint transformation in Petri nets with uncontrollable transitions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 8, pp. 1–8, Feb. 2017.
- [21] Y. Li, L. Yin, Y. Chen, Z. Yu, and N. Wu, "Optimal Petri net supervisor synthesis for forbidden state problems using marking mask," *Inf. Sci.*, vol. 505, pp. 183–197, Dec. 2019.
- [22] Z. Li, G. Liu, H.-M. Hanisch, and M. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 1, pp. 178–191, Jan. 2012.
- [23] Y. Wen, C. Li, and M. Der Jeng, "A polynomial algorithm for checking diagnosability of Petri nets," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2005, pp. 2542–2547.
- [24] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 137–141, Feb. 2003.
- [25] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions," *Int. J. Adv. Manuf. Technol.*, vol. 19, no. 3, pp. 192–208, Feb. 2002.
- [26] Y. Tong, Z. Li, C. Seatzu, and A. Giua, "Verification of state-based opacity using Petri nets," *IEEE Trans. Autom. Control*, vol. 62, no. 6, pp. 2823–2837, Jun. 2017.
- [27] 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., Comput. Cybern. Simulation*, Oct. 2015, pp. 3750–3755.
- [28] G. Liu, P. Li, Z. Li, and N. Wu, "Robust deadlock control for automated manufacturing systems with unreliable resources based on Petri net reachability graphs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 7, pp. 1371–1385, Jul. 2019.
- [29] L. Piroddi, R. Cordone, and I. Fumagalli, "Selective siphon control for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 6, pp. 1337–1348, Nov. 2008.
- [30] L. Piroddi, R. Cordone, and I. Fumagalli, "Combined siphon and marking generation for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 3, pp. 650–661, May 2009.
- [31] J. Park and S. A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Trans. Autom. Control*, vol. 46, no. 10, pp. 1572–1583, Oct. 2001.
- [32] D. Y. Chao, "Improved controllability test for dependent siphons in  $S^3PR$  based on elementary siphons," *Asian J. Control*, vol. 12, no. 3, pp. 377–391, May 2010.
- [33] Y. Huang, M. Jeng, X. Xie, and S. Chung, "Deadlock prevention policy based on Petri nets and siphons," *Int. J. Prod. Res.*, vol. 39, no. 2, pp. 283–305, Jan. 2001.
- [34] D. You, S. Wang, W. Dai, W. Wu, and Y. Jia, "An approach for enumerating minimal siphons in a subclass of Petri nets," *IEEE Access*, vol. 6, pp. 4255–4265, 2018.
- [35] Z. Li, M. Zhou, and M. Jeng, "A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 182–188, Jan. 2008.
- [36] M. Uzam and M. Zhou, "Iterative synthesis of Petri net based deadlock prevention policy for flexible manufacturing systems," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2004, pp. 4260–4265.
- [37] M. Uzam and M. C. Zhou, "An iterative synthesis approach to Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 37, no. 3, pp. 362–371, Apr. 2007.
- [38] Y. Chen and Z. Li, "Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 47, no. 5, pp. 1028–1034, May 2011.
- [39] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpeleta, "A Petri net structure-based deadlock prevention solution for sequential resource allocation systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 271–277.
- [40] Y. Wen and M. Jeng, "Diagnosability analysis based on T-invariants of Petri nets," in *Proc. IEEE Netw., Sens. Control*, Mar. 2005, pp. 371–376.
- [41] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995.
- [42] F. Chu and X.-L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. Robot. Autom.*, vol. 13, no. 6, pp. 793–804, Dec. 1997.
- [43] Z. Li and M. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Ind. Informat.*, vol. 2, no. 4, pp. 313–325, Nov. 2006.
- [44] Z. Li and M. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 1, pp. 38–51, Jan. 2004.
- [45] K. Xing, F. Tian, and X. Yang, "Optimal deadlock avoidance Petri net supervisors for automated manufacturing systems," *J. Control Theory Appl.*, vol. 5, no. 2, pp. 152–158, May 2007.
- [46] C. Gu, Z. Li, N. Wu, M. Khalgui, T. Qu, and A. Al-Ahmari, "Improved multi-step look-ahead control policies for automated manufacturing systems," *IEEE Access*, vol. 6, pp. 68824–68838, 2018.
- [47] C. Gu, Z. Li, and A. Al-Ahmari, "A multistep look-ahead deadlock avoidance policy for automated manufacturing systems," *Discrete Dyn. Nature Soc.*, vol. 2017, pp. 1–16, Aug. 2017.
- [48] S. H. Yang, "Structural analysis of Petri nets based on linear program," M.S. thesis, School Mechano-Electron. Eng., Xidian Univ., Xi'an, China, 2019.
- [49] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. New York, NY, USA: Springer, 2008.
- [50] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
- [51] K. Xing, M. Chu Zhou, H. Liu, and F. Tian, "Optimal Petri-net-based polynomial-complexity deadlock-avoidance policies for automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 1, pp. 188–199, Jan. 2009.
- [52] X. Li, Z. Yu, Z. Li, and N. Wu, "Group consensus via pinning control for a class of heterogeneous multi-agent systems with input constraints," *Inf. Sci.*, vol. 542, pp. 247–262, Jan. 2021, doi: [10.1016/j.ins.2020.05.085](https://doi.org/10.1016/j.ins.2020.05.085).
- [53] Z. Yu, L. Zhou, Z. Ma, and M. A. El-Meligy, "Trustworthiness modeling and analysis of cyber-physical manufacturing systems," *IEEE Access*, vol. 5, pp. 26076–26085, 2017.
- [54] S. Zhou, Z. Yu, E. S. A. Nasr, H. A. Mahmoud, E. M. Awwad, and N. Wu, "Homomorphic encryption of supervisory control systems using automata," *IEEE Access*, vol. 8, pp. 147185–147198, 2020, doi: [10.1109/ACCESS.2020.3014217](https://doi.org/10.1109/ACCESS.2020.3014217).

• • •