Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 4847072, 12 pages
https://doi.org/10.1155/2019/4847072

Research Article

One Novel and Optimal Deadlock Recovery Policy for Flexible Manufacturing Systems Using Iterative Control Transitions Strategy

Ter-Chan Row,[1] Wei-Ming Syu,[2] Yen-Liang Pan ,[3,4,5] and Ching-Cheng Wang[2]

[1]*Department of Electronic Engineering, Army Academy, Taoyuan, Taiwan*
[2]*Institute of Manufacturing Information and System, National Cheng Kung University, Tainan, Taiwan*
[3]*Mathematics and Physics Division, General Education Center, Air Force Academy, Kaohsiung, Taiwan*
[4]*Artificial Intelligence and Robotics Design Lab, Air Force Academy, Kaohsiung, Taiwan*
[5]*Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan*

Correspondence should be addressed to Yen-Liang Pan; peterpan960326@gmail.com

Received 13 November 2018; Revised 30 January 2019; Accepted 17 February 2019; Published 27 March 2019

Guest Editor: Sutasn Thipprakmas

This paper focuses on solving deadlock problems of flexible manufacturing systems (FMS) based on Petri nets theory. Precisely, one novel control transition technology is developed to solve FMS deadlock problem. This new proposed technology can not only identify the maximal saturated tokens of idle places in Petri net model (PNM) but also further reserve all original reachable markings whatever they are legal or illegal ones. In other words, once the saturated number of tokens in idle places is identified, the maximal markings of system reachability graph can then be checked. Two classical S^3PR (the Systems of Simple Sequential Processes with Resources) examples are used to illustrate the proposed technology. Experimental results indicate that the proposed algorithm of control transition technology seems to be the best one among all existing algorithms.

1. Introduction

Nowadays, in a rapidly changing society, every factory tries to use robots to possess complicated manufacturing workings. Flexible manufacturing systems (FMS) [1] are therefore designed to process these workings. However, for FMS there could exist an undesired phenomenon while robots are scramble for the shared resources of one FMS. The kind of competition is called the deadlock state. Besides, in order to solve above deadlock problem, three technologies are proposed [2]. The first one is deadlock avoidance. It can immediately avoid a deadlock and do the next step. The next one is called deadlock detection. It can find all deadlocks and recover the blocks. Please note that the above two technologies are adopted in dynamic method. They could have the time-consuming problem. The last one belongs to static. It considers the whole deadlocks in a model and solves them in advance. This technology is called deadlock prevention. Papers [3–11] presented a deadlock prevention to maintain the optimal permissiveness.

In deadlock prevention domain, Barkaoui and Abdallah firstly propose a deadlock prevention concept and method for a class of FMS called System of Simple Sequential Processes with Resources (S^3PR) [12]. In the following, Ezpeleta *et al.* [1] also presented a particular Petri net model for this kind of FMS. They used siphons concept and tried to add control places and related arcs to ensure the liveness of the FMS model. However, permissive is not large enough in deadlock free situation.

In the following, Huang *et al.* [3] presented a method called maximal siphons control technology to solve the deadlock problem. The proposed algorithm can extract the strict minimal siphon from maximal siphons such that all siphons are successfully avoided. The method seems to be better than [1] since its permissive markings are more than [1]. Besides, Huang *et al.* [4] proposed another algorithm named iterative approach with mixed integer programming

(MIP) method to improve conventional maximal siphons so as to enhance the computational efficiency. However, the two kinds of siphon based methods are still not maximally permissive.

On the other hand, Li and Zhou [5] proposed the elementary siphon concept to minimize the redundant siphons. The dependent siphons can then be controlled simultaneously once elementary siphons are controlled. Besides, Li and Zhou [6] clarified the elementary siphons and then simplified the number of control places. The permissiveness is also ameliorative. Liu *et al.* [7] introduced and proved a live Petri net controller can be established by adding a control place and related arcs to each strict minimal siphon in a controllable siphon basis. They presented a new novel deadlock prevention policy based on the concept of a controllable siphon basis. It is pity; the above deadlock prevention policies are still not maximally permissive. For all siphon control methods, refer to [14, 15].

Besides, Uzam [8] and Uzam and Zhou [9] follow the theory of region to obtain the optimal permissiveness. They classified all reachable states into two zones, deadlock-zone and deadlock-free-zone. The deadlock-zone was composed of dead markings and crucial dead markings. Finally, using the algorithm was successful to avoid the crucial dead markings and to make the net live. However, redundant the event-state-separation-problems (ESSPs) cannot be entirely avoided for large FMS cases. Huang and Pan [16, 17] and Huang *et al.* [18] further develop a computationally more efficient optimal deadlock control policy to improve the computational efficiency of the conventional theory of region.

Piroddi *et al.* [10] also proposed selective siphon control method for solving deadlock problems in deadlock prevention domain. The policy merged selective siphons and crucial markings. By every iteration, the redundant siphons will be controlled simultaneously once selective siphons are controlled. Its most advantage is maximally permissive markings which can still be held. However, its drawback is that reachability graph is needed to run in every iteration. Therefore, Pan *et al.* [19] develop one computationally improved methodology to enhance its computational efficiency. For improving above shortcoming, Pan *et al.* [11] further try to merge theory of regions and the selective siphons control method to achieve optimal control goal. This proposed method can identify all crucial markings in first reachability graph and use the theory of regions to check these crucial markings. Finally, selective siphons method is used to control all minimal siphons. Maximally permissive markings are surely still held. Preciously, reachability graph is just one time needed to run. At the same time, Pan *et al.* [20] also use the selective siphons technology with the method in [18] to further enhance the computational time of traditional theory of regions.

In siphon control domain, Chao and Pan [21] also proposed the uniform formulas for compound siphons, complementary siphons, and characteristic vectors to solve the deadlock problems of flexible manufacturing systems easily. This can simplify the computational efficiency.

On the other hand, Huang *et al.* [13] firstly presented transition-based deadlock prevention policy to solve the deadlocks problem of FMS. However, there are three main drawbacks in [13]. Firstly, they cannot identify the exact number of tokens in idle places in advance. Besides, they need to check all reachable markings of FMS reachability graph to avoid getting live lock. Finally, their some controllers seem to be redundant. Therefore, in our previous work [22], we first propose all reachability graph and First Dead Marking (FDM) viewpoints based on control transition to improve the computational efficiency of [13]. The improved work [22] does overcome the disadvantage of [13]. It is a pity; its example is a small PNM.

Zhang and Uzam [23] also adopted control transition method to solve the deadlocks of flexible manufacturing systems. They improved the computational efficiency of [13] since the redundant controller's problem was overcome. However, they still cannot identify the correct saturated number of tokens of idle places in the process of seeking the maximal number of reachable markings.

Recently, Chen *et al.* [24] also propose one deadlock policy to solve FMS deadlock problem using control transition concept. They define a minimal recovery transitions using iterative approach by an integer linear programming problem (ILPP) and objective functions. However, it is not an efficient method since ILPP is used to obtain proper supervisors.

Based on above survey, one can know that how to solve deadlock problems of flexible manufacturing systems is a very difficult issue. Many experts make their most effort in the issue [1, 3, 5, 9, 10, 25–34]. Most important, how to keep systems maximally permissive is needed to concern.

To focus the maximally live performance, this paper proposes one new, different, and simple control method based on control transition technology. It presents a very different and novel concept in solving deadlock problems. First of all, the number of saturated tokens in idle (source or sink) places must be identified. Then, the maximal markings of its reachability graph can then be checked. Further, control transitions can be obtained. Finally, the maximally permissive live markings can be present once the controllers are added it into PNM.

The rest of this paper is as follows. Section 2 presents the preliminaries about the Petri nets. Section 3 proposes a new deadlock prevention approach and proves it can be executed. Section 4 presents two examples to verify by our prevention policy. The last, Section 5 is the conclusion.

2. Preliminaries

A Petri net model (PNM) is formed by five elements. The model will become actively according to the tokens firing. A PNM will be defined as follows.

Definition 1 (see [1, 2, 35]). $N = (P, T, F, W, M_0)$, where

P is a finite, nonempty, and disjoint set of places which shows $P = \{p_1, p_2, p_3, \ldots, p_m\}$.$T$ is also a finite, nonempty, and disjoint set of transitions which shows $T = \{t_1, t_2, t_3, \ldots, t_m\}$. The third element F is called the flow relation or the set of directed arcs with the arrows from places to transitions or from transitions to places. The characteristic of F is $F \subseteq (P \times T) \cup (T \times P)$. $W(\cdot)$ is assigned the weight to

an arc. Furthermore, the set of arcs in PNM are nonnegative integer. The net is called ordinary when the weight of arcs in PNM is all one. M_0 is named the initial marking combined with net system (N, M_0).

In a net system (N, M_0) and a node $x \in P \cup T$, then the preset of node x is defined as $\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$. The postset of node x is defined as $x\bullet = \{y \in P \cup T \mid (x, y) \in F\}$.

A transition $t \in T$ is enabled at marking M by iff $\forall p \in \bullet t$ and $M(p) \geq W(p, t)$ that will be denoted as $M[t\rangle$. Afterward enabled and firing a transition, M will get a new marking M'. It indicates $M[t\rangle M'$. A marking M reaches to M' shown by $M[\sigma\rangle M'$ if there exists a firing sequence $\sigma = t_1 t_2 t_3 \ldots t_n$. The set of marking reachable from M in N is called a reachability set as $R(N, M)$ in net system (N, M_0). The net system (N, M_0) is live iff $\forall t \in T$ is enabled and dead iff $\exists t \in T$ is disabled.

Definition 2 (see [35, 36]). Given ${}^\bullet S \subset P$, (resp., $S^\bullet$) denotes the set of transitions with at least one output (resp., input) place belonging to S. Formally, ${}^\bullet S = \bigcup_{p \in s} {}^\bullet p$ and $S^\bullet = \bigcup_{p \in s} p^\bullet$. S is called a *siphon* if ${}^\bullet S \subseteq S^\bullet$; i.e., any input transition of S is also an output transition of S. It is called a *trap* if $S^\bullet \subseteq {}^\bullet S$. A siphon (resp. trap) is said to be *minimal* if it does not contain other siphons (resp., traps).

Definition 3 (see [1, 37–39]). A S^3PR is the union of a set from the two combined simple sequential processes with resources (S^2PR) nets $N_i = (P_i \cup \{p_i^0\} \cup P_{Ri}, T_i, F_i)$ by sharing common places, and the following statements are true.

(1) P_i and P_{Ri} are called operation and resource places of N_i, respectively. $\boldsymbol{p}_i^0$ is called the process idle places.

(2) $P_i \neq \phi$; $P_{Ri} \neq \phi$; $p_i^0 \notin P_i$; $(P_i \cup \{p_i^0\}) \cap P_{Ri} = \phi$; $\forall p \in P_i$, $\forall t \in {}^\bullet p$, $\forall t' \in p^\bullet$, $\exists r_p \in P_{Ri}$, ${}^\bullet t \cap P_{Ri} = t'^\bullet \cap P_{Ri} = \{r_p\}$; $\forall r \in P_{Ri}$, ${}^{\bullet\bullet} r \cap P_i = r^{\bullet\bullet} \cap P_i \neq \phi$; $\forall r \in P_{Ri}$, ${}^\bullet r \cap r^\bullet = \phi$; ${}^{\bullet\bullet}(p_i^0) \cap P_{Ri} = (p_i^0)^{\bullet\bullet} \cap P_{Ri} = \phi$.

(3) $N_i{}'$ is a strongly connected state machine, where $N_i' = (P_i \cup \{p_i^0\}, T_i, F_i)$ is the resulting net after the places in P_{Ri} and related arcs are removed from N_i.

(4) Every circuit of $N_i{}'$ contains place p_i^0.

(5) Any two N_i and N_j are composable if they share a set of common resource places. Every shared place must be a resource.

(6) The transitions of $(p_i^0)^\bullet$ and ${}^\bullet(p_i^0)$ are called source and sink transitions of S^3PR, respectively.

(7) For $r \in P_R$, $H(r) = ({}^{\bullet\bullet} r) \cap P_A$ is the set of operation places that use r and are called holders of r.

(8) For $p \in P_A$, $({}^{\bullet\bullet} p) \cap P_R = \{r_p\}$ where resource place r_p is called the resource used by p.

3. One Novel Deadlock Prevention Policy

Since this paper tries to propose one novel deadlock recovery policy by adopting control transition method, in this section, the authors will present the important concept of the proposed policy and how it works. First of all, the number of saturated tokens in idle (e.g., source or sink) places should be identified. Secondly, the maximal number of markings in its reachability graph can then be checked and calculated.

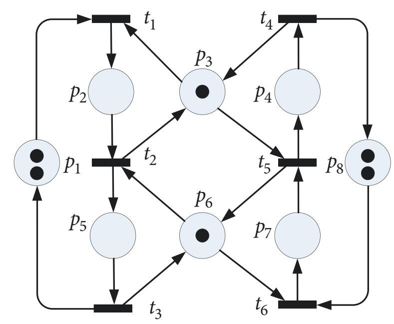

Figure 1: Petri net model of two idle places.

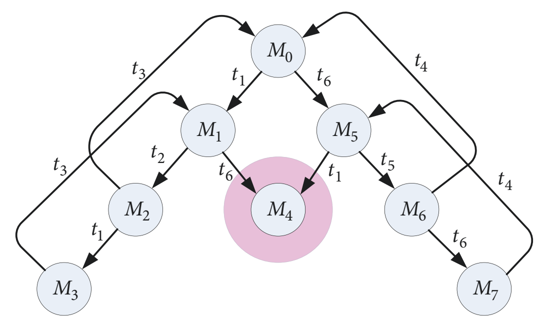

Figure 2: The reachability states of above simple PNM.

Further, control transitions can hence be obtained. Finally, the maximally permissive live markings can be present once the transition-based controllers are added to the deadlock prone PNM. In the following, the authors will show the detailed information as follows.

3.1. Saturated Number of Tokens in Idle Places. In this subsection, we will show how to identify the saturated number of tokens in PNM's idle places. The first step is that one needs to pick all idle places and try to change the number of their tokens. Please note that the number is from small to large. Then, all reachable states will hence change due to the different number of tokens in idle places. Finally, the number of reachable states will reach one maximal number whatever the number of tokens increased in idle places. Under the situation, we call it the optimal number or saturated number of token in idle places. For example (please refer to Figure 1), Table 1 shows the different number of tokens in two the idle places (i.e., p_1 and p_8). From the Table 1 we can see that the PNM reaches the maximal reachable states when (p_1, p_8) is $(2, 2)$. The experimental groups 15~16, 18~20, and 22~24 are just used to present that the maximal reachable states of the PNM are equal to 8. Further, the maximal reachable states are 8 whatever how large number of tokens is added or increased into the idle place. The set of $(2, 2)$ is what we want, it is the saturated number (or called optimal) of tokens in idle places of this PNM.

In the following, we demonstrate our algorithm by using above example. Please review the first example again; from Figure 2, we can easily see that there exists one deadlock in its reachability graph. All information of reachable markings (states) is shown in Table 2 so that all readers can easily check them.

Table 1: The variety of reachability states by different tokens in idle places of Figure 1.

Experimental Group	(p_1, p_8)	Reachable States	Experimental Group	(p_1, p_8)	Reachable States
1	(1, 0)	3	13	(2, 1)	7
2	(2, 0)	4	**14**	**(2, 2)**	**8**
3	(3, 0)	4	15	(2, 3)	8
4	(4, 0)	4	16	(2, 4)	8
5	(0, 1)	3	17	(3, 1)	7
6	(0, 2)	4	18	(3, 2)	8
7	(0, 3)	4	19	(3, 3)	8
8	(0, 4)	4	20	(3, 4)	8
9	**(1, 1)**	**6**	21	(4, 1)	7
10	(1, 2)	7	22	(4, 2)	8
11	(1, 3)	7	23	(4, 3)	8
12	(1, 4)	7	24	(4, 4)	8

Table 2: The detailed information of all reachable markings in Figure 2.

Marking No.	Classification	Information of Markings $[p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8]^T$
M_0	*Initial Marking*	$[2, 0, 1, 0, 0, 1, 0, 2]^T$
M_1	*Live marking*	$[1, 1, 0, 0, 0, 1, 0, 2]^T$
M_2	*Live Marking*	$[1, 0, 1, 0, 1, 0, 0, 2]^T$
M_3	*Live Marking*	$[0, 1, 0, 0, 1, 0, 0, 2]^T$
M_4	*Dead Marking*	$[1, 1, 0, 0, 0, 0, 1, 1]^T$
M_5	*Live Marking*	$[2, 0, 1, 0, 0, 0, 1, 1]^T$
M_6	*Live Marking*	$[2, 0, 0, 1, 0, 1, 0, 1]^T$
M_7	*Live Marking*	$[2, 0, 0, 1, 0, 0, 1, 0]^T$

3.2. Identify the Control Transition. According to the definitions as follows, control transitions can be calculated and obtained in one PNM.

Definition 4. A Petri net N is with an initial marked M_0; the set of all reachable markings for the Petri net N is denoted by $R(N, M_0)$.

Definition 5 (see [40]: reversibility). (1) A Petri net N has a home state M_h for an initial state M_0 if for every reachable state $M \in R(N, M_0)$, a firing sequence σ_i exists such that $M[\sigma_i\rangle M_h$.

(2) A Petri net N is reversible for an initial state M_0 if M_0 is a home state.

Definition 6 (dead marking). M_d is a dead marking and is defined as follows:

$$M_d = \{M \mid \forall M \in R(N, M_0), \nexists t \in T, \ni M[t\rangle\} \tag{1}$$

Definition 7 (reachability tree). Reachability tree is called reachability graph (RG) or reachability states (RS). A Petri net N with an initial marking M_0 is called a Petri net system (N, M_0). The reachability graph of (N, M_0) is defined as follows:

$$\forall t \in T, \exists t, \text{s.t. } M_i[t\rangle M_j \Longrightarrow \cup \left(M_i[t\rangle M_j\right) \\ \text{where } M_i, M_j, \text{and } t \in R(N, M_0) \tag{2}$$

The reachability graph is represented as $R(N, M_0)$.

Definition 8 (a transition t is enabled and firing). A transition t is enabled $\forall p \in P \wedge p \in {}^{\bullet}t \wedge M(p) \geq W(p, t)$. One can show that, as $M[t\rangle$, the next step is firing. A transition t is firing if a new marking M' yields. One can denote $M[t\rangle M'$.

Definition 9 (control transition). A Petri net with a dead marking M_d, $\forall M_d \in R(N, M_0)$, $M \in R(N, M_0)$, $M_d \subset M$, $\exists t_c \notin T$, s.t. $M_d[t_c\rangle M$. t_c is called a control transition.

Definition 10 (saturated number of tokens in idle places (SNTIP)). $\exists \min(n)$ in a P_{id} such that the RG is maximal.

$\sum P_{id}(\min(n))$ are obtained the maximal RG. Then, $\sum P_{id}(\min(n))$ are called SNTIP.

n is a token number in an idle place. $n = 0, 1, 2, 3, \ldots$.

P_{id} is an idle place.

Theorem 11. *A deadlocked Petri net (S^3PR), $\forall M_d \in R(N, M_0)$, $\exists t_c \notin T$, such that M_d can reach to M_0 and M_d, becomes live.*

Proof. It is well known that the state equation is $M_k = M_{k-1} + A^T u_k$, $k = 1, 2, 3, \ldots$ [41]. Here, we change the state equation to $M_{k-1} = M_k - A^T u_k$ then $M_{k-1} = M_k + A^T(-u_k)$. Let $-u_k = y_k$. Obtaining the new state equation is $M_{k-1} = M_k + A^T y_k$ $k = 1, 2, 3, \ldots$. One expands and pluses by every series. The result is $M_0 = M_k + A^T \Sigma y_k$. Assume $M_k = M_d$ and $x = \Sigma y_k$, then $M_0 - M_d = A^T x$, which can be written $M_0 - M_d = \Delta M$. This means $A^T x = \Delta M$. When $\Delta M = 0$, then $A^T x = 0$ is a homogenous equation. $\exists x$ such that $M_0 =$

TABLE 3: The flow step of our novel deadlock recovery policy.

Maximally Permissive Deadlock Recovery Policy
(i) Input: one deadlock S^3PR PNM.
(1) Identify the saturated number of tokens in idle places.
(2) Identify the crucial dead markings from reachability graph.
(3) Calculate all control transitions.
(4) Add all control transitions into original PNM.
(ii) Output: one live S^3PR PNM.

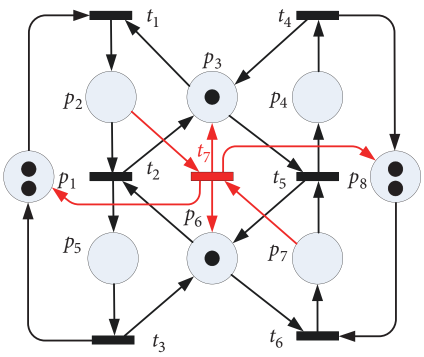

FIGURE 3: The Petri net model with control transition.

$M_d + A^T x$. Based on [42], ones know $A^T x = O(t,p) - I(t,p)$. It means $\exists t_c$ ($t_c = O(t,p) - I(t,p)$) such that M_d can reach M_0. We can infer that M_0 is reachable from M_d.

To obtain the equation of $M_0 = M_k + A^T \Sigma y_k$, one has the following:

$$M_0 = M_1 + A^T y_1 \quad (3)$$

$$M_1 = M_2 + A^T y_2 \quad (4)$$

$$M_2 = M_3 + A^T y_3 \quad (5)$$

$$\vdots \quad (6)$$

$$M_{k-2} = M_{k-1} + A^T y_{k-1} \quad (k\text{-}2)$$

$$M_{k-1} = M_k + A^T y_k \quad (k\text{-}1)$$

Equations (3) + (4) + (5)...(k-2) + (k-1) are equal to $M_0 = M_k + A^T \Sigma y_{k\Diamond}$. □

The example of Figure 1 is present to obey Theorem 11 and obtain a control transition in the next section.

3.3. The Flow Steps of Our Deadlock Prevention Policy. The deadlock recovery policy and its steps are shown in Table 3.

In the following, we want to calculate the optimal control transitions so that they can be added to one deadlock PNM and live it. Please notice that the controller is calculated by initial marking ($M_0 = [2,0,1,0,0,1,0,2]^T$) and the dead marking ($M_4 = [1,1,0,0,0,0,1,1]^T$) based on Theorem 11. The firing count vector $x = [-2,-1,-1,0,0,-1]^T$ and the transpose of incidence matrix A^T are as follows:

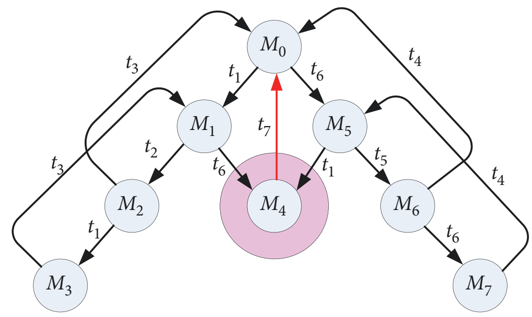

FIGURE 4: The reachability states of Figure 3.

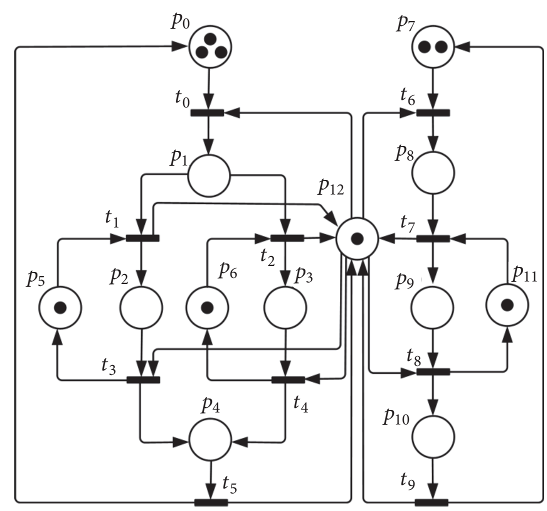

FIGURE 5: One simple Petri net model [13].

$$A^T = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \quad (7)$$

The equation of $M_0 = M_d + A^T x$ is shown as follows:

$$\begin{bmatrix} 2 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} -2 \\ -1 \\ -1 \\ 0 \\ 0 \\ -1 \end{bmatrix} \quad (8)$$

Therefore, according to our proposed algorithm, the all output places of the control transition are $O(t_7) = p_1 + p_3 + p_6 + p_8$, and the all input places of the control transition are $I(t_7) = p_2 + p_7$. The detailed information of the controller is shown in Figure 3. The PNM is deadlock free when we added the controller to initial deadlock PNM. The detailed information is shown in Figure 4.

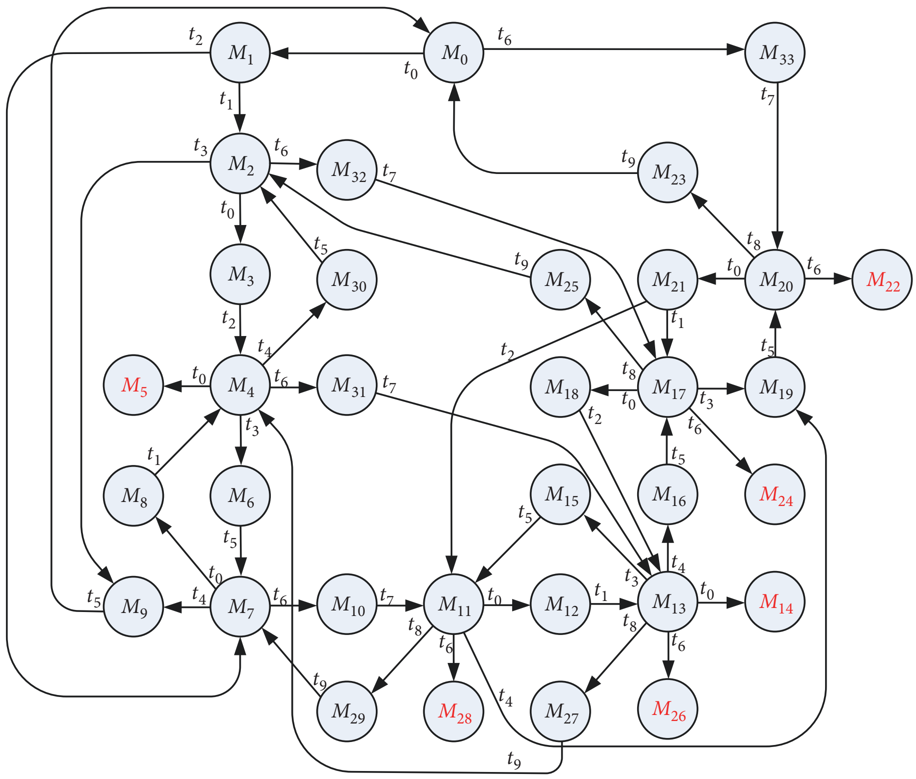

FIGURE 6: The reachability graph of Figure 5.

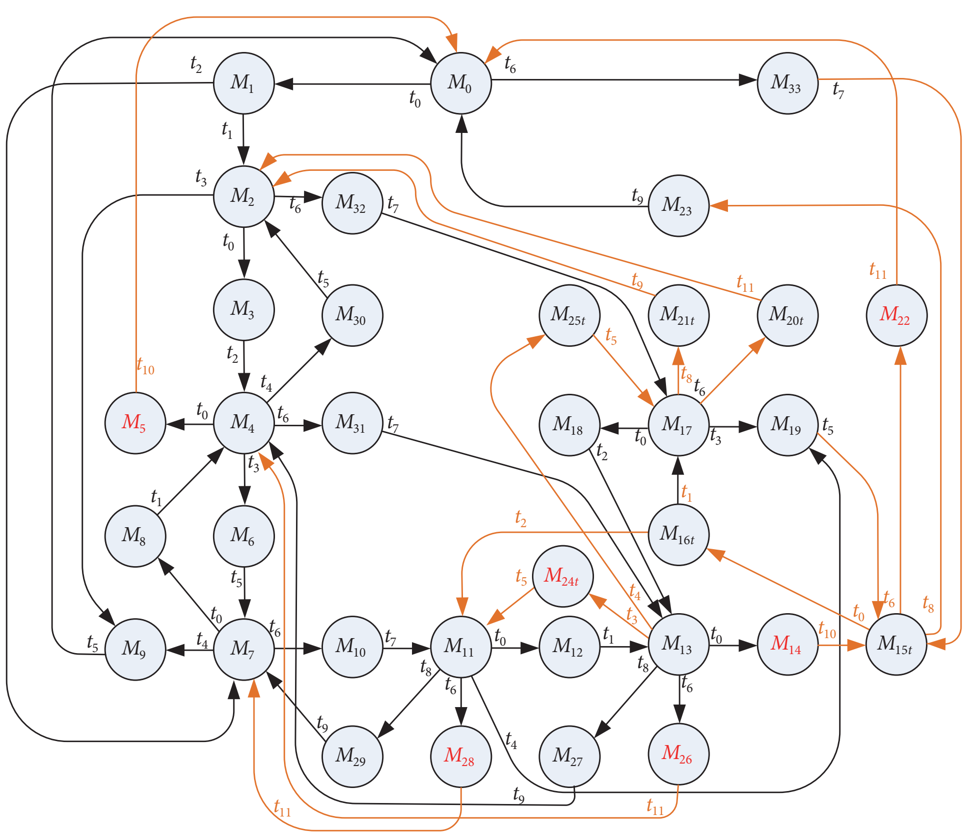

FIGURE 7: Its deadlock free reachability graph of Figure 5.

Table 4: The variety of reachability states by different tokens in idle places of Figure 5.

Set No.	(p_0, p_7)	Reachable States	Set No.	(p_0, p_7)	Reachable States
1	(1, 0)	5	10	(1, 2)	19
2	(2, 0)	10	11	(1, 3)	19
3	**(3, 0)**	**11**	12	(2, 1)	28
4	(4, 0)	11	13	(2, 2)	32
5	(0, 1)	4	14	(2, 3)	32
6	**(0, 2)**	**5**	15	(3, 1)	30
7	(0, 3)	5	**16**	**(3, 2)**	**34**
8	(0, 4)	5	17	(3, 3)	34
9	(1, 1)	16	18	(4, 3)	34

Table 5: Two crucial control transitions.

No.	Property of Crucial Dead Markings	$I(t_{ci})$	$O(t_{ci})$	Identified in Which Idle Places
1	$p_1 + p_2 + p_3 + p_{11}$	$I(t_{c1}) = p_1 + p_2 + p_3$	$O(t_{c1}) = 3p_0 + p_5 + p_6 + p_{12}$	(3, 0)
2	$p_5 + p_6 + p_8 + p_9$	$I(t_{c2}) = p_8 + p_9$	$O(t_{c2}) = 2p_7 + p_{11} + p_{12}$	(0, 2)

4. Examples and Comparisons

In this section, we want to evaluate our proposed deadlock prevention and make some comparisons with existing literature. Two classical examples are used to illustrate the proposed algorithm. The first example is one simple PNM. For easy tracing, we present the detailed information step by step. Please note that there are at least two dead markings in the two PNMs. In other words, some deadlock markings are redundant. Therefore, we not only want to identify the optimal number of tokens in its idle places, but also identify its crucial dead markings so that we can use them to calculate final controllers. For easy checking, the two PNM used TINA to run their reachability graph.

4.1. The Simple Petri Net Model [13]

Step 1 (identify the optimal number of tokens in idle places). In this step, we firstly used one famous software TINA [43] to check the optimal number of tokens in idle places. Table 4 shows the variety of reachable states by changing different number of tokens in idle places of Figure 5. Please notice that the maximal number of tokens in p_0 is equal to 3 when the number of tokens in p_7 is set to zero. Similarly, the maximal number of tokens in p_7 is 2 when the number of tokens in p_0 is set to zero. The third group and the sixth group in Table 4 show the optimal number of tokens in its idle places, respectively. It is obvious that in sixteenth group when (p_0, p_7) is equal to $(3, 2)$ the idle places have the optimal number of tokens. $(p_0, p_7) = (3, 2)$ is the saturated number of tokens because the reachable states of this PNM are in its maximal value even one gives more tokens in (p_0, p_7). The maximal number of reachable markings is checked once the optimal number of tokens in one's idle places is identified. The seventeenth and the eighteenth groups in Table 4 show the above description.

Step 2 (identify the crucial dead markings from reachability graph). In this PNM, there are 34 reachable markings in total including 6 dead markings when the idle places are in saturation (please refer to Figure 6). Based on TINA, the detailed information of the 6 dead markings is $M_5 = (p_1, p_2, p_3, 2p_7, p_{11})$, $M_{14} = (p_1, p_2, p_3, p_7, p_9)$, $M_{22} = (3p_0, p_5, p_6, p_8, p_9)$, $M_{24} = (2p_0, p_2, p_6, p_8, p_9)$, $M_{26} = (p_0, p_2, p_3, p_8, p_9)$, and $M_{28} = (2p_0, p_3, p_5, p_8, p_9)$, respectively. However, the 6 dead markings could be dependent. In other words, controllers could be redundant although the PNM also can be deadlock free when all controllers are added to above PNM. Therefore, one needs to identify crucial dead markings so as to obtain all independent controllers. By using the proposed algorithm, we can obtain 2 crucial dead markings, $(p_1+p_2+p_3+p_{11})$ and $(p_5+p_6+p_8+p_9)$, respectively. The detailed information is shown in Table 5.

Step 3 (calculate all control transitions). According to the state equation (i.e., $M_0 - M_d = O(t, p) - I(t, p)$), two sets of control transitions can then be obtained. The detailed information of them is (1) $I(t_{c1}) = p_1 + p_2 + p_3$, $O(t_{c1}) = 3p_0 + p_5 + p_6 + p_{12}$ and (2) $I(t_{c2}) = p_8 + p_9$, $O(t_{c2}) = 2p_7 + p_{11} + p_{12}$ also shown in Table 5.

Step 4 (add all control transitions into original PNM). The PNM is deadlock free when the two sets of control transitions are put into the original PNM. Besides, the number of its final reachable live markings is still 34. The proposed deadlock prevention policy not only solves the deadlock problem of the PNM of Figure 5 but also holds all reachable markings even they are dead markings in the beginning. Its deadlock free reachability graph is shown in Figure 7. Please notice that the contents of six markings M_{15t}, M_{16t}, M_{20t}, M_{21t}, M_{24t}, and M_{25t} are different from the initial markings.

4.2. Another Classical Complex PNM. In the following, we want to show that the proposed deadlock prevention policy can be used to solve a complex PNM shown in Figure 8 [1]. In this PNM, it is composed of three robots named R_1, R_2, and R_3, denoted by p_{20}, p_{21}, and p_{22}, four machines named M_1, M_2, M_3, and M_4 denoted by p_{23}, p_{24}, p_{25} and p_{26}, respectively. Finally, three idle places (I_1/O_1, I_2/O_2, and I_3/O_3) are given named p_1, p_{14}, and p_5, respectively. The others in this model are operation places. Further, there are

Table 6: The 20 sets of crucial dead markings of Figure 8.

No.	Idle Places (p_1, p_5, p_{14})	Crucial Dead Markings
1	(3, 0, 0)	$p_2 + 2p_3 + p_{20} + p_{22} + 2p_{23} + 2p_{25} + 2p_{26}$
2	(1, 1, 2)	$p_1 + p_{12} + 2p_{18} + p_{20} + p_{22} + 2p_{23} + 2p_{24} + 2p_{25}$
3	(1, 2, 1)	$p_1 + 2p_{11} + p_{17} + p_{20} + p_{22} + 2p_{23} + 2p_{24} + 2p_{26}$
4	(1, 2, 2)	$p_1 + p_{12} + p_{13} + p_{18} + p_{19} + p_{20} + 2p_{23} + 2p_{24} + 2p_{25}$
5	(1, 2, 3)	$p_2 + 2p_9 + 2p_{18} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
6	(1, 2, 3)	$p_3 + p_8 + p_9 + 2p_{18} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
7	(1, 3, 1)	$p_1 + p_{12} + 2p_{13} + p_{19} + p_{20} + 2p_{23} + 2p_{24} + 2p_{25}$
8	(1, 3, 2)	$p_2 + 2p_9 + p_{13} + p_{18} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
9	(1, 3, 2)	$p_3 + p_8 + p_9 + p_{13} + p_{18} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
10	(1, 3, 3)	$p_1 + p_6 + 2p_7 + 2p_{16} + p_{17} + p_{22} + 2p_{24} + 2p_{26}$
11	(1, 3, 3)	$p_1 + p_8 + 2p_9 + 2p_{18} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
12	(1, 4, 1)	$p_3 + p_8 + p_9 + 2p_{13} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
13	(1, 4, 1)	$p_2 + 2p_9 + 2p_{13} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
14	(1, 4, 2)	$p_1 + p_6 + 2p_7 + p_{11} + p_{16} + p_{17} + p_{22} + 2p_{24} + 2p_{26}$
15	(1, 4, 2)	$p_1 + p_8 + 2p_9 + p_{13} + p_{18} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
16	(1, 5, 1)	$p_1 + p_8 + 2p_9 + 2p_{13} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
17	(2, 1, 0)	$2p_3 + p_8 + p_{20} + p_{22} + 2p_{23} + 2p_{25} + 2p_{26}$
18	(2, 1, 3)	$p_2 + p_3 + p_9 + 2p_{18} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
19	(2, 2, 2)	$p_2 + p_3 + p_9 + p_{13} + p_{18} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$
20	(2, 3, 1)	$p_2 + p_3 + p_9 + 2p_{13} + p_{19} + p_{20} + 2p_{23} + 2p_{25}$

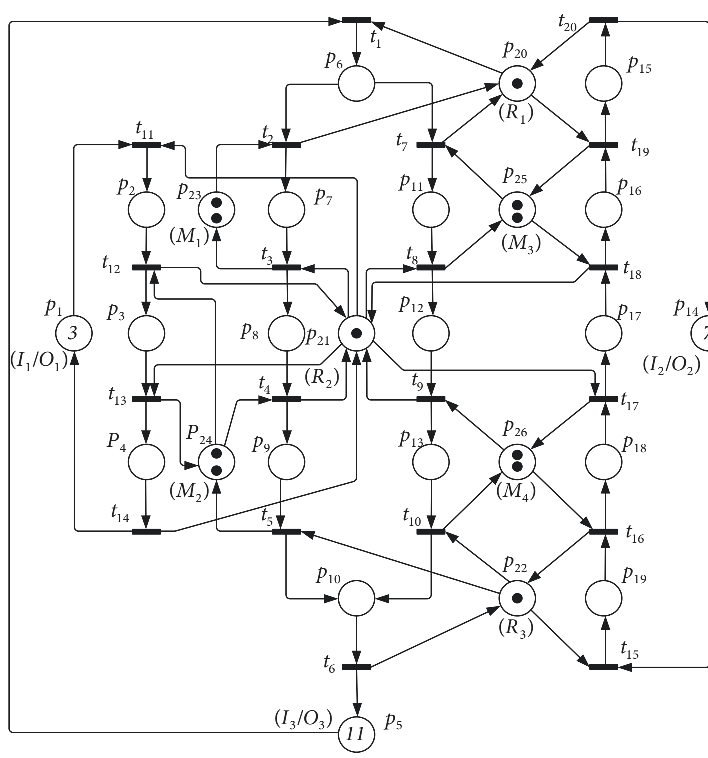

Figure 8: One classical complex S^3PR system [1].

26750 reachable markings and 120 dead markings when its idle places are in saturation. Therefore, we want to show that our control policy can not only recover all 120 dead markings of the complex PNM but reserve all original 26750 reachable markings.

According to our deadlock prevention policy, the optimal number of tokens in each idle place are $M_0(p_1) = 3$, $M_0(p_5) = 11$ and $M_0(p_{14}) = 7$ in this PNM. And 20 crucial dead markings in total are identified and are shown in Table 6. Please notice that the original dead markings are

TABLE 7: The 20 crucial control transitions of Figure 8.

No.	In formation of Crucial control Transitions
1	$O(t_{c21}) = (3p_1 + p_{21} + 2p_{24})$ $I(t_{c21}) = (p_2 + 2p_3)$
2	$O(t_{c22}) = (p_5 + 2p_{14} + p_{21} + 2p_{26})$ $I(t_{c22}) = (p_{12} + 2p_{18})$
3	$O(t_{c23}) = (2p_5 + p_{14} + p_{21} + 2p_{25})$ $I(t_{c23}) = (2p_{11} + p_{17})$
4	$O(t_{c24}) = (2p_5 + 2p_{14} + p_{21} + p_{22} + 2p_{26})$ $I(t_{c24}) = (p_{12} + p_{13} + p_{18} + p_{19})$
5	$O(t_{c25}) = (p_1 + 2p_5 + 3p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c25}) = (p_2 + 2p_9 + 2p_{18} + p_{19})$
6	$O(t_{c26}) = (p_1 + 2p_5 + 3p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c26}) = (p_3 + p_8 + p_9 + 2p_{18} + p_{19})$
7	$O(t_{c27}) = (3p_5 + p_{14} + p_{21} + p_{22} + 2p_{26})$ $I(t_{c27}) = (p_{12} + 2p_{13} + p_{19})$
8	$O(t_{c28}) = (p_1 + 3p_5 + 2p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c28}) = (p_2 + 2p_9 + p_{13} + p_{18} + p_{19})$
9	$O(t_{c29}) = (p_1 + 3p_5 + 2p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c29}) = (p_3 + p_8 + p_9 + p_{13} + p_{18} + p_{19})$
10	$O(t_{c30}) = (3p_5 + 3p_{14} + p_{20} + p_{21} + 2p_{23} + 2p_{25})$ $I(t_{c30}) = (p_6 + 2p_7 + 2p_{16} + p_{17})$
11	$O(t_{c31}) = (3p_5 + 3p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c31}) = (p_8 + 2p_9 + 2p_{18} + p_{19})$
12	$O(t_{c32}) = (p_1 + 4p_5 + p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c32}) = (p_3 + p_8 + p_9 + 2p_{13} + p_{19})$
13	$O(t_{c33}) = (p_1 + 4p_5 + p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c33}) = (p_2 + 2p_9 + 2p_{13} + p_{19})$
14	$O(t_{c34}) = (4p_5 + 2p_{14} + p_{20} + p_{21} + 2p_{23} + 2p_{25})$ $I(t_{c34}) = (p_6 + 2p_7 + p_{11} + p_{16} + p_{17})$
15	$O(t_{c35}) = (4p_5 + 2p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c35}) = (p_8 + 2p_9 + p_{13} + p_{18} + p_{19})$
16	$O(t_{c36}) = (5p_5 + p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c36}) = (p_8 + 2p_9 + 2p_{13} + p_{19})$
17	$O(t_{c37}) = (2p_1 + p_5 + p_{21} + 2p_{24})$ $I(t_{c37}) = (2p_3 + p_8)$
18	$O(t_{c38}) = (2p_1 + p_5 + 3p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c38}) = (p_2 + p_3 + p_9 + 2p_{18} + p_{19})$
19	$O(t_{c39}) = (2p_1 + 2p_5 + 2p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c39}) = (p_2 + p_3 + p_9 + p_{13} + p_{18} + p_{19})$
20	$O(t_{c40}) = (2p_1 + 3p_5 + p_{14} + p_{21} + p_{22} + 2p_{24} + 2p_{26})$ $I(t_{c40}) = (p_2 + p_3 + p_9 + 2p_{13} + p_{19})$

120. In our deadlock prevention policy, we can identify the 20 deadlock markings and use them to calculate and obtain 20 controllers.

First of all, please refer to Table 6; the first crucial dead marking (i.e., $p_2 + 2p_3 + p_{20} + p_{22} + 2p_{23} + 2p_{25} + 2p_{26}$) is identified when the three idle places are in $M_0(p_1) = 3$, $M_0(p_5) = 0$, and $M_0(p_{14}) = 0$, respectively. And then, its relative controller ($I(t_{c21}) = p_2 + 2p_3$ and $O(t_{c21}) = 3p_1 + p_{21} + 2p_{24}$) can be calculated and obtained (i.e., the first set in Table 7). Eight dependent dead markings (M_1, M_2, M_{45}, M_{55}, M_{66}, M_{80}, M_{93}, and M_{105} are checked in saturated situation based on TINA) also become live simultaneously (please refer to No. 1 in Table 8). Please notice that all markings presented in Table 9 are identified initially by the famous software TINA. Due to the space limitation of this paper, we just show the marking number in Table 8. For further detailed information, please refer to all data by using TINA. Anyway, 120 dead markings will reduce to 112. Surely, the total number of all reachable markings still is 26750. Keep doing the same process and step by step, 112 dead markings eventually become live. In sum, our deadlock prevention policy based on control transition live all reachable markings.

4.3. Comparison with Existing Researches. In this subsection, we make some comparisons with existing literature based on the classical complex PNM. Please note that the "No. C_P" represents the number of control places, "No. C_T" represents the number of control transitions, "No. R_L" represents the number of live reachable markings, and "% R" represents the recover percent relative to original 26750 markings. From Table 9, it is obvious that our algorithm is the only algorithm which can achieve 100 % maximally permissive markings among all published researches.

5. Conclusions

In this paper, we successfully propose one novel iterative method to obtain maximally permissive states based on control transitions. Generally, "maximal permissive" means the number of legal markings in an original deadlock PNM. Further, it does not include the initial illegal markings. For example, they just can obtain 21581 maximally permissive markings in the second example of Section 4. However, our proposed algorithm does obtain the real 26750 maximally permissive markings. In other words, our deadlock prevention recovers one deadlock PNM. In our approach, we do not consider any controllability of siphons. We just focus on crucial dead markings. All markings are live and reachable once these crucial markings are controlled. Our approach is now suitable to all S^3PR model. The future work is to reduce the computing time (i.e., reduce running the number of reachability graph or reduce the number of controllers) and extends the method applying for more general models in Petri net.

Data Availability

The data used to support the findings of this study are included within the supplementary information file(s).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Table 8: The 20 dependent dead markings checked in saturated situation based on TINA of Table 6.

No.	Dependent Deadlock Markings	The Number of Deadlock Markings After Relative controllers are added into PNM
1	$M_1, M_2, M_{45}, M_{55}, M_{66}, M_{80}, M_{93}, M_{105}$	112
2	$M_6, M_7, M_{14}, M_{15}, M_{21}, M_{22}, M_{31}, M_{32}, M_{33}, M_{34}, M_{78}, M_{79}$	100
3	$M_{24}, M_{26}, M_{27}, M_{28}, M_{29}, M_{30}, M_{47}, M_{48}, M_{51}, M_{52}, M_{53}, M_{58}$	88
4	$M_{42}, M_{43}, M_{50}, M_{54}, M_{56}, M_{57}, M_{60}, M_{92}, M_{117}, M_{118}, M_{119}, M_{120}$	76
5	M_{17}, M_{23}, M_{73}	73
6	M_{12}, M_{13}, M_{71}	70
7	$M_{96}, M_{99}, M_{100}, M_{101}, M_{102}, M_{104}, M_{108}, M_{111}, M_{112}, M_{113}, M_{114}, M_{116}$	58
8	M_{38}, M_{44}, M_{87}	55
9	M_{49}, M_{65}, M_{85}	52
10	$M_{67}, M_{69}, M_{70}, M_{72}, M_{74}, M_{76}, M_{81}, M_{83}, M_{84}, M_{86}, M_{88}, M_{90}$	40
11	M_{19}, M_{20}, M_{75}	37
12	M_{95}, M_{107}	35
13	M_{97}, M_{109}	33
14	$M_3, M_8, M_{10}, M_{11}, M_{16}, M_{18}, M_{36}, M_{37}, M_{39}, M_{61}, M_{63}, M_{64}$	21
15	M_{40}, M_{41}, M_{89}	18
16	M_{98}, M_{110}	16
17	$M_4, M_5, M_{59}, M_{62}, M_{77}, M_{91}, M_{103}, M_{115}$	8
18	M_9, M_{25}, M_{68}	5
19	M_{35}, M_{46}, M_{82}	2
20	M_{94}, M_{106}	0

Table 9: Comparison with existing literature.

	Ezpeleta et al. (1995) [1]	Huang et al (2001) [3]	Li & Zhou (2004) [5]	Uzam & Zhou (2006) [34]	Chao (2009b) [26]	Piroddi et al. (2008) [10]	Chen et al (2011) [27]	Chen et al (2012) [28]	Hong et al. (2015) [29]	Row et al. [This paper]
No. C_P	18	16	5	19	7	13	17	6	9	0
No. C_T	0	0	0	0	0	0	0	0	0	20
No.R_L	6287	12656	15999	21562	21585	21581	21581	21581	20444	26750
% R	23.5%	47.3%	59.8%	80.6%	80.7%	80.7%	80.7%	80.7%	94.7%	100%

Supplementary Materials

The detailed information of all 120 deadlocks of Figure 8 based on the famous software TINA [31]. *(Supplementary Materials)*

References

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

[2] G. Liu, Z. Li, and C. Zhong, "New controllability condition for siphons in a class of generalised Petri nets," *IET Control Theory & Applications*, vol. 4, no. 5, pp. 854–864, 2010.

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

[4] Y. S. Huang, M. D. Jeng, X. L. Xie, and T. H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems,"' *IEEE Transactions on Systems, Man and Cybernetics, Part A, Systems and Humans*, vol. 36, no. 6, pp. 2152–2160, 2006.

[5] 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, Cybernetics A, Humans*, vol. 34, no. 1, pp. 38–51, 2004.

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

[7] H. Liu, K. Xing, W. Wu, M. C. Zhou, and H. Zou, "Deadlock Prevention for Flexible Manufacturing Systems via Controllable Siphon Basis of Petri Nets," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 45, no. 3, pp. 519–529, 2015.

[8] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using petri net models with resources and the theory of region," *International Journal of Advanced Manufacturing Technology*, vol. 19, no. 3, pp. 192–208, 2002.

[9] M. Uzam and M. Zhou, "An iterative synthesis approach to Petri net-based deadlock prevention policy for flexible manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 37, no. 3, pp. 362–371, 2007.

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

[11] Y.-L. Pan, Y.-S. Huang, M. Jeng, and S.-L. Chung, "Enhancement of an efficient control policy for FMSs using the theory of regions and selective siphon method," *The International Journal of Advanced Manufacturing Technology*, vol. 66, no. 9-12, pp. 1805–1815, 2013.

[12] K. Barkaoui and I. B. Abdallah, "Deadlock prevention method for a class of FMS," in *Proceedings of the International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 5, pp. 4119–4124, IEEE, October 1995.

[13] Y. Huang, Y. Pan, and P. Su, "Transition-based deadlock detection and recovery policy for FMSs using graph technique," *ACM Transactions on Embedded Computing Systems*, vol. 12, no. 1, pp. 1–13, 2013.

[14] Y. Hou and K. Barkaoui, "Deadlock analysis and control based on petri nets: a siphon approach review," *Advances in Mechanical Engineering*, vol. 9, no. 5, pp. 1–30, 2017.

[15] G. Liu and K. Barkaoui, "A survey of siphons in Petri nets," *Information Sciences*, vol. 363, pp. 198–220, 2016.

[16] H. Yi-Sheng and P. Yen-Liang, "Enhancement of an efficient liveness-enforcing supervisor for flexible manufacture systems," *The International Journal of Advanced Manufacturing Technology*, vol. 48, pp. 725–737, 2010.

[17] 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 & Applications*, vol. 5, no. 9, pp. 1069–1078, 2011.

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

[19] Y.-L. Pan, C.-F. Yang, and M.-D. Jeng, "Enhancement of selective siphon control method for deadlock prevention in FMSs," *Mathematical Problems in Engineering*, vol. 2015, Article ID 196514, 6 pages, 2015.

[20] Y.-L. Pan, Y.-S. Huang, Y.-S. Weng, W. Wu, and M. Jeng, "Computationally improved optimal control methodology for linear programming problems of flexible manufacturing systems," *Journal of Applied Mathematics*, vol. 2013, Article ID 294835, 11 pages, 2013.

[21] D. Y. Chao and Y. Pan, "Uniform formulas for compound siphons, complementary siphons and characteristic vectors in deadlock prevention of flexible manufacturing systems," *Journal of Intelligent Manufacturing*, vol. 26, no. 1, pp. 13–23, 2015.

[22] T.-C. Row and Y.-L. Pan, "Maximally permissive deadlock prevention policies for flexible manufacturing systems using control transition," *Advances in Mechanical Engineering*, vol. 10, no. 7, pp. 1–10, 2018.

[23] X. Zhang and M. Uzam, "Transition-based deadlock control policy using reachability graph for flexible manufacturing systems," *Advances in Mechanical Engineering*, vol. 8, no. 2, pp. 1–9, 2016.

[24] Y. Chen, Z. Li, A. Al-Ahmari, N. Wu, and T. Qu, "Deadlock recovery for flexible manufacturing systems modeled with Petri nets," *Information Sciences*, vol. 381, pp. 290–303, 2017.

[25] Y. Pan, C. Tseng, and T. Row, "Design of improved optimal and suboptimal deadlock prevention for flexible manufacturing systems based on place invariant and reachability graph analysis methods," *Journal of Algorithms & Computational Technology*, vol. 11, no. 3, pp. 261–270, 2017.

[26] D. Y. Chao, "Reaching more states for control of FMS," *International Journal of Production Research*, vol. 48, no. 4, pp. 1217–1220, 2009.

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

[28] Y. F. Chen, Z. W. Li, and M. C. Zhou, "Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems," *IEEE Trans. System, Man, Cybernetics A*, vol. 42, no. 3, pp. 615–629, 2012.

[29] L. Hong, Y. Hou, J. F. Jing et al., "Deadlock prevention policy with behavioral optimality or suboptimality achieved by the redundancy identification of constraints and the rearrangement of monitors," *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 579623, 15 pages, 2015.

[30] S. Wang, W. Wu, and J. Yang, "Deadlock prevention policy for a class of petri nets based on complementary places and elementary siphons," *Journal of Intelligent Manufacturing*, vol. 26, no. 2, pp. 321–330, 2015.

[31] F. Wang, K. Xing, X. Xu, and L. Han, "Research on finding elementary siphon in a class of petri nets," *Springer, Communications and Information Processing*, vol. 289, no. 2, pp. 435–442, 2012 (Arabic).

[32] M. H. Abdul-Hussin, "Elementary siphons of petri nets and deadlock control in FMS," *Journal of Computer and Communications*, vol. 3, no. 7, pp. 1–12, 2015.

[33] 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.

[34] 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.

[35] T. Murata, "Petri nets: properties, analysis and application," *Pro. IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[36] Z. W. Li, H. M. Hanish, and M. C. Zhou, "Deadlock prevention based on structure reuse of petri net supervisors for flexible manufacturing systems,"' *Systems, Man and Cybernetics, part A: System and Humans, IEEE Transactions*, vol. 42, no. 1, 2012.

[37] S. Wang, C. Wang, M. Zhou, M. Zhou, Z. Li, and Z. Li, "A method to compute strict minimal siphons in a class of petri nets based on loop resource subsets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems and Humans*, vol. 42, no. 1, pp. 226–237, 2012.

[38] H. Liu, K. Xing, W. M. Wu, M. C. Zhou, and H. L. Zou, "Deadlock prevention for flexible manufacturing systems via controllable siphon basis of petri nets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 3, pp. 519–529, 2015.

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

[40] R. David and H. Alla, *Discrete, Continuous and Hybrid PetriNets*, Springer-Verlag, Berlin, Germany, 2nd edition, 2005.

[41] T. Murata, "State equation, controllability, and maximal matchings of petri net," *IEEE Transactions on Automatic Control*, vol. 22, no. 3, pp. 412–416, 1977.

[42] M. A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modeling with Generalized Stochastic Petri Nets*, John Wiley and Sons Press, 1st edition, 1995.

[43] "TINA (Time petri Net Analyzer) has been developed in the OLC then VerTICS, research groups of LAAS/CNRS," http://projects.laas.fr/tina//download.php.