Maximally permissive deadlock prevention policies for flexible manufacturing systems using control transition

Ter-Chan Row¹ and Yen-Liang Pan²,³

Abstract
Nowadays, many kinds of flexible manufacturing systems are used to process many complex manufacturing works due to their machine flexibility and routing flexibility. However, such competition (i.e. robots and machines) for shared resources by concurrent job processes can lead to the problem of a system deadlock. In existing researches, almost experts adopted place-based as controllers to solve the deadlock problems of flexible manufacturing systems whatever the concept of siphons or the reachability graph method are used. Among them, only the reachability graph ones can obtain maximally permissive live states. In this article, the authors try to propose one novel transition-based deadlock prevention concept to solve flexible manufacturing system's deadlock problem. In addition, two algorithms are developed to support above concept. The experimental results indicate that the proposed policy not only can obtain maximally permissive controllers but also recover all original deadlock markings.

Keywords
Deadlock prevention, maximally permissive, Petri nets, flexible manufacturing systems, reachability graph

Date received: 25 June 2017; accepted: 15 June 2018

Handling Editor: Artde Lin

Introduction
Flexible manufacturing systems (FMSs)¹ are a variable working system using all kinds of machines, robots, and conveyors that are shared resources. The FMSs may consider varied resource allocation systems (RASs).² The deadlock occurs if two or more competing actions in shared resources. Thus, engineers must consider and solve the deadlock problems when they want to design and control the deadlock-prone FMS. Petri net³–⁸ is good to model and analyze FMSs because of the wonderful dynamic and graphic representation. Therefore, almost researchers use Petri nets as their tool to obtain all kinds of controllers. Furthermore, there are three kinds of controllers used to solve FMS' deadlock problems. They are control places based on siphon concept,¹⁹–¹⁴ control places in reachability graph analysis,¹⁵–¹⁷ control arcs,² and control transitions.¹⁸–²⁰

Liu et al.⁹ presented a concept of controllable siphon basis that adding a control place and related arcs to each strict minimum siphon forming a controllable siphon basis. A new deadlock prevention policy was proposed by an algorithm for constructing a controllable siphon basis. This novel deadlock prevention policy by controllable siphon basis is better than the elementary siphon algorithm.¹¹ But how to find a controllable

¹Department of Electronic Engineering, Army Academy, Taoyuan, Taiwan
²Mathematics and Physics Division, General Education Center, Air Force Academy, Kaohsiung, Taiwan
³Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan

Corresponding author:
Yen-Liang Pan, Mathematics and Physics Division, General Education Center, Air Force Academy, Kaohsiung 820, Taiwan.
Email: peterpan960326@gmail.com

Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (http://www.creativecommons.org/licenses/by/4.0/) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
siphon basis to get a best performance is a heavy cost to finish in the large Petri net (PN). Wang et al.\textsuperscript{10} proposed an algorithm using an emptied siphon based on loop resource subnets to extract a desired emptied strict minimum siphon (DESMS). Then, they applied the mixed-integer programming (MIP) to obtain a maximally permissive liveness-enforcing supervisor. When one extracts a desire emptied strict minimum siphon, it must be going through three steps of processing. The complexity increases. Although the MIP obtains a maximally permissive liveness-enforcing supervisor, the marking states of PN will be sacrificed some.

Uzam and colleagues\textsuperscript{15,16} develop a deadlock prevention policy based on theory of regions to obtain maximal liveness performance. However, the policy fails to determine all sets of event-state-separation-problems (ESSPs), and its application seems limited to some special nets only. Therefore, some works\textsuperscript{21–24} aim to develop a computationally more efficient optimal deadlock control policy based on the theory of regions.

Another new method named interval inhibitor arcs\textsuperscript{2} is proposed. Two main stages are present as follows: the first stage is to analyze the reachability graph of PN. The next step is searching the good markings, dangerous markings, bad markings, and dead markings. And, it prevents good markings into the bad markings. Finally, the deadlock system will be live. Note that this approach needs the reachability graph to distinguish the various markings. When the PN system is large, there are too many constraints and variables to consider. Thus, the method is hard likelihood to obtain a live PN. Furthermore, these too many constraints and variables cause not to be an achieved live PN.

Recently, Chen et al.\textsuperscript{25,26} propose one novel technology called maximal number of forbidding first bad marking (FBM) problem (MFFP) to obtain optimal states with a small number of control places. It solves the time-consuming problem successfully. According to our survey, the MFFP method seems the best one in existing literature based on place-based controller concept. However, this policy merely can be used in some special nets. For general cases, deadlocks could still exist.

Above means adopt place-based or arcs-based to prevent deadlocks from occurring. The last mean is transition-based controller concept. Huang et al.\textsuperscript{18} first used the concept of control transitions to solve the deadlock problem. Advantage of control transition is that it can obtain more permissive markings using state equations.\textsuperscript{3} However, it is pity that it cannot identify the correct optimal number of reachability graph of one FMS. In following, Zhang and Uzam\textsuperscript{19} apply the reachability graph to obtain the control transitions using formulations of the set covering technique.\textsuperscript{27} However, the efficiency seems too low of the proposed method since it depends on the reachability graph completely. In addition, it also cannot identify the correct optimal number of reachability graph of one FMS in advance.

Based on above disadvantages of two transition-based control policies, this article proposes two improved deadlock prevention policies. The two transition-based deadlock prevention policies not only can identify correct maximally permissive number of reachable markings of one FMS in advance but also recover all original illegal markings.

The rest of this article is organized as follows. Section “Preliminaries” briefly reviews the basic concept of PNs used in this article. Section “The proposed algorithm of transition-based controllers” presents our proposed two transition-based deadlock prevention policies. Section “Examples” gives two examples to test the efficiency of the proposed policy. Section “Conclusion” makes the conclusion.

\section*{Preliminaries}

One PN original structure is defined as follows.

\begin{definition}
N = (P, T, F, W), where P = \{p_1, p_2, p_3, ..., p_m\} is a finite, non-empty, and disjoint set of places. T = \{t_1, t_2, t_3, ..., t_n\} is a finite, non-empty, and disjoint set of transitions. F \subseteq (P \times T) \cup (T \times P) is a set of arcs called flow relation. The element F is a set of directed arcs with the arrows from places to transitions or from transitions to places. W(\cdot) is assigned the weight to an arc. Furthermore, the set of arcs in PN are nonnegative integer. A PN n is called ordinary when its weight of arcs is equal to one. Furthermore, for one real FMS net (N) modeled by PN, we called it PN model (PNM). It can be defined as follows.
\end{definition}

\begin{definition}
PNM = (P_0 \cup \{p_0\} \cup P_{Ri}, T_i, F_i), where P_i is the operation place. p_0 is the process-idle place. P_{Ri} is the resource places. T_i is the set of transitions. F_i is the set of arcs with weight.
\end{definition}

\begin{definition}
Enabling and firing rule:\textsuperscript{18} in one PNM, if a node x \in P \cup T, then the preset of node x is defined \{y \in P \cup T \mid (y, x) \in F\}. The post set of node x is defined as x* = \{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 t and M(p) \geq W(p, t) that will be denoted as M[t\geq]. Afterward firing and enabled a transition t at M will get a new marking M'. It indicates as M[t > M']. A marking M reaches to M' shown by M[\sigma > M'] if there exists a firing sequence \sigma = t_1, t_2, t_3, ..., t_n. The set of marking reachable from M in one PNM is called a
reachability set as $R(N, M_0)$. The $R(N, M_0)$ is live iff $\forall t \in T$ is enabled and dead iff $\exists t \in T$ is disabled.

Definition 4. State equation:

\[ M_k = M_{k-1} + A^T \cdot u_k, \quad k = 1, 2, \ldots \]

where $M_k$ denotes the change of the marking. $Z^t$ denotes a number of times that transition $i$ fires at the $k$th firing. The $k$th row of the incidence matrix $A$ deduces the change in the marking. $M_0$ is reachable if for every reachable state $s$, $s 
subseteq \{f_p\}$.

Definition 5. Necessary reachability condition:

\[ A^T \cdot x = \Delta M \]

where $\Delta M = M_d - M_0$. Here, $x$ is a $n \times 1$ column vector of nonnegative integers that is called the firing-count vector. The $i$th entry of $x$ deduces the number of times that transition $i$ must fire to change $M_0$ to $M_d$.

Definition 6. Livelock:

\[ x \in Z^m, x \neq 0, \]

is a T-invariant if $A^T \cdot x = 0$, where $x$ is called the firing-count vector and matrix $A$ denotes the change of the marking. $Z^m$ is the firing set of $x$.

One can deduce that a livelock is formed if a reachability graph is not a T-invariant. The characteristic of a livelock marking is not a deadlock and cannot reach the home state.

Definition 7. Reversibility:

1. A PN $N$ has a home state $M_h$ for an initial state $M_0$ if for every reachable state $M_i \in R(N, M_0)$, a firing sequence $\sigma$ exists such that $M_0 \sigma > M_h$.
2. A PN $N$ is reversible for an initial state $M_0$ if $M_0$ is a home state.

Definition 8. Simple sequential processes with resources (S$^R$PR):

\[ N_i = (P_i \cup \{p_i^0\}) \cup P_{Ri}, T, F, W, \]

where:

1. Places in $P_i$ are called operation places. $P_{Ri}$ are called resource ones. $\{p_i^0\}$ are called the process-idle places of $N_i$.
2. $P_i \neq \varnothing$, $p_i^0 \notin P_i$, $P_{Ri} \neq \varnothing$, $(P_i \cup \{p_i^0\}) \cap P_{Ri} = \varnothing$.

Figure 1. One small Petri net.

Definition 9. Saturated number of tokens in idle places:

for one real PNM, the size of its reachability graph will become larger when the number of tokens in idle places is adding from small to large. Finally, the number of reachable states will reach to one maximal number whatever the number of tokens increased in idle places.

Under the situation, we call the number of tokens saturated number. Please note that the situation also can be called the optimal number of tokens.

For example, in Figure 1, Table 1 shows the different number of tokens in two the idle places (i.e. $p_1$ and $p_0$). From Table 1, we can see that the PNM reaches the maximal reachable states when the $(p_1, p_0)$ is $(3, 3)$. The experimental groups 21–24 are just used to present that the maximal reachable states of the PNM are equal to 15. Furthermore, the maximal reachable states are 15 whatever how large number of tokens is added into or increased into the idle place. Therefore, the set of $(3, 3)$...
is what we want, it is the saturated number of tokens in idle places of this PNM.

**Definition 10.** The first deadlock marking (FDM): based on Definition 9, for one real PNM, one deadlock marking appear first in this system while the number of tokens for idle places is adding from small to large. The deadlock marking is defined as the FDM.

**The proposed algorithm of transition-based controllers**

In this section, two novel algorithms based on control transitions will be developed. Please note that the proposed algorithms can not only solve the deadlock problem but also recover all original markings of one PNM whatever they are live or deadlock initially. In other words, in this article, the so-called “maximally permissive states” means that all original deadlock markings can be further recovered under the proposed algorithms. Please also note that under the two algorithms, the saturated number of tokens in one PNM’s idle places can be identified. Therefore, the maximal reachable markings of one PNM’s reachability graph can then calculated correctly. Huang et al.’s\(^{18}\) deadlock prevention policy just uses the dead marking and the initial marking to calculate the control transition. In other words, it does not know what the saturated number of tokens is. In this article, we can identify the correct and maximal number of one PNM’s reachable markings once the framework is established.

One small typical example\(^6\) shown in Figure 1 is used to illustrate the proposed algorithms. In Figure 1, one can see that there are two idle places (i.e. \(p_1\) and \(p_3\)) in this PNM, and their saturated or called optimal value is equal to 3. Under this situation, the number of its reachable markings shown in Figure 2 is equal to 15. Please note that there are four illegal markings (i.e. three quasi-dead markings and one deadlock marking) in the reachability graph. The detailed information is shown in Tables 1 and 2, respectively.

**Table 1.** The variety of the number of RG when given different number tokens of \((p_1, p_6)\).

<table>
<thead>
<tr>
<th>Group no.</th>
<th>(M(p_1))</th>
<th>(M(p_6))</th>
<th>Number of RGs</th>
<th>Group no.</th>
<th>(M(p_1))</th>
<th>(M(p_6))</th>
<th>Number of RGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>13</td>
<td>1</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>0</td>
<td>5</td>
<td>14</td>
<td>1</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0</td>
<td>6</td>
<td>15</td>
<td>2</td>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>6</td>
<td>16</td>
<td>2</td>
<td>2</td>
<td>13</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>0</td>
<td>6</td>
<td>17</td>
<td>3</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>18</td>
<td>2</td>
<td>4</td>
<td>14</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>2</td>
<td>5</td>
<td>19</td>
<td>3</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>20</td>
<td>3</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>4</td>
<td>6</td>
<td>21</td>
<td>3</td>
<td>3</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>5</td>
<td>6</td>
<td>22</td>
<td>3</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>23</td>
<td>4</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>2</td>
<td>9</td>
<td>24</td>
<td>5</td>
<td>5</td>
<td>15</td>
</tr>
</tbody>
</table>

RG: reachability graph.

**Figure 2.** The reachability graph of Figure 1.
must find another relatively firing sequence from \( M_d \) to \( M_0 \) as follows: \( M_d - A^T u_d = M_0 \). The equation can be rearranged as \( M_0 = M_d - A^T u_d \). Then \( M_0 = M_d + A^T (-u_d) \). Let \( y_d = -u_d \). Then, one can obtain the new state equation for calculating transition-based controllers of a deadlock-prone PNM as follows

\[
M_0 = M_d + A^T \cdot (y_d)
\]

From above equation, undoubtedly, one can identify one set of firing sequence such that \( M_d \) can be recovered.

**Two algorithms based on different viewpoints**

Two algorithms based on different viewpoints are present as follows:

- Algorithm 1: based on the all reachability graph viewpoint.

First, from Figure 2 and Table 2, one can identify that the property of the initial marking and the deadlock marking are \( M_0 = 3p_1 + 3p_6 + 2p_7 + 2p_8 \) and \( M_6 = p_1 + 2p_2 + 2p_4 + p_6 \), respectively. Therefore, according to definitions 4 and 5, the control transition equation (CTE) can be listed as follows: \( M_h = M_d + (O(T_{c0}) - I(T_{c0})) \). Thus, the output control transition is \( O(t_7) = 2p_1 + 2p_6 + 2p_7 + 2p_8 \) and the input control transition is \( I(t_7) = 2p_2 + 2p_4 \). When the set of control transition is put into the original PNM shown in Figure 3, all 15 reachable markings including the 4 illegal markings are live shown in Figure 4.

- Algorithm 2: based on the FDM viewpoint.

In the following, according to Definition 10 one can identify that the system meets the FDM while the two idle places \( (p_1, p_6) \) are given \( (1, 1) \). The detailed information is shown in Tables 1 and 3.

Based on Table 2, one can realize that the saturated number of the two idle places \( (p_1, p_6) \) is \( (3, 3) \). It means

<table>
<thead>
<tr>
<th>Marking no.</th>
<th>Classification</th>
<th>Information of marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_0 )</td>
<td>Initial marking</td>
<td>( M_0 = p_1 + p_6 + 2p_7 + 2p_8 )</td>
</tr>
<tr>
<td>( M_1 )</td>
<td>Live marking</td>
<td>( M_1 = p_2 + p_6 + p_7 + 2p_8 )</td>
</tr>
<tr>
<td>( M_2 )</td>
<td>Live marking</td>
<td>( M_2 = p_1 + 2p_2 + 2p_4 + p_6 )</td>
</tr>
<tr>
<td>( M_3 )</td>
<td>Live marking</td>
<td>( M_3 = p_1 + p_2 + p_3 + 3p_6 + p_7 )</td>
</tr>
<tr>
<td>( M_4 )</td>
<td>Live marking</td>
<td>( M_4 = p_1 + p_2 + 2p_4 + p_6 )</td>
</tr>
<tr>
<td>( M_5 )</td>
<td>Quasi-dead marking</td>
<td>( M_5 = p_1 + 2p_2 + p_4 + 2p_6 + p_8 )</td>
</tr>
<tr>
<td>( M_6 )</td>
<td>Deadlock marking</td>
<td>( M_6 = p_1 + p_2 + 2p_4 + p_6 )</td>
</tr>
<tr>
<td>( M_7 )</td>
<td>Live marking</td>
<td>( M_7 = 2p_1 + p_3 + 3p_6 + 2p_7 )</td>
</tr>
<tr>
<td>( M_8 )</td>
<td>Quasi-dead marking</td>
<td>( M_8 = 2p_1 + p_2 + p_4 + 2p_6 + p_7 + p_8 )</td>
</tr>
<tr>
<td>( M_9 )</td>
<td>Quasi-dead marking</td>
<td>( M_9 = 2p_1 + p_2 + 2p_4 + p_6 + p_8 )</td>
</tr>
<tr>
<td>( M_{10} )</td>
<td>Live marking</td>
<td>( M_{10} = 3p_1 + p_4 + 2p_6 + 2p_7 + p_8 )</td>
</tr>
<tr>
<td>( M_{11} )</td>
<td>Live marking</td>
<td>( M_{11} = 3p_1 + 2p_4 + p_6 + p_7 )</td>
</tr>
<tr>
<td>( M_{12} )</td>
<td>Live marking</td>
<td>( M_{12} = 3p_1 + p_4 + p_5 + p_6 + p_8 )</td>
</tr>
<tr>
<td>( M_{13} )</td>
<td>Live marking</td>
<td>( M_{13} = 3p_1 + 2p_4 + p_5 )</td>
</tr>
<tr>
<td>( M_{14} )</td>
<td>Live marking</td>
<td>( M_{14} = 3p_1 + p_5 + 2p_6 + 2p_8 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Marking no.</th>
<th>Classification</th>
<th>Information of marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_0 )</td>
<td>Initial marking</td>
<td>( M_0 = p_1 + p_6 + 2p_7 + 2p_8 )</td>
</tr>
<tr>
<td>( M_1 )</td>
<td>Live marking</td>
<td>( M_1 = p_2 + p_6 + p_7 + 2p_8 )</td>
</tr>
<tr>
<td>( M_2 )</td>
<td>Live marking</td>
<td>( M_2 = p_1 + 2p_2 + 2p_4 + p_6 )</td>
</tr>
<tr>
<td>( M_3 )</td>
<td>First deadlock marking (FDM)</td>
<td>( M_3 = p_1 + 2p_2 + 2p_4 + p_7 + p_8 )</td>
</tr>
<tr>
<td>( M_4 )</td>
<td>Live marking</td>
<td>( M_4 = p_1 + p_2 + 2p_4 + p_8 )</td>
</tr>
<tr>
<td>( M_5 )</td>
<td>Live marking</td>
<td>( M_5 = p_1 + p_2 + 2p_4 + p_8 )</td>
</tr>
</tbody>
</table>
that the maximal number of the reachability graph of the PNM is equal to 15 whatever how many tokens are added into \((p_1, p_8)\). This is also called the full reachability viewpoint. On the contrary, in Algorithm 2, we just check when the first one deadlock marking will be produced by the PNM. Since the 12th group starts to produce the FDM, we use its data to calculate the controller. From Table 3, one can identify the deadlock marking is \(M_3\), and its property is \(p_2 + p_4 + p_7 + p_8\). Besides, the property of the initial marking \(M_0\) is \(p_1 + p_6 + 2p_7 + 2p_8\). From CTE, one can obtain the output control transition \(O(t_8) = p_1 + p_6 + p_7 + p_8\) and the input control transition \(I(t_8) = p_2 + p_4\). The PNM is deadlock-free when we put the controller into it. Figures 5 and 6 show the results.

Algorithms 1 and 2 present the very different viewpoints. Furthermore, Algorithm 1 adopts the viewpoint of global framework. However, Algorithm 2 adopts the viewpoint of local framework. Whatever what viewpoint is adopted, the PNM will be live and can also hold the maximally permissive states. More important, both of the proposed algorithms recover the all deadlock markings.

**Theorem 1.** Proposed algorithms can recover the deadlock PNM and is maximally permissive.

**Proof.** According to equation (1), one set firing sequence must exist, such that, \(M_d\) can be led back to \(M_0\) once controllers exist. And, the controllers will not make the dead markings vanished. Therefore, the proposed algorithms can recover the deadlock PNM and is maximally permissive.

**Examples**

In this section, we want to present the high efficiency of the two proposed algorithms by testing two classical examples.\(^{15,18}\) The first example is used to present the proposed algorithms and the second example is used to make comparison with previous literature. First of all, two deadlock prevention policies are listed based on the two algorithms as follows.

---

**Figure 4.** The reachability graph of Figure 3.

**Figure 5.** A live Petri net system based on Algorithm 2.

**Figure 6.** The reachability graph of Figure 5.
Deadlock Prevention Policy I.

In the following, we use example 1 to test the proposed Deadlock Prevention Policies I and II. Figure 7 shows its PNM. According to the Deadlock Prevention Policy I, the full reachability graph is surely needed to run. Figure 8 presents the full reachability graph of Figure 7. Due to the limitation of paper space, we just show the properties of initial markings and two deadlock markings in Table 4.

Then, we can obtain two controllers $t_9$ and $t_{10}$ based on Table 4 and the CTE. The detailed information of $t_9$ and $t_{10}$ are $I(t_9) = p_2 + p_5 + p_{10}$, $O(t_9) = 2p_1 + p_5 + p_6 + p_9 + p_{11}$, and $I(t_{10}) = p_2 + p_7 + p_{10}$, $O(t_{10}) = p_1 + p_3 + p_6 + p_9 + 2p_{11}$, respectively. When the set of control transition is put into the original PNM of Figure 7, all 20 reachable markings including the 5 illegal markings (i.e., $M_7$, $M_9$, $M_{10}$, $M_{11}$, and $M_{12}$) are live (please refer to Figures 9 and 10).

Deadlock Prevention Policy II.

<table>
<thead>
<tr>
<th>Marking no.</th>
<th>Classification</th>
<th>Information of Marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M_0$</td>
<td>Initial marking</td>
<td>$M_0 = 3p_1 + p_3 + p_4 + 3p_{11}$</td>
</tr>
<tr>
<td>$M_7$</td>
<td>Deadlock marking</td>
<td>$M_7 = p_1 + p_2 + p_5 + p_{10} + 2p_{11}$</td>
</tr>
<tr>
<td>$M_{12}$</td>
<td>Deadlock marking</td>
<td>$M_{12} = 2p_1 + p_2 + p_7 + p_{10} + p_{11}$</td>
</tr>
</tbody>
</table>

Step 1: Input one deadlock-prone PNM
Step 2: Identify saturated number of idle places
Step 3: Run its full RG based on saturated number of idle places
Step 4: Identify the properties of initial and all deadlock markings
Step 5: Calculate all the controllers based on CTE
Step 6: Add controllers into the deadlock-prone PNM
Step 7: Output one deadlock-free PNM

Step 1: Input one deadlock-prone PNM
Step 2: Run partial RG until the first deadlock marking (FDM) is produced
Step 3: Identify the properties of initial and first deadlock markings
Step 4: Calculate the controller based on CTE
Step 5: Add the controller into the deadlock-prone PNM
Step 6: Repeat steps 2–5 until initial reachable markings are live
Step 7: Output one deadlock-free PNM

In the following, we use example 1 to test the proposed Deadlock Prevention Policies I and II. Figure 7 shows its PNM. According to the Deadlock Prevention Policy I. The full reachability graph is surely needed to run. Figure 8 presents the full reachability graph of Figure 7.

Due to the limitation of paper space, we just show the properties of initial markings and two deadlock markings in Table 4.

Then, we can obtain two controllers $t_9$ and $t_{10}$ based on Table 4 and the CTE. The detailed information of $t_9$ and $t_{10}$ are $I(t_9) = p_2 + p_5 + p_{10}$, $O(t_9) = 2p_1 + p_5 + p_6 + p_9 + p_{11}$, and $I(t_{10}) = p_2 + p_7 + p_{10}$, $O(t_{10}) = p_1 + p_3 + p_6 + p_9 + 2p_{11}$, respectively. When the set of control transition is put into the original PNM of Figure 7, all 20 reachable markings including the 5 illegal markings (i.e., $M_7$, $M_9$, $M_{10}$, $M_{11}$, and $M_{12}$) are live (please refer to Figures 9 and 10).
In the following, we adopt Deadlock Prevention Policy II to test the example 1. The system meets two FDM while the two idle places \((p_1, p_11)\) are given \((1, 1)\). The detailed information is shown in Table 5.

Similarly, we can obtain two controllers \(t_{11}\) and \(t_{12}\) based on Table 5 and the CTE. The detailed information of \(t_{11}\) and \(t_{12}\) are \(I(t_{11}) = p_5 + p_{10}\), \(O(t_{11}) = p_1 + p_6 + p_9 + p_{11}\), and \(I(t_{12}) = p_2 + p_7\), \(O(t_{12}) = p_1 + p_3 + p_6 + p_{11}\), respectively. Please notice that, when the two sets of control transition are put into the original PNM of Figure 7, all 20 reachable markings (also including the 5 illegal markings) are live (please refer to Figures 11 and 12).

The second example shown in Figure 13 is from Huang et al.\(^{18}\) and Zhang and Uzam.\(^{19}\) Huang et al.\(^{18}\) identify 32 reachable markings including 4 deadlock markings (e.g. \(2p_0 + p_5 + p_6 + p_8 + p_9\), \(p_0 + p_2 + p_6 + p_8 + p_9\), \(p_2 + p_3 + p_8 + p_9\), and \(p_0 + p_3 + p_5 + p_8 + p_9\)). Finally, they recover all reachable markings by their two control transitions \((I(t_i) = 2p_7 + p_{11} + p_{12}\) and \(O(t_i) = p_8 + p_{10}\). \(I(t_i) = p_7 + p_{10} + p_{11}\) and \(O(t_i) = p_8 + p_9\)).

Zhang and Uzam\(^{19}\) discover a set covering problem and obtain two control transitions to solve the deadlock problem of this example. They also use two same transition-based controllers to make the system deadlock-free. However, in example 2, in fact, according to our study in this article, there should be 34 reachable markings including 6 deadlock markings (shown in

\begin{table}[h]
\centering
\caption{The properties of initial and two deadlock markings based on Algorithm 2.}
\begin{tabular}{|c|c|c|}
\hline
Marking no. & Classification & Information of marking \\
\hline
\(M_0\) & New initial marking & \(M_0 = p_1 + p_3 + p_6 + p_9 + p_{11}\) \\
\(M_{D_1}\) & First deadlock marking & \(M_{D_1} = p_3 + p_5 + p_{10}\) \\
\(M_{D_2}\) & First deadlock marking & \(M_{D_2} = p_2 + p_7 + p_9\) \\
\hline
\end{tabular}
\end{table}
controllers are added into example 2. TINA. Due to the limitation of paper space, we just show the property of the 6 deadlock markings.

Finally, two controllers \( I(t_{13}) = p_8 + p_9 \), \( O(t_{13}) = 2p_7 + p_{11} + p_{12} \), and \( I(t_{14}) = p_1 + p_2 + p_3 \), \( O(t_{14}) = 3p_0 + p_5 + p_6 + p_{11} + p_{12} \) are hence obtained under the proposed Deadlock Prevention Policy II, and all 34 reachable markings are recovered when the two controllers are added into example 2.

### Table 6. The properties of six deadlock markings checked by TINA in example 2.

<table>
<thead>
<tr>
<th>Marking no.</th>
<th>Classification</th>
<th>Information of marking</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M_0 )</td>
<td>Initial marking</td>
<td>( M_0 = 3p_0 + p_5 + p_6 + 2p_7 + p_{11} + p_{12} )</td>
</tr>
<tr>
<td>( M_5 )</td>
<td>Deadlock marking</td>
<td>( M_5 = p_1 + p_2 + p_3 + 2p_7 + p_{11} )</td>
</tr>
<tr>
<td>( M_{14} )</td>
<td>Deadlock marking</td>
<td>( M_{14} = p_1 + p_2 + p_3 + p_7 + p_9 )</td>
</tr>
<tr>
<td>( M_{22} )</td>
<td>Deadlock marking</td>
<td>( M_{22} = 3p_0 + p_5 + p_6 + p_8 + p_9 )</td>
</tr>
<tr>
<td>( M_{24} )</td>
<td>Deadlock marking</td>
<td>( M_{24} = 2p_0 + p_2 + p_6 + p_8 + p_9 )</td>
</tr>
<tr>
<td>( M_{26} )</td>
<td>Deadlock marking</td>
<td>( M_{26} = p_0 + p_2 + p_3 + p_8 + p_9 )</td>
</tr>
<tr>
<td>( M_{28} )</td>
<td>Deadlock marking</td>
<td>( M_{28} = 2p_0 + p_3 + p_5 + p_8 + p_9 )</td>
</tr>
</tbody>
</table>

Table 6) when the idle places \((p_0, p_7)\) are equal to (3, 2). Furthermore, saturated number of tokens in idle places for example 2 should be (3, 2). Please note that the six deadlock markings are checked by famous software TINA.\(^\text{30}\) Due to the limitation of paper space, we just show the property of the 6 deadlock markings.

### Conclusion

In the existing literature, almost all researches adopt place-based deadlock prevention method to solve the FMS deadlock problem. However, they cannot obtain real maximally permissive controllers since some bad and deadlock markings could not be recovered in the process of controlling FMS. Based on above reason, this research presents one novel deadlock recovery policy using transition-based technology. The proposed policy is obvious better than all existing literatures even all publishing transition-based deadlock prevention policies. Moreover, the proposed policy not only obtains maximally permissive controllers but also recovers all original deadlock markings. Preciously, we can make all markings alive whatever they initially belong to deadlock or illegal ones. In other words, this work is the innovation in deadlock prevention domain. In future works, we will use the proposed policy to test all kinds of FMS or a larger PNM.

### Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

### Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

---

**ORCID iD**

Yen-Liang Pan [https://orcid.org/0000-0002-4111-9564](https://orcid.org/0000-0002-4111-9564)

### References


30. 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