IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received July 24, 2021, accepted August 2, 2021, date of publication August 5, 2021, date of current version August 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3102855

An Optimal Deadlock Recovery Algorithm for Special and Complex Flexible Manufacturing Systems-S^{4}PR

TER-CHAN ROW[1] **AND YEN-LIANG PAN**[2]**, (Member, IEEE)**
[1]Department of Electronic Engineering, Army Academy, Taoyuan 32092, Taiwan
[2]Mathematics and Physics Division, General Education Center, Air Force Academy, Kaohsiung 82047, Taiwan
Corresponding author: Yen-Liang Pan (peterpan960326@gmail.com)

ABSTRACT Eliminating deadlocks and reserving maximal permissiveness together is an important and hot issue for all kinds of flexible manufacturing systems. Many experts are making much effort to obtain optimal control algorithms. However, most researchers cannot obtain the real optimal controllers even though their policy is maximally permissive, especially for special S^{4}PR (Systems of Sequential Systems with Shared Resources) systems. Based on this reason, this paper tries to propose a novel recovery policy to recover systems' all states based on control transitions (CT). Therefore, the system can hold the real maximal permissive states. Especially, we propose the concept of the shortest path selected marking (SPSM) so that we can hence obtain the recovery transition. Furthermore, our proposed concept avoids solving integer linear programming problems (ILPP) based on our algorithm method. Moreover, three examples of S^{4}PR Petri net models (PNM) with deadlocks are illustrated in our proposed algorithm. Experimental data shows that our policy can not only prevent the deadlocks but reserve all initial markings in S^{4}PR. Please notice that this proposed recovery policy seems the first one applied for S^{4}PR by using control transitions and still obtaining the best permissiveness whatever their algorithms among existing literature belong to use control places or control transitions.

INDEX TERMS Flexible manufacturing systems, Petri net, deadlock recovery, control transition.

I. INTRODUCTION

Nowadays is an era of digitization, mechanization, diversification, and artificial intelligence. The flexible manufacturing systems (FMS) are more suitable for this era working on factory production. Fortunately, the Petri net (PN) has been recognized as the most potent semantic-based tools for modeling discrete events and dynamic systems. Moreover, PN can search deadlocks prediction when one constructs an FMS. Therefore, how to avoid, prevent, or detect recovery deadlocks methods are presented.

Generally, two analysis methods are mainly used to formulate FMS deadlock prevention strategies: system structure analysis [3]–[6], [9], [19], [21]–[24], [30], [34], [35] and reachability graphs analysis [7], [8], [20], [39]–[43], [46]–[48]. The former relies on structural analysis, such as the concept of siphon control, to design a suitable control subnet to solve the deadlock problem. With this strategy, the calculations are usually simple, but will encounter explosion problems, because as the model becomes larger and more complex, the number of siphons increases exponentially. Li and Zhou [22]–[24] first presented searching the elementary siphon method. Their motivation is to control explicitly and reduce the dependent siphons. Then, the effect of elementary siphon control is more than siphon control to prevent the deadlocks.

The latter can be adopted to design the maximally permissive controllers. Under this analysis method, a system's reachability graph is needed to compute firstly. Besides, all reachable markings in a reachability graph are also needed to identify. Accordingly, Uzam [7] firstly proposed the novel concepts of Deadlock Zone (DZ) and Deadlock-Free Zone (DFZ) based on the theory of regions [25] to design an optimal deadlock prevention policy. On the other hand, Ghaffari *et al.* [39] also redefine the theory of regions and introduce three new definitions of dangerous

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

markings, forbidden markings, legal markings, and the set of Marking/Transition-Separation Instances (MTSI). For enhancing computational efficiency of the conventional theory of regions, Huang and Pan [40], [41], Huang *et al.* [42], and Pan *et al.* [43] even put forward the concept of crucial MTSI (CMTSI). It is worth noting that these adopted CMTSI deadlock prevention policies still make a controlled system maximally permissive.

In addition, many experts have invested some new concepts in the field of deadlock prevention for solving the FMS's deadlock problem [1], [2], [44]–[48], [50]–[52]. For example, Chen *et al.* [1], [2] use interval inhibitor arcs to forbid reaching illegal markings. Pirrodi *et al.* [44] utilize siphons and markings to design maximally permissive controllers. Pan *et al.* [45] combine the theory of regions and selective siphons to obtain more efficient controllers. Chen *et al.* [46] proposed an MFFP (i.e., the Maximal number of Forbidding First bad marking (FBM) Problem) concept to obtain maximally permissive controllers, etc. However, above all mentioned algorithms could inevitably lose parts of original reachable markings even they are maximally permissive since they cannot recover deadlock and quasi-deadlock markings. They merely prevent legal markings from running to deadlock or quasi-deadlock zone.

Thus, Huang *et al.* [10] were the first to present the idea that prevented deadlocks and kept original markings by using control transitions. Simultaneously, Huang *et al.* [11] constructed the algorithm roughly to prevent deadlocks from occurring and reserve original reachable markings successfully. Yet, they cannot avoid the redundant control transitions. In [12], Huang *et al.* proposed a deadlock detection and recovery method to reduce redundant control transitions. However, its controllers could not become optimal in a larger or more complicated system due to its manual designing process. In other words, in [12], it does not show any formalized equation in its algorithm.

Motivated by the work in [12], Zhang and Uzam [13] develop a set covering technique to derive the minimal control set from enhancing computational efficiency. Finally, a deadlock S^3PR PNM can be recovered successfully by adding the minimal control transitions. However, it is a pity that they merely demo two small S^3PR PNM from [12], although according to their conclusion, the proposed method can be applied in pure and bounded Petri net models of FMSs.

Based on transition-based techniques, Row and Pan [15] also present two novel deadlock prevention algorithms, the All Reachability Graph (ARG) viewpoint and the First Deadlock Marking (FDM) viewpoint, to obtain maximally permissive live states and recover all original deadlock markings for an original PN model.

Similarly, Row *et al.* [17] present a deadlock recovery policy using control transitions. Its method can detect the maximum reachable markings under the same system structure before the recovery algorithm is used. Besides, it can use fewer controllers among existing literature [12], [15] to recover all deadlock and quasi-deadlock markings in a controlled system. The first one uses transition-based controllers to successfully recover the large classical S^3PR PNM with 26750 reachable markings [3].

At the same time, Bashir *et al.* [16] also used iterative procedures to design deadlock recovery based on control transitions. First, to prevent the state explosion problem in the reachability graph analysis, the structural characteristic of PNM is considered. Then, calculate the control transition and add it to the deadlock PNM to recover the deadlock markings. Nevertheless, this method is still not the most ideal because it cannot fully recover PNM.

On the other hand, Chen *et al.* [14] proposed another novel transition-based recovery policy and developed two different kinds of algorithms. The first one uses an iterative approach. In each iteration step, the integer linear programming problem (ILPP) is calculated to obtain a control transition until all deadlock states are covered. The second is a one-time skill. Therefore, one can find and pick up all controllers. According to their experimental results, these two algorithms have indeed successfully restored two classic S^3PR examples. However, according to [26], it has a computational complexity problem because there are too many constraints and variables in the formulated ILPP. Therefore, Dong *et al.* [26] proposed an improved deadlock recovery method to solve the FMS deadlock problem based on vector analysis. In addition, iteratively calculates a small number of recovery controllers to recover all illegal markings. Therefore, it does show better computational efficiency than [14]. However, this method is still NP-hard because it must solve ILPP.

Recently, Pan [18] proposed a deadlock recovery strategy based on the concept of a three-dimensional matrix called Generation and Comparison Auxiliary Matrix (GCAM). The recovery controllers of a deadlock PNM can hence be obtained once its GCAM is built. Finally, an iterative method is used until all deadlock and quasi-deadlock markings become legal markings. According to the experimental results [18], the GCAM method does obtain optimal with the least number of controllers. However, its computational cost will rise sharply in a large PNM since it has to build a three-dimensional matrix between every deadlock marking with all legal markings.

The above all deadlock recovery methods by using control transitions focus on classic S^3PR PNM, although we all know that all transition-based methods are not limited to S^3PR. In fact, in the existing literature, we have not seen any research paper using control transition in S^4PR PNM. Thus, in this paper, we first try to solve deadlocks for S^4PR PNM by using control transitions.

The rest of this article is organized as follows. The second section describes the basic definition of Petri nets and reachability graph analysis. The third section proposes the proposed control method. The fourth section introduces the experimental results. The fifth section gives a comparison. Finally, conclude Section VI.

II. PRELIMINARY

A. PETRI NETS [29]

A Petri Net (PN) is a five-tuple $N = (P, T, F, W, M_0)$. P and T are two finite, nonempty, and disjoint sets. P represents a set of places, and T represents a set of transitions. The relation between P and T is that $(P \cup T \neq \emptyset)$ and $(P \cap T = \emptyset)$. The relation F, P, and T is $F \subseteq \{(P \times T) \cup (T \times P)$. It represents the set of arcs of a PN, symbolized by arrows going from P to T or T to P denotes a flow relation in the PN model. $W\colon (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ for each $(x, y) \in F$, where $x, y \in (P \cup T)$ and $\mathbb{N}$ is the set of non-negative integers. ${}^\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ is called the preset of x, and $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$ is called the postset of x. A marking M is a multiset of PN places, which allocates tokens to each place of a PN model. $M(p)$ denotes the number of tokens in place p. M_0 represents the initial marking of N. A PN is pure if no place is both input and output places of the same transition. The incidence matrix $[N]$ of the pure net N is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$.

A transition $t \in T$ is enabled at marking M if $\forall p \in {}^\bullet t, M(p) \geq W(p, t)$, which is denoted as $M[t\rangle$. Once an enabled transition t fires, it generates a new marking M', denoted as $M[t\rangle M'$, where $M'(p) = M(p) - W(p, t) + W(t,p)$. $M[\rangle$ represents the set of all markings reachable from M by firing any possible sequence of transitions. $M_0[\rangle$ is the set of reachable markings of net N with initial marking M_0, often denoted as $R(N, M_0)$. It can be graphically shown by a reachability graph, which can, in turn, be denoted as $G(N,M_0)$. It is a directed graph in which each node represents a marking in $R(N,M_0)$, and arcs are labeled by the fired transitions. Let (N, M_0) be a net system with $N = (P, T,F, W)$. A transition $t \in T$ is live at M_0if $\forall M \in R(N, M_0), \exists M' \in R(N, M_0), M'[t\rangle$. (N, M_0) is live if $\forall t \in T$, t is live at M_0. It is dead at M_0if $t \in T, M_0[t\rangle$.

B. PETRI NET MODEL

Definition 1: A Simple Sequential Process (S^{2}P) is a Petri net $N = (P_A \cup \{p^0\}, T, F)$, where: (1) $P_A \neq \emptyset$ ($p \in P_A$ is called an operation place),$p^0 \notin P_A$,(p^0 is called a process idle place); (2) N is a strongly connected state machine; and (3) every circuit of N contains place p^0.

Definition 2: A Simple Sequential Process with Resources (S^{2}PR) [3] is a Petri net $N = (P_A \cup \{p^0\}, T,P_R, F)$, where: (1) $P_R \neq \emptyset$, $\left(\mathrm{P} \cup \left\{p^0\right\}\right) \cap P_R = \emptyset$; (2) $\forall p{\in}P, \forall t \in {}^\bullet p, t' \in p^\bullet, {}^\bullet t \cap P_R = t'^\bullet \cap P_R = \{r_p$; (3) The following two statements are varied: i) $\forall r \in P_R. {}^{\bullet\bullet}r \cap P = r^{\bullet\bullet} \cap P \neq \emptyset$; ii) $\forall r \in P_R. {}^\bullet r \cap r^\bullet = \emptyset$. (4) ${}^{\bullet\bullet}\left(p^0\right) \cap P_R = \left(p^0\right)^{\bullet\bullet} \cap P_R = \emptyset$.

Definition 3: A System of Simple Sequential Process with Resources (S^{3}PR) [3] is defined as the union of a set of nets N_i sharing common places, called resource places P_{Ri}, such that $N_i = (P_i \cup \{p_i^0 \cup P_{Ri}, T_i, F_i)$, where (1)$\notin P_i \neq \emptyset, p_i^0 P_i$ (2)N_i' is a strongly connected state machine where $N_i' = (P_i \cup \left\{p_i^0\right\}, T_i, F_i')$ is the resultant net after P_{Ri} is removed from N_i; (3) every circuit of N_i' contains the place p_i^0 and (4) any two N_i' are composable when they share a set of common resource idle places. In the above definition, p_i^0 is called the process idle place and P_i the set of operation places. The N_i' shall be called a process net.

Definition 4: A Systems of Sequential Systems with Shared Resources (S^{4}PR) [36], [38]:

A well-marked S^{4}PR is a generalized connected self-loop-free Petri net $N = (P^0 \cup P_A \cup P_R, T, F, W)$ defined as the union of a set of nets $N_i = (\left\{p_i^0\right\} \cup P_{Ai} \cup P_{Ri}, T_i, F_i, W_i)$, $i \in \mathbb{N}_m = \{1, 2, \ldots, m\}$, where:

1) $P^0 = \cup_{i\in\mathbb{N}_m} \left\{p_i^0\right\}, P_A = \cup_{i\in\mathbb{N}_m} P_{Ai}, P_R = \cup_{i\in\mathbb{N}_m} P_{Ri}, T = \cup_{i\in\mathbb{N}_m} T_i, F = \cup_{i\in\mathbb{N}_m} F_i, W = \cup_{i\in\mathbb{N}_m} W_i$. $\forall i \neq j \in \mathbb{N}, P_{Ai} \cap P_{Aj} = \emptyset, T_i \cap T_j = \emptyset$.
2) P^0, P_A, P_R are called the process idle places, operation places and resource places respectively. T is called the set of transitions.
3) $P_R \neq \emptyset, P_A \neq \emptyset, P^0 \in /P_A, (P_A \cup \{p^0) \cap P_R = \emptyset$.
4) $\forall i \in \mathrm{N}_m$, the subset N_i generated by $P_A \cup \{p_0\} \cup T_i$ is a strongly connected state machine such that every circuit of the state machine contains idle place p_0.
5) $\forall r \in P_R$, there exists a unique minimal P-semiflow Y_r such that$\{\mathrm{r}\} = \|Y_r\| \cap P_R, P^0 \cap \|Y_r\| = \emptyset, P_A \cap \|Y_r\| \neq \emptyset$ and $Y_r(r) = 1$. Moreover, $P_A = (\bigcup_{r\in P_R} \|Y_r\|) \backslash P_R$.
6) $\forall p \in P_A, M_0(p) = 0; \forall r \in P_R, M_0(r) \geq max_{p\in\|Y_r\|} Y_r(p)$, and $\forall p_j^0 \in P^0, M_{0\left(p_j^0\right)} \geq 1$.

C. REACHABILITY GRAPH (RG) ANALYSIS

In this paper, the proposed algorithm needs to analyze the reachability graph and identify all reachable markings. The detailed information of these definitions is shown as follows:

Definition 5: Deadlock marking $M_D = \{M \in R(N, M_0) |$ all transition $t \in T$ is not enabled at marking $M\}$. $\mathcal{M}_D$ represents the set of deadlock markings.

Definition 6: Quasi-deadlock marking $M_Q = \{M \in R(N, M_0) |$ one regardless of transition firing sequences, M must eventually evolve to a deadlock marking $\}$. $\mathcal{M}_Q$ represents the set of quasi-deadlock markings.

Deadlock markings and Quasi-deadlock markings are collectively called illegal markings. The illegal marking is defined as follows:

Definition 7: Illegalmarking$M_I = \left\{M \in R(N, M_0) \wedge M \in (M_D \cup M_Q)\right\}$. $\mathcal{M}_I$ represents the set of illegal markings.

Except for quasi-deadlock and deadlock markings, the other reachable markings are legal ones.

Definition 8: $M_L = \left\{M \in R(N, M_0) \wedge (M = R(N,M_0) - M_D - M_Q)\right\}$. $\mathcal{M}_L$ represents the set of legal markings.

Besides,

Definition 9: An area composed of all deadlock markings is called the deadlock zone, denoted by Z_D.

Definition 10: An area composed of all quasi-deadlock markings is called the quasi-deadlock zone, denoted by Z_Q.

Definition 11: An area composed of all quasi-deadlock and deadlock markings is called the illegal zone, denoted by Z_I.

Definition 12: An area composed of all legal markings is called a legal zone, denoted by Z_L.

III. CONTROL METHODOLOGY

This section wants to further propose a novel control methodology with Petri nets and reachability graph analysis. Note that the deadlock markings and legal markings are used to calculate control transition (CT).

A. THE DESIGN CONCEPT OF CONTROL TRANSITION (CT)

According to above mention Petri nets theory and reachability graph analysis. An enabled transition undoubtedly can change one marking to another one. Therefore, if we can find an enabled transition to make a deadlock marking a legal one, the enabled transition could be used to design control transition (CT). Note that it will form a livelock [49] situation if we use an enabled transition between M_D and M_Q as control transition (CT). In detail, a livelock situation is similar to a deadlock, except that the states of the processes involved in the livelock constantly change to one another, none progressing. The term was coined firstly by Edward A. Ashcroft in a 1975 paper [49]. The design concept diagram of control transition (CT) is shown in FIGURE 1. FIGURE 2 is a typical S^{4}PR PNM [27], [32], and its reachable graph is shown in FIGURE 3.

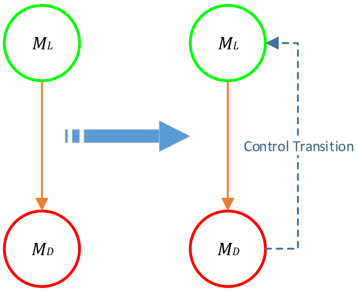

FIGURE 1. The design concept diagram of control transition (CT).

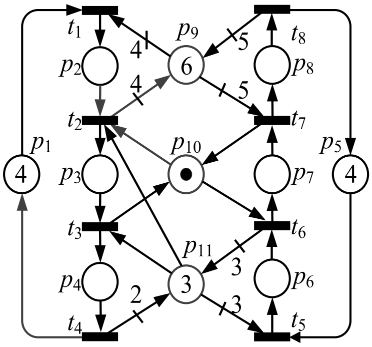

FIGURE 2. S^{4}PR from [27], [32].

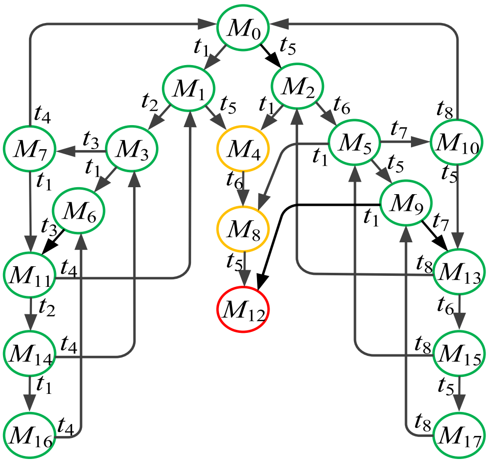

FIGURE 3. The reachability graph of FIGURE 2.

TABLE 1. The detailed information of markings in Example 1.

Marking No.	The Property of Marking	Attribute
M_0	$4p_1 + 4p_5 + 6p_9 + p_{10} + 3p_{11}$	M_L
M_1	$3p_1 + p_2 + 4p_5 + 2p_9 + p_{10} + 3p_{11}$	M_L
M_2	$4p_1 + 3p_5 + p_6 + 6p_9 + p_{10}$	M_L
M_3	$3p_1 + p_3 + 4p_5 + 6p_9 + 2p_{11}$	M_L
M_4	$3p_1 + p_2 + 3p_5 + p_6 + 2p_9 + p_{10}$	M_Q
M_5	$4p_1 + 3p_5 + p_7 + 6p_9 + 3p_{11}$	M_L
M_6	$2p_1 + p_2 + p_3 + 4p_5 + 2p_9 + 2p_{11}$	M_L
M_7	$3p_1 + p_4 + 4p_5 + 6p_9 + p_{10} + p_{11}$	M_L
M_8	$3p_1 + p_2 + 3p_5 + p_7 + 2p_9 + 3p_{11}$	M_Q
M_9	$4p_1 + 2p_5 + p_6 + p_7 + 6p_9$	M_L
M_{10}	$4p_1 + 3p_5 + p_8 + p_9 + p_{10} + 3p_{11}$	M_L
M_{11}	$2p_1 + p_2 + p_4 + 4p_5 + 2p_9 + p_{10} + p_{11}$	M_L
M_{12}	$3p_1 + p_2 + 2p_5 + p_6 + p_7 + 2p_9$	M_D
M_{13}	$4p_1 + 2p_5 + p_6 + p_8 + p_9 + p_{10}$	M_L
M_{14}	$2p_1 + p_3 + p_4 + 4p_5 + 6p_9$	M_L
M_{15}	$4p_1 + 2p_5 + p_7 + p_8 + p_9 + 3p_{11}$	M_L
M_{16}	$p_1 + p_2 + p_3 + p_4 + 4p_5 + 2p_9$	M_L
M_{17}	$4p_1 + p_5 + p_6 + p_7 + p_8 + p_9$	M_L

From reachability graph analysis using TINA [31], one can know that Marking 12 (M_{12}) belongs to deadlock marking (M_D). Marking 4 (M_4) and Marking 8 (M_8) belong to quasi-deadlock marking (M_Q). Marking 0 (M_0), Marking 1 (M_1), Marking 2 (M_2), Marking 3 (M_3), Marking 5 (M_5), Marking 6 (M_6), Marking 7 (M_7), Marking 9 (M_9), Marking 10 (M_{10}), Marking 11 (M_{11}), Marking 13 (M_{13}), Marking 14 (M_{14}), Marking 15 (M_{15}), Marking 16 (M_{16}), and Marking 17 (M_{17}) belong to legal marking (M_L), respectively. To facilitate tracking, the detailed information of all reachable markings is shown in TABLE 1 based on TINA.

According to [14], [18], for anyone deadlock marking, one can choose all legal markings to form a control transition (CT) once the M_D and M_L are identified. In other words, in this example, $M_0 \sim M_3$, $M_5 \sim M_7$, $M_9 \sim M_{11}$, and $M_{13} \sim M_{17}$ could be chosen to obtain a control transition (CT). To easily understand our design concept of control transition, we try to pick the legal marking M_9 and the deadlock marking M_{12} as an illustration. Firstly, based on Petri nets' state equation, the legal marking M_9 can be fired to the deadlock marking M_{12} through the single transition t_1. In this case, the state equation in Equation (1) is as below:

$$\begin{bmatrix}4\\0\\0\\0\\2\\1\\1\\0\\6\\0\\0\end{bmatrix}+\begin{bmatrix}-1&1&0&0&0&0&0&0&-4&0&0\\0&-1&1&0&0&0&0&0&4&-1&-1\\0&0&-1&1&0&0&0&0&0&1&-1\\1&0&0&-1&0&0&0&0&0&0&2\\0&0&0&0&-1&1&0&0&0&0&-3\\0&0&0&0&0&-1&1&0&0&-1&3\\0&0&0&0&0&0&-1&1&-5&1&0\\0&0&0&0&1&0&0&-1&5&0&0\end{bmatrix}^T \cdot \begin{bmatrix}1\\0\\0\\0\\0\\0\\0\\0\end{bmatrix}=\begin{bmatrix}3\\1\\0\\0\\2\\1\\1\\0\\2\\0\\0\end{bmatrix} \quad (1)$$

In other words, if we want to design a control transition (CT) denoted t_c in this deadlock PNM and push the deadlock marking M_{12} back the legal marking M_9, the state equation can be rewritten in Equation (2) as follows:

$$\begin{bmatrix}3\\1\\0\\0\\2\\1\\1\\0\\2\\0\\0\end{bmatrix}+\begin{bmatrix}-1&1&0&0&0&0&0&0&-4&0&0\\0&-1&1&0&0&0&0&0&4&-1&-1\\0&0&-1&1&0&0&0&0&0&1&-1\\1&0&0&-1&0&0&0&0&0&0&2\\0&0&0&0&-1&1&0&0&0&0&-3\\0&0&0&0&0&-1&1&0&0&-1&3\\0&0&0&0&0&0&-1&1&-5&1&0\\0&0&0&0&1&0&0&-1&5&0&0\\a&b&c&d&e&f&g&h&i&j&k\end{bmatrix}^T \cdot \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\\1\end{bmatrix}\begin{matrix}\\\\\\\\\\\\\\\\\leftarrow t_c\end{matrix}=\begin{bmatrix}4\\0\\0\\0\\2\\1\\1\\0\\6\\0\\0\end{bmatrix} \quad (2)$$

Note that in Equation (2), the unknown *a, b, c, d, e, f, g, h, i, j*, and *k* belong to the integer since they are the elements of the incidence matrix of FIGURE 2. Besides, we move M_{12} to the right side of the state equation. Equation (2) can be present in Equation (3) form as below:

$$\begin{bmatrix}a\\b\\c\\d\\e\\f\\g\\h\\i\\j\\k\end{bmatrix}=\begin{bmatrix}4\\0\\0\\0\\2\\1\\1\\0\\6\\0\\0\end{bmatrix}-\begin{bmatrix}3\\1\\0\\0\\2\\1\\1\\0\\2\\0\\0\end{bmatrix} \quad (3)$$

Accordingly, we can obtain that $a = 1, b = -1, i = 4$, and $c = d = e = f = g = h = j = k = 0$ based on the legal marking M_9 − the deadlock marking M_{12}. The designed control transition (CT) in this case can then be identified shown in Equation (4):

$$\begin{aligned}CT = [t_c]_{11\times 1} &= \begin{bmatrix}1\\-1\\0\\0\\0\\0\\1\\0\\4\\0\\0\end{bmatrix}_{11\times 1}\\ &= [M_9]_{11\times 1} - [M_{12}]_{11\times 1}\end{aligned} \quad (4)$$

In this paper, we adopt CT $= t_c = M_9 - M_{12}$ instead of $[t_c] = [M_9] - [M_{12}]$. Further, in general case, we can obtain the control transition (CT) from $t_c = M_L - M_D$.

Definition 13: Based on $M_L - M_D$, a designed transition t_c which can lead a deadlock marking M_D to a legal marking M_L is called the control transition (CT).

According to Definition 13, there are usually many legal markings in a deadlock PNM. Therefore, identifying the crucial legal marking so that it can be used to design control transition (CT) is a worth issue. In the following, we will give the formal definitions to identify crucial legal markings from the set of legal markings $\mathcal{M}_L$, called the Shortest Path Selected Marking (SPSM). The deadlock S^{4}PR PNM could hence be controlled with a simpler algorithm and better computational efficiency. The SPSM is defined as follows:

B. THE SHORTEST PATH SELECTED MARKING (SPSM)

There are two kinds of types of SPSM are defined as below:

Definition 14: Type I SPSM: $M'_{SPSM} = \{M \in M_L, t \in T$, and $\exists M' \in M_D, M'' \in M_L$, and $t' \in T$ such that $M\ [t > M'$ and $M\ [t' > M''\}$. Denote this kind of the deadlock marking related to M'_{SPSM} as M'_D called Type I deadlock marking.

Definition 14 explains Type I SPSM that can be led into a Z_D and Z_L as shown in FIGURE 4 through a single transition's

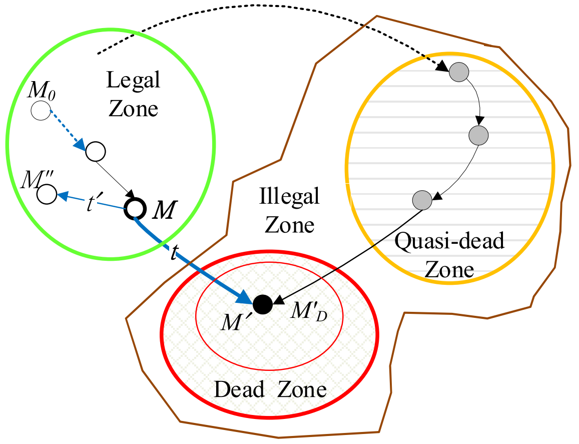

FIGURE 4. A diagram of Type I M'_{SPSM} in a reachability graph.

firing. For those markings that are not Type I SPSM, we need to introduce Type II SPSM.

Definition 15: σ_{k} is defined as a firing sequence starting in a quasi-deadlock marking (M_Q) and ending in a deadlock marking (M_D) where $k = |\sigma_k|$ is the number of transitions in σ_k, called its length. Denote a firing sequence with the shortest length (i.e., smallest k) from any quasi-deadlock marking $M_Q$to deadlock marking M_D as $\sigma^*(M_D)$.

Definition 16: Type II SPSM: $M_{SPSM}{}'' = \{M \in M_L, t \in T$, and $\exists M' \in M_Q, M'' \in M_L, M''' \in M_D, t' \in T$, and a firing sequence from M' to M''' such that $M\ [t > M', M\ [t' > M'', M'\ [\sigma^* > M'''\}$. The deadlock marking associated with a Type II SPSM $M''_{SPSM}$is denoted as M''_D, called Type II deadlock marking. A diagram of Type II M''_{SPSM} in a reachability graph is shown in FIGURE 5.

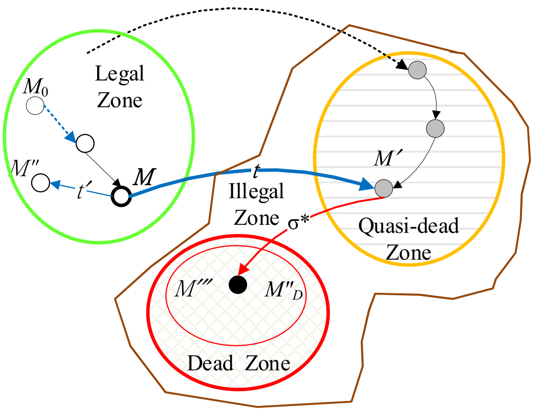

FIGURE 5. A diagram of Type II M''_{SPSM} in a reachability graph.

For further understanding the detailed information of seeking Type II SPSM. FIGURE 6 shows that the shorter path σ^* is what we want since $|\sigma^*| < |\sigma'|$ (i.e. $|\sigma^*| = 1$ and $|\sigma'| = 3$).

Remark 1: A deadlock marking must be with its corresponding SPSM. According to above definitions, the corresponding SPSM is of either Type I SPSM (M'_{SPSM}) or II (M''_{SPSM}).

Remark 2: Note that Type I SPSM (M'_{SPSM}) should be identified and processed firstly for a deadlock marking.

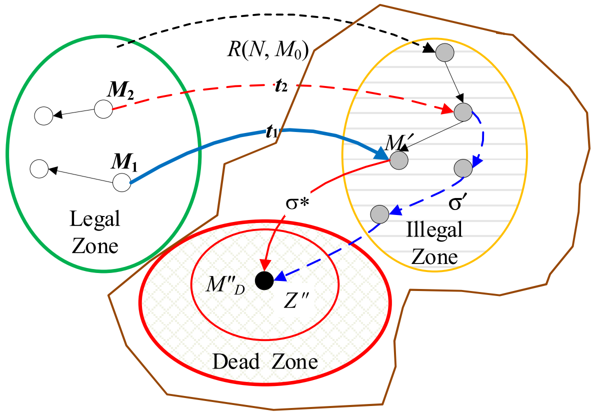

FIGURE 6. The diagram of a shorter path σ^* for seeking Type II M''_{SPSM}.

Theorem 1: A deadlock marking must be covered with its corresponding SPSM.

Proof: GA designed transition t_c which can lead a deadlock marking M_D to a legal marking M_L is called the control transition (CT) if $t_c = M_L - M_D$. A deadlock marking $M_D$must be covered with its corresponding SPSM since (M'_{SPSM} and $M''_{SPSM}) \in \mathcal{M}_L$. ◇

Recall FIGURE 3. The deadlock marking M_{12} belongs to the Type I deadlock marking according to the above definitions. And, M_9 is its corresponding M'_{SPSM}.

C. PROCEDURE OF DEADLOCK PREVENTION POLICY

Definition 17: Let function $f(t_c)$ denote the recovery ability of a control transition, which is used to identify the most effective CT. The values of$f(t_c)$ are possible zero or positive integer and initially reset as zero before comparing the comparison from t_c to others.

Theorem 2: The proposed Algorithm can recover all deadlock markings for a deadlock S^{4}PR PNM.

Proof: According to Theorem 1, a deadlock marking must be covered with its corresponding SPSM if $\mathcal{M}_D \neq \emptyset$. Therefore, the obtained all controllers t_c based on $M'_{SPSM} \cup M''_{SPSM}$ can recover their corresponding deadlock markings. Then, we use loop and iteration to find out the optimal controllers which can recover the most deadlock markings. The process cannot terminate until all deadlock markings in $\mathcal{M}_D$ are recovered. Finally, we can obtain a set of recovery controllers to recover all deadlock markings. The conclusion holds. ◇

We discuss the computational complexity of the proposed algorithm. Firstly, it requires computing the reachability graph of a PNM. Undoubtedly, it suffers from the state's explosion problem. The reachability graph will be calculated at each iteration due to the $\mathcal{M}_L$ and $\mathcal{M}_D$ change after adding the controller to the PNM. Under the proposed policy, generally, $\mathcal{M}_L$ extends, and $\mathcal{M}_D$ reduces after every iteration. Secondly, this proposed algorithm is not an NP-hard problem since it does not to solve the Linear Programming Problems (LPPs). Although repeating analysis to generate a reachability graph makes more computation heavy, designing

Algorithm 1 An Optimal Deadlock Recovery Algorithm for S4PR

Input: A deadlock S^{4}PR PNM model with deadlock.
Output: A live S^{4}PR PNM without any deadlock.

```
1. Generate $R(N, M_0)$ with all reachable markings.
2. Identify all $M_D$, $M_Q$, and $M_L$.
3. Find $M'_D$.
   If all deadlock markings belong to $M'_D$ goes to Step 6.
4. Identify $M'_{SPSM}$.
5. Find $M''_D$.
6. Identify $M''_{SPSM}$.
7. Calculate all control transitions      /*($t_c$).
8. Comparing computation.
   while ($\mathcal{M}_D \neq \emptyset$) do
      for ($i \in \mathbb{N}_m^+ = \{1, 2, \ldots, m\}, \forall t_i \in t_c$) do
      /*$m$ = the number of control transitions
         $f(t_i) \leftarrow 0$
       for ($j \in N_m^+ = \{i, i+1, i+2, \ldots, m\}, \forall t_j \in t_c$) do
         if ($t_i$ = t$_i$ = t$_j$) then
         $f(t_i) \leftarrow f(t_i) + 1$
         end if
       end for
      end for
     Find out the maximal value of $f(t_i)$
     Add the control transition $t_i$ into $R(N, M_0)$
     Run reachability graph analysis
    end while
 9. Output completely recovered system with $\mathcal{M}_D = \emptyset$.
```

supervisors is still much more time-saving and systematic than other methods, such as Chen *et al.* [14] and Pan [18], since they both need to check all legal markings for calculating every deadlock marking.

IV. EXAMPLES

There are three examples used to illustrate our proposed deadlock recovery algorithm.

A. EXAMPLE 1

The example 1 [27], [32] shown in FIGURE 2 is a simple S^{4}PR PNM with a deadlock marking, and the number of reachable markings of reachability graph is 18. To prevent deadlock, 18 reachable markings (M_0 to M_{17}) are identified according to the software TINA [31]. A deadlock marking M_{12}, two quasi-deadlock markings M_4 and M_8 are then located. According to the reachability graph analysis and the definition of SPSM, one can know that Marking 12 (M_{12}) belongs to Type I deadlock marking (M'_D) and Marking 9 (M_9) is its corresponding Type I SPSM.

Therefore, we can obtain the control transition from $M_9 - M_{12} = (4p_1 + 2p_5 + p_6 + p_7 + 6p_9) - (3p_1 + p_2 + 2p_5 + p_6 + p_7 + 2p_9) = p_1 - p_2 + 4p_9$.

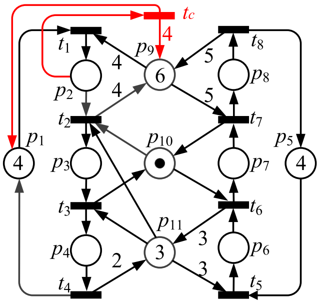

FIGURE 7. The controlled S^{4}PR PNM and its controller of Example 1.

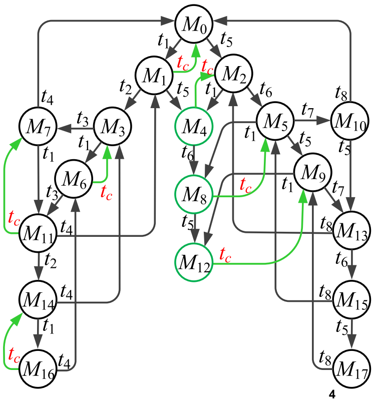

FIGURE 8. The reachability graph of the controlled S^{4}PR PNM of Example 1.

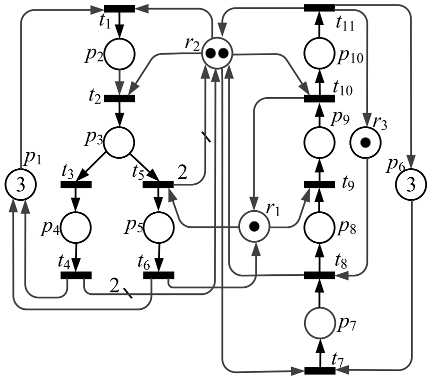

FIGURE 9. A more complex S^{4}PR PNM of Example 2.

Accordingly, the optimally controlled PNM is obtained with the obtained control transition t_c when added into the deadlock S^{4}PR PNM. FIGURES 7-8 show the controlled system and its reachability graph, respectively. From FIGURE 8, all deadlock and quasi-deadlock markings are controlled and recovered. Furthermore, all original 18 reachable markings are held totally. The controlled PNM is real maximally permissive.

B. EXAMPLE 2

A more complex model S^{4}PR PNM than Example 1 from [28, 33] shown in FIGURE 9 is discussed in this subsection. There are three resource places (r_1, r_2, and r_3), two idle

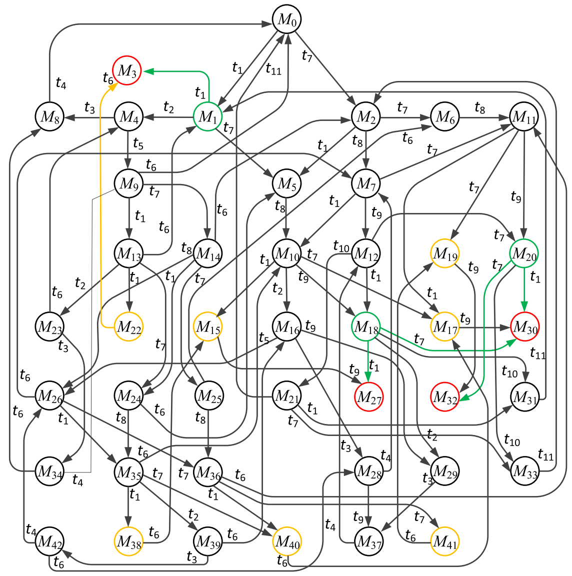

FIGURE 10. The reachability graph of FIGURE 9.

places (p_1, p_6), and eight activity places (p_2-p_5, p_7-p_{10}) in this model. Moreover, the number of its initial reachable markings is 43, including 4 deadlock markings (i.e., M_3, M_{27}, M_{30}, and M_{32}) and 7 quasi-deadlock markings (i.e., M_{15}, M_{17}, M_{19}, M_{38}, M_{40}, and M_{41}) based on the reachability analysis. Please note that we still use the software TINA to run reachability analysis for easy tracking and checking. The full reachability graph of Example 2 is present in FIGURE 10, and the detailed information of total 43 reachable markings are shown in TABLE 2, respectively.

According to the definitions of SPSM (also can refer to FIGURE 4), one can know that these four deadlock markings all belong to Type I deadlock marking (M'_D). Their corresponding Type I SPSMs are shown in TABLE 3. Note that the deadlock marking M_{30} has 2 Type I SPSMs since the two legal markings fit the definition 14. Accordingly, their control transition (CT) can then be calculated. TABLE 4 shows the five control transitions. Obviously, $t_{c1} = t_{c2} = t_{c4} = (p_1 + r_2) - (p_2)$ and $t_{c3} = t_{c5} = (p_1 + r_2) - (p_2)$ from TABLE 4. Therefore, we choose $t_{c1} = (p_1 + r_2) - (p_2)$ as the first controller to add into the PNM since it can control the three deadlock markings (i.e., M_3, M_{27}, and M_{30}). When we add the controller to the PNM, obviously, there is only one deadlock marking M_{32}left in the system. FIGURE 11 shows this situation. From the reachability graph in FIGURE 11, we can see that all 43 reachable markings are reserved in this PNM. In other words, the first control transition controls and recovers the initial three deadlock markings and their corresponding quasi-deadlock markings. Finally, the optimally controlled system net is obtained with the second control transition $t_{c5=} (p_1 + r_2) - (p_2)$ when it is added into the deadlock S^{4}PR PNM. FIGURE 12 shows the controlled reachability graph. From FIGURE 12, all initial illegal markings are recovered these two control transitions (CT) and become live.

TABLE 2. The detailed information of all reachable markings of Example 2.

Marking No.	Information of Markings	Attribute
M_0	$3p_1 + 3p_6 + r_1 + 2r_2 + r_3$	M_L
M_1	$2p_1 + p_2 + 3p_6 + r_1 + r_2 + r_3$	M_L
M_2	$3p_1 + 2p_6 + p_7 + r_1 + r_2 + r_3$	M_L
M_3	$p_1 + 2p_2 + 3p_6 + r_1 + r_3$	M_D
M_4	$2p_1 + p_3 + 3p_6 + r_1 + r_3$	M_L
M_5	$2p_1 + p_2 + 2p_6 + p_7 + r_1 + r_3$	M_L
M_6	$3p_1 + p_6 + 2p_7 + r_1 + r_3$	M_L
M_7	$3p_1+2p_6+p_8+r_1+2r_2$	M_L
M_8	$2p_1+p_4+3p_6+r_1+r_3$	M_L
M_9	$2p_1+p_5+3p_6+2r_2+r_3$	M_L
M_{10}	$2p_1+p_2+2p_6+p_8+r_1+r_2$	M_L
M_{11}	$3p_1+p_6+p_7+p_8+r_1+r_2$	M_L
M_{12}	$3p_1+2p_6+p_9+2r_2$	M_L
M_{13}	$p_1+p_2+p_5+3p_6+r_2+r_3$	M_L
M_{14}	$2p_1+p_5+2p_6+p_7+r_2+r_3$	M_L
M_{15}	$p_1+2p_2+2p_6+p_8+r_1$	M_Q
M_{16}	$2p_1+p_3+2p_6+p_8+r_1$	M_L
M_{17}	$2p_1+p_2+p_6+p_7+p_8+r_1$	M_Q
M_{18}	$2p_1+p_2+2p_6+p_9+r_2$	M_L
M_{19}	$3p_1+2p_7+p_8+r_1$	M_Q
M_{20}	$3p_1+p_6+p_7+p_9+r_2$	M_L
M_{21}	$3p_1+2p_6+p_{10}+r_1+r_2$	M_L
M_{22}	$2p_2+p_5+3p_6+r_3$	M_Q
M_{23}	$p_1+p_3+p_5+3p_6+r_3$	M_L
M_{24}	$p_1+p_2+p_5+2p_6+p_7+r_3$	M_L
M_{25}	$2p_1+p_5+p_6+2p_7+r_3$	M_L
M_{26}	$2p_1+p_5+2p_6+p_8+2r_2$	M_L
M_{27}	$p_1+2p_2+2p_6+p_9$	M_D
M_{28}	$2p_1+p_4+2p_6+p_8+r_1$	M_L
M_{29}	$2p_1+p_3+2p_6+p_9$	M_L
M_{30}	$2p_1+p_2+p_6+p_7+p_9$	M_D
M_{31}	$2p_1+p_2+2p_6+p_{10}+r_1$	M_L
M_{32}	$3p_1+2p_7+p_9$	M_D
M_{33}	$3p_1+p_6+p_7+p_{10}+r_1$	M_L
M_{34}	$p_1+p_4+p_5+3p_6+r_3$	M_L
M_{35}	$p_1+p_2+p_5+2p_6+p_8+r_2$	M_L
M_{36}	$2p_1+p_5+p_6+p_7+p_8+r_2$	M_L
M_{37}	$2p_1+p_4+2p_6+p_9$	M_L
M_{38}	$2p_2+p_5+2p_6+p_8$	M_Q
M_{39}	$p_1+p_3+p_5+2p_6+p_8$	M_L
M_{40}	$p_1+p_2+p_5+p_6+p_7+p_8$	M_Q
M_{41}	$2p_1+p_5+2p_7+p_8$	M_Q
M_{42}	$p_1+p_4+p_5+2p_6+p_8$	M_L

C. EXAMPLE 3

The third most complex S^{4}PR PNM among the three examples from [27], [35], [37] shown in FIGURE 13 is a classic example. The PNM has 23 places (i.e., three idle places (p_1, p_5, and p_{15}) contained 8 tokens each, 6 shared resource places ($p_{18} - p_{23}$ or signed $R_1 - R_6$) contained two tokens each, and 14 operation places) and 18 transitions. We not only recover this PNM but make a comparison with other

TABLE 3. Deadlock markings and their corresponding SPSM in Example 2.

No.	Deadlock marking No.	Corresponding SPSM
1	M_3	M_1
2	M_{27}	M_{18}
3	M_{30}	M_{18} & M_{20}
4	M_{32}	M_{20}

TABLE 4. The detailed information of control transitions in Example 2.

No.	Deadlock Markings No.	Corresponding SPSM	Control transition No.	The detailed information of CT
1	M_3	M_1	t_{c1}	(p_1+r_2) - p_2
2	M_{27}	M_{18}	t_{c2}	(p_1+r_2) - p_2
3	M_{30}	M_{18}	t_{c3}	(p_6+r_2) - p_7
		M_{20}	t_{c4}	(p_1+r_2) - p_2
4	M_{32}	M_{20}	t_{c5}	(p_6+r_2) - p_7

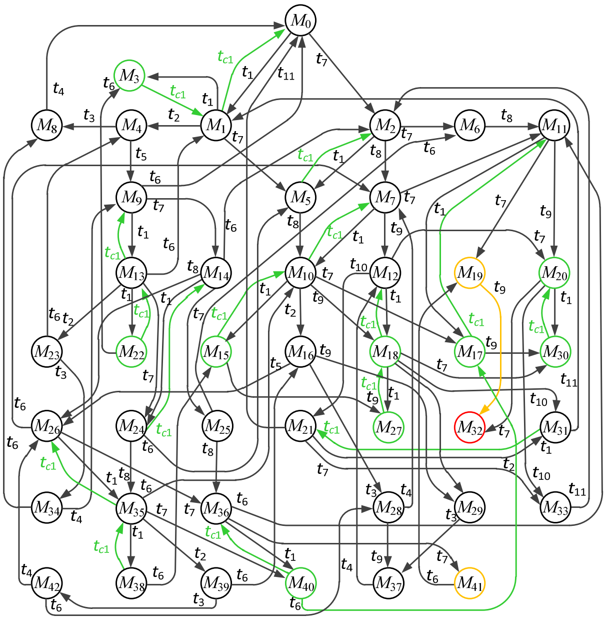

FIGURE 11. The reachability graph of Example 2 when the first controller is added into the PNM.

researches in the existing literature. Firstly, according to the reachability graph analysis, 25 deadlock markings are identified from 19300 initial reachable markings. TABLE 5 shows the detailed information of all 25 deadlock markings based on the software TINA for easy tracking and checking. Note that we do not show the detailed information of legal and quasi-deadlock markings due to the limitation of paper space. In the first iteration, 9 deadlock markings are recovered when the first controller (i.e., $^{\bullet}t = p_2$, $t^{\bullet} = p_1 + p_{19}$) is added into the PNM. In second iteration, 10 deadlock markings are recovered when the second controller (i.e., $^{\bullet}t = p_{10} + p_{14}$, $t^{\bullet} = p_{12} + p_{13}+p_{19}$) is added into the PNM. In third iteration, 4 deadlock markings are recovered when the third controller (i.e., $^{\bullet}t = p_8 + p_{15}$, $t^{\bullet} = p_5 + p_{16} + p_{19} + p_{22}$) is added into the PNM. In final iteration, the rest final 2 deadlock markings are recovered by the fourth controller (i.e., $^{\bullet}t = 2p_6 + p_{16}$, $t^{\bullet} = 2p_5 + p_{17} + p_{21}$). The controlled system still has 19300 reachable markings, and there are no more illegal markings in this PNM. In other words, the deadlock PNM is completely recovered under the proposed deadlock recovery algorithm with four control transitions (CTs). TABLE 6 shows the detailed process of adding the four controllers.

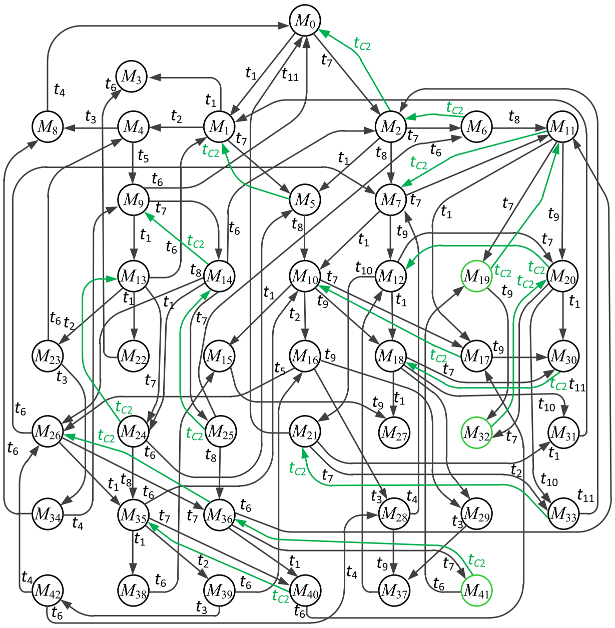

FIGURE 12. The reachability graph of the controlled S^{4}PR PNM of Example 2.

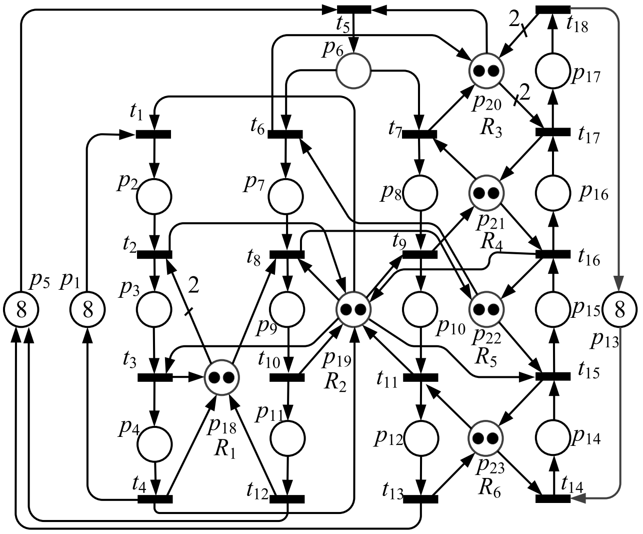

FIGURE 13. S^{4}PR PNM from [27], [35], [37].

TABLE 5. The detailed information of all deadlock markings in Example 3.

No.	Deadlock Marking No.	The detailed Information of Deadlock Marking
1	M_{4248}	$8p_1+4p_5+2p_6+2p_8+4p_{13}+2p_{14}+2p_{15}+2p_{18}$
2	M_{5513}	$8p_1+5p_5+2p_6+p_8+3p_{13}+2p_{14}+2p_{15}+p_{16}+2p_{18}$
3	M_{6361}	$7p_1+p_3+4p_5+2p_6+2p_8+4p_{13}+2p_{14}+2p_{15}$
4	M_{6944}	$8p_1+6p_5+2p_6+2p_{13}+2p_{14}+2p_{15}+2p_{16}+2p_{18}$
5	M_{7553}	$6p_1+p_2+p_3+3p_5+2p_6+p_7+2p_8+5p_{13}+2p_{14}+p_{15}$
6	M_{7718}	$7p_1+p_3+5p_5+2p_6+p_8+3p_{13}+2p_{14}+2p_{15}+p_{16}$
7	M_{8629}	$8p_1+2p_5+2p_6+p_7+2p_8+p_{10}+5p_{13}+2p_{14}+p_{15}+2p_{18}$
8	M_{9066}	$5p_1+2p_2+p_3+2p_5+2p_6+2p_7+2p_8+6p_{13}+2p_{14}$
9	M_{9082}	$6p_1+p_2+p_3+4p_5+2p_6+p_7+p_8+4p_{13}+2p_{14}+p_{15}+p_{16}$
10	M_{9224}	$7p_1+p_3+6p_5+2p_6+2p_{13}+2p_{14}+2p_{15}+2p_{16}$
11	M_{10061}	$8p_1+3p_5+2p_6+p_7+p_8+p_{10}+4p_{13}+2p_{14}+p_{15}+p_{16}+2p_{18}$
12	M_{10657}	$5p_1+2p_2+p_3+3p_5+2p_6+2p_7+p_8+5p_{13}+2p_{14}+p_{16}$
13	M_{10661}	$6p_1+p_2+p_3+5p_5+2p_6+p_7+3p_{13}+2p_{14}+p_{15}+2p_{16}$
14	M_{10976}	$7p_1+p_3+2p_5+2p_6+p_7+2p_8+p_{10}+5p_{13}+2p_{14}+p_{15}$
15	M_{11589}	$8p_1+4p_5+2p_6+p_7+p_{10}+3p_{13}+2p_{14}+p_{15}+2p_{16}+2p_{18}$
16	M_{12179}	$5p_1+2p_2+p_3+4p_5+2p_6+2p_7+4p_{13}+2p_{14}+2p_{16}$
17	M_{12228}	$6p_1+p_2+p_3+p_5+2p_6+2p_7+2p_8+p_{10}+6p_{13}+2p_{14}$
18	M_{12324}	$7p_1+p_3+3p_5+2p_6+p_7+p_8+p_{10}+4p_{13}+2p_{14}+p_{15}+p_{16}$
19	M_{13390}	$8p_1+2p_6+2p_7+2p_8+2p_{10}+6p_{13}+2p_{14}+2p_{18}$
20	M_{13627}	$6p_1+p_2+p_3+2p_5+2p_6+2p_7+p_8+p_{10}+5p_{13}+2p_{14}+p_{16}$
21	M_{13734}	$7p_1+p_3+4p_5+2p_6+p_7+p_{10}+3p_{13}+2p_{14}+p_{15}+2p_{16}$
22	M_{14507}	$8p_1+p_5+2p_6+2p_7+p_8+2p_{10}+5p_{13}+2p_{14}+p_{16}+2p_{18}$
23	M_{14990}	$6p_1+p_2+p_3+3p_5+2p_6+2p_7+p_{10}+4p_{13}+2p_{14}+2p_{16}$
24	M_{15254}	$7p_1+p_3+2p_6+2p_7+2p_8+2p_{10}+6p_{13}+2p_{14}$
25	M_{16149}	$7p_1+p_3+p_5+2p_6+2p_7+p_8+2p_{10}+5p_{13}+2p_{14}+p_{16}$

V. DISCUSSION AND COMPARISON OF THE COMPUTATIONAL COMPLEXITY WITH EXISTING LITERATURE

This section will make some comparisons with the existing literature based on the third classical complex number PNM. Notice that the "CPs" represents the number of control places, the "CTs" represents the number of control transitions, "R_L" represents the number of live reachable markings, and "% R_L" represents the recovery percent relative to the original 19300 reachable markings. TABLE 7 shows the performance comparison of eight deadlock solutions, where our work, Chen *et al.* [14], and Pan [18] are based on the control transition technology, and the others [27], [30], [35]–[37] are based on the control place technology. Further, the methods [30], [35], [36] adopt the siphon concept, but the methods [27], [37] use the P-invariant concept. The computational complexity of their methods is also shown in TABLE 7. The place-based approach needs to add control places to solve the deadlock problems of flexible manufacturing systems. Still, the proposed transition-based method adds transitions to deal with deadlock problems. From the viewpoint of permissiveness in a PNM, all the place-based methods can hold partial or all initial legal markings in the controlled net system. Still, the transition-based method can recover to all initial reachable markings, whether deadlock or quasi-deadlock markings. Therefore, from TABLE 7, it is obvious that our proposed method is the only one that can achieve real 100 % maximally permissive markings using four control transitions among all published researches.

Although Chen *et al.* [14] and Pan [18] did not apply their algorithm to this example, it has been observed that their proposed algorithm can indeed solve the deadlock problems of this S^{4}PR PNM and recover all deadlock markings. In other words, their algorithms can also achieve real 100 % maximally permissive. However, note that their algorithms belong to an NP-hard since they need to solve integer linear programming problems (ILPP). Further, in literature [14] and [28], they both need to solve ILPP formed by deadlock markings and almost all legal markings. However, this paper can identify the crucial legal marking according to our algorithm to obtain the controller directly. There's no doubt that

TABLE 6. The detailed information of deadlock markings, corresponding SPSM, and their control transitions in every iteration of Example 3.

Iteration No.	Deadlock Marking No.	Corresponding SPSM	Control Transition (CT) $^{\bullet}t$	Control Transition (CT) $t^{\bullet}$	The rest number of the deadlock markings
1	M_{7553}	M_{6363}	p_2	p_1+p_{19}	16
2	M_{8629}	M_{8769}	$p_{10}+p_{14}$	$p_{12}+p_{13}+p_{19}$	6
3	M_{5513}	M_{4217}	p_8+p_{15}	$p_5+p_{16}+p_{19}+p_{22}$	2
4	M_{6944}	M_{5506}	$2p_6+p_{16}$	$2p_5+p_{17}+p_{21}$	0

TABLE 7. Comparisons with the existing literature for Example 3.

	Park and Reveliotis [36]	Barkaoui *et al.* [30]	Hu *et al.* [35]	Uzam *et al.* [37]	Uzam *et al.* [27]	Chen *et al.* [14]	Pan [18]	Our Method
CPs	6	12	4	9	10	0	0	0
CTs	0	0	0	0	0	4	4	4
R_L	2854	10539	10613	17101	18065	19300	19300	19300
% R_L	14.78%	54.6%	54.98%	88.6%	93.6%	100%	100%	100%
Computational Complexity	Polynomial	Polynomial	Polynomial	Exponential	Exponential	NP-hard	NP-hard	Exponential

the proposed method's computational efficiency in this paper is the best among the three ones.

VI. CONCLUSION

Solving FMS's deadlock problems is a boiling issue. Many experts make their all life to do this research. They use all kinds of algorithms to achieve maximally permissive states. However, it is a challenging target even they all claimed that their controllers are optimal. In fact, they do not obtain the real optimal or maximally permissive controllers, whatever algorithms are used. Our previous research successfully calculated the optimal controllers based on control transition (CT) and recovered the S^{3}PR PNM. It is a pity that our algorithm cannot obtain optimal controllers for S^{4}PR PNM [15], [17]. In theory, the proposed method in this paper is not limited to S^{4}PR since it adopted the concept of control transition (CT). Based on scientific spirit, we merely claimed we could use our proposed method to recover the S^{4}PR PNM because all present examples in this paper belong to S^{4}PR PNM.

In this paper, a novel algorithm based on control transitions (CT) is finally proposed to eliminate deadlocks for S^{4}PR PNM. Please note that the proposed algorithm succeeds in recovering all deadlock and quasi-deadlock markings and holds all original markings, whatever they belong to what kinds of markings initially. Moreover, according to our knowledge, this paper is the first one to really apply their algorithm for S^{4}PR PNM. Most importantly, we first propose the novel concept of the shortest path selected marking (SPSM) in this domain to avoid solving integer linear programming problems (ILPP) based on our algorithm method. The contribution is obvious and very precious. Of course, some deficiencies need improvements, like the formal proof and redundant calculation. Our future work is to expand the algorithm method and apply it to more complex PNM with deadlocks recovery.

ACKNOWLEDGMENT

This research received no external funding and the authors declare no conflict of interest.

REFERENCES

[1] Y. Chen, Z. Li, K. Barkaoui, and M. Uzam, "New Petri net structure and its application to optimal supervisory control: Interval inhibitor arcs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 10, pp. 1384–1400, Oct. 2014, doi: 10.1109/TSMC.2014.2307284.

[2] Y. Chen, Z. Li, K. Barkaoui, N. Wu, and M. Zhou, "Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 47, no. 2, pp. 364–379, Feb. 2017, doi: 10.1109/TSMC.2016.2521833.

[3] J. Ezpeleta, J. M. Colom, and J. Martínez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995, doi: 10.1109/70.370500.

[4] 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, doi: 10.1109/70.650158.

[5] Z. W. Li, M. C. Zhou, and M. D. Jeng, "A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 182–188, Jan. 2008, doi: 10.1109/TASE.2006.884674.

[6] 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, doi: 10.1080/00207540010002405.

[7] 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, doi: 10.1007/s001700200014.

[8] M. Uzam, "The use of the Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems," *Int. J. Adv. Manuf. Technol.*, vol. 23, nos. 3–4, pp. 204–219, Dec. 2004, doi: 10.1007/s00170-002-1526-5.

[9] G. Y. Liu and K. Barkaoui, "A survey of siphons in Petri nets," *Inf. Sci.*, vol. 363, no. 1, pp. 198–220, Oct. 2016, doi: 10.1016/j.ins.2015.08.037.

[10] Y. S. Huang, T. C. Row, and P. J. Su, "A graph-based deadlock prevention technique for FMSs Petri nets," in *Proc. SICE Annu. Conf.*, Taipei, Taiwan, 2010, pp. 2248–2252.

[11] Y.-S. Huang, T.-H. Chung, and P.-J. Su, "Synthesis of deadlock prevention policy using Petri nets reachability graph technique," *Asian J. Control*, vol. 12, no. 3, pp. 336–346, Feb. 2010, doi: 10.1002/asjc.188.

[12] Y.-S. Huang, Y.-L. Pan, and P.-J. Su, "Transition-based deadlock detection and recovery policy for FMSs using graph technique," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 1, pp. 1–13, Jan. 2013, doi: 10.1145/2406336.2406347.

[13] X. Zhang and M. Uzam, "Transition-based deadlock control policy using reachability graph for flexible manufacturing systems," *Adv. Mech. Eng.*, vol. 8, no. 2, pp. 1–9, Feb. 2016, doi: 10.1177/1687814016631503.

[14] Y. Chen, Z. Li, A. Al-Ahmari, N. Wu, and T. Qu, "Deadlock recovery for flexible manufacturing systems modeled with Petri nets," *Inf. Sci.*, vol. 381, pp. 290–303, Mar. 2017, doi: 10.1016/j.ins.2016.11.011.

[15] T. C. Row and Y. L. Pan, "Maximally permissive deadlock prevention policies for flexible manufacturing systems using control transition," *Adv. Mech. Eng.*, vol. 10, no. 7, pp. 1–10, 2018, doi: 10.1177/1687814018787406.

[16] M. Bashir, D. Liu, M. Uzam, N. Q. Wu, A. Al-Ahmari, and Z. Li, "Optimal enforcement of liveness to flexible manufacturing systems modeled with Petri nets via transition-based controllers," *Adv. Mech. Eng.*, vol. 10, no. 1, pp. 1–19, Jan. 2018, doi: 10.1177/1687814017750707.

[17] T.-C. Row, W.-M. Syu, Y.-L. Pan, and C.-C. Wang, "One novel and optimal deadlock recovery policy for flexible manufacturing systems using iterative control transitions strategy," *Math. Problems Eng.*, vol. 2019, pp. 1–12, Mar. 2019, doi: 10.1155/2019/4847072.

[18] Y. L. Pan, "One computational innovation transition-based recovery policy for flexible manufacturing systems using Petri nets," *Appl. Sci.*, vol. 10, no. 7, pp. 1–20, Mar. 2020, doi: 10.3390/app10072332.

[19] C.-F. Zhong and Z.-W. Li, "Design of liveness-enforcing supervisors via transforming plant Petri net models of FMS," *Asian J. Control*, vol. 12, no. 3, pp. 240–252, Feb. 2010, doi: 10.1002/asjc.182.

[20] M. Uzam, Z. Li, and M. Zhou, "Identification and elimination of redundant control places in Petri net based liveness enforcing supervisors of FMS," *Int. J. Adv. Manuf. Technol.*, vol. 35, nos. 1–2, pp. 150–168, Oct. 2006, doi: 10.1007/s00170-006-0701-5.

[21] Z. Li and M. Shpitalni, "Smart deadlock prevention policy for flexible manufacturing systems using Petri nets," *IET Control Theory Appl.*, vol. 3, no. 3, pp. 362–374, Mar. 2009, doi: 10.1049/iet-cta:20070399.

[22] 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, doi: 10.1109/TSMCA.2003.820576.

[23] Z. Li and M. Zhou, "Clarifications on the definitions of elementary siphons in Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 6, pp. 1227–1229, Nov. 2006, doi: 10.1109/TSMCA.2006.878966.

[24] Z. W. Li and M. C. Zhou, "Control of elementary and dependent siphons in Petri nets and their application," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 133–148, Jan. 2008, doi: 10.1109/TSMCA.2007.909548.

[25] E. Badouel and P. Darondeau, "Theory of regions," in *Lectures on Petri Nets I: Basic Models* (Lecture Notes in Computer Science), vol. 1491, W. Reisig and G. Rozenberg, Eds. Berlin, Germany: Springer, 1998, pp. 529–586, doi: 10.1007/3-540-65306-6_22.

[26] Y. Dong, Y. Chen, S. Li, M. A. El-Meligy, and M. Sharaf, "An efficient deadlock recovery policy for flexible manufacturing systems modeled with Petri nets," *IEEE Access*, vol. 7, pp. 11785–11795, Feb. 2019, doi: 10.1109/ACCESS.2018.2889305.

[27] M. Uzam, G. Gelen, and T. L. Saleh, "Think-globally-act-locally approach with weighted arcs to the synthesis of a liveness-enforcing supervisor for generalized Petri nets modeling FMSs," *Inf. Sci.*, vol. 363, pp. 235–260, Oct. 2016, doi: 10.1016/j.ins.2015.09.010.

[28] 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, Oct. 2018, doi: 10.1109/ACCESS.2017.2763783.

[29] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989, doi: 10.1109/5.24143.

[30] 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. 1997, pp. 2755–3750, doi: 10.1109/ICSMC.1997.633253.

[31] (2019). *TINA (TIme Petri Net Analyzer) is a Toolbox for the Editing and Analysis of Petri Nets, With Possibly Inhibitor and Read Arcs, Time Petri Nets, With Possibly Priorities and Stopwatches, and an Extension of Time Petri Nets With Data Handling Called Time Transition Systems*. [Online]. Available: http://projects.laas.fr/tina/home.php

[32] C. Zhong, Z. Li, and K. Barkaoui, "Monitor design for siphon control in S4R nets: From structure analysis point of view," *Int. J. Innov. Comput. Inf. Control*, vol. 7, no. 12, pp. 6677–6690, Dec. 2011. [Online]. Available: https://www.researchgate.net/publication/267203633

[33] F. Yricas, J. M. Colom, and J. Ezpeleta, "A solution to the problem of deadlocks in concurrent systems using Petri nets and integer linear programming," in *Proc. 11th Eur. Simulation Symp.*, Erlangen, Germany, 2014, pp. 645–651.

[34] S. Wang, D. You, and M. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S 4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017, doi: 10.1109/TAC.2017.2677859.

[35] H. Hu, M. Zhou, and Z. Li, "Supervisor optimization for deadlock resolution in automated manufacturing systems with Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 4, pp. 794–804, Jun. 2011, doi: 10.1109/TASE.2011.2156783.

[36] 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, doi: 10.1109/9.956052.

[37] M. Uzam, Z. Li, and U. S. Abubakar, "Think globally act locally approach for the synthesis of a liveness-enforcing supervisor of FMSs based on Petri nets," *Int. J. Prod. Res.*, vol. 54, no. 15, pp. 4634–4657, Oct. 2015, doi: 10.1080/00207543.2015.1098785.

[38] Q. Zhuang, W. Dai, S. Wang, and F. Ning, "Deadlock prevention policy for S4PR nets based on siphon," *IEEE Access*, vol. 6, pp. 50648–50658, Sep. 2018, doi: 10.1109/ACCESS.2018.2868981.

[39] 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, doi: 10.1109/TRA.2002.807555.

[40] Y.-S. Huang and Y.-L. Pan, "Enhancement of an efficient liveness-enforcing supervisor for flexible manufacture systems," *Int. J. Adv. Manuf. Technol.*, vol. 48, nos. 5–8, pp. 725–737, May 2010, doi: 10.1007/s00170-009-2299-x.

[41] 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 Appl.*, vol. 5, no. 9, pp. 1069–1078, Jun. 2011, doi: 10.1049/iet-cta.2010.0371.

[42] Y. Huang, Y. Pan, and M. Zhou, "Computationally improved optimal deadlock control policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 2, pp. 404–415, Mar. 2012, doi: 10.1109/TSMCA.2011.2164241.

[43] 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," *J. Appl. Math.*, vol. 2013, pp. 1–11, Jan. 2013, doi: 10.1155/2013/294835.

[44] 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, doi: 10.1109/TSMCA.2008.2003535.

[45] Y.-L. Pan, C.-F. Yang, and M.-D. Jeng, "Enhancement of selective siphon control method for deadlock prevention in FMSs," *Math. Problems Eng.*, vol. 2015, Jan. 2015, Art. no. 196514.

[46] Y. Chen, Z. Li, and M. Zhou, "Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 3, pp. 615–629, Dec. 2012, doi: 10.1109/TSMCA.2011.2169956.

[47] Y.-L. Pan, C.-Y. Tseng, and T.-C. Row, "Design of improved optimal and suboptimal deadlock prevention for flexible manufacturing systems based on place invariant and reachability graph analysis methods," *J. Algorithms Comput. Technol.*, vol. 11, no. 3, pp. 261–270, Sep. 2017, doi: 10.1177/1748301817710922.

[48] Y.-L. Pan, C.-W. Tai, C.-Y. Tseng, and J.-C. Huang, "Enhancement of computational efficiency in seeking liveness-enforcing supervisors for advanced flexible manufacturing systems with deadlock states," *Appl. Sci.*, vol. 10, no. 7, pp. 1–20, Apr. 2020, doi: 10.3390/app10072620.

[49] E. A. Ashcroft, "Proving assertions about parallel programs," *J. Comput. Syst. Sci.*, vol. 10, pp. 110–135, Feb. 1975, doi: 10.1016/S0022-0000(75)80018-3.

[50] X. Cong, A. Wang, Y. Chen, N. Wu, T. Qu, M. Khalgui, and Z. Li, "Most permissive liveness-enforcing Petri net supervisors for discrete event systems via linear monitors," *ISA Trans.*, vol. 92, pp. 145–154, Sep. 2019, doi: 10.1016/j.isatra.2019.02.003.

[51] C. Zhong, W. He, Z. Li, N. Wu, and T. Qu, "Deadlock analysis and control using Petri net decomposition techniques," *Inf. Sci.*, vol. 482, pp. 440–456, May 2019, doi: 10.1016/j.ins.2019.01.029.

[52] J. Luo, Z. Liu, S. Wang, and K. Xing, "Robust deadlock avoidance policy for automated manufacturing system with multiple unreliable resources," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 3, pp. 812–821, May 2020, doi: 10.1109/JAS.2020.1003096.

TER-CHAN ROW received the B.S. degree from the Department of Electronic Engineering, Feng Chia University, Taichung, Taiwan, the M.S. degree from the Institute of Maritime Technology, National Taiwan Ocean University, Keelung, Taiwan, in 1996, and the Ph.D. degree in electrical and electronic engineering from National Defense University, Taiwan, in 2012. Since 2007, he has been an Electronic Teacher and a Petri Net and Automatic Control Researcher with Army Academy, Taiwan. Since 2018, he has been an Associate Professor with Department of Electronic Engineering, Army Academy.

YEN-LIANG PAN (Member, IEEE) received the B.S. and M.S. degrees in aeronautical engineering from Chung Cheng Institute of Technology, Taoyuan, Taiwan, in 1996 and 2001, respectively, and the Ph.D. degree in electrical and electronic engineering from National Defense University, Taipei, Taiwan, in 2011. He is currently an Associate Professor and the Chair with the General Education Center, Air Force Academy, Kaohsiung, Taiwan. He was a Visiting Scholar with the Discrete Event System Laboratory, New Jersey Institute of Technology (NJIT), in 2009. His research interests include Petri net theory and applications, computer-integrated manufacturing, automation, and supervisory control of discrete event systems. In 2012, 2013, 2014, 2015, and 2019, he was awarded excellent research at Air Force Academy. In 2013, he was invited as the Chairman of the Artificial Intelligence and Soft Computing at AsicMIC 2013. He was invited to organize one special section "Petri Nets' Applications in Intelligent Technologies and Engineering Systems," and be the Chairman at ICITES 2013.