Received November 10, 2020, accepted December 4, 2020, date of publication December 21, 2020,
date of current version December 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3045980

Partial Reachability Graph Analysis of Petri Nets for Flexible Manufacturing Systems

MENGHUAN HU[1], SHAOHUA YANG[2], AND YUFENG CHEN[3], (Senior Member, IEEE)
[1]Department of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[2]The 20th Research Institute of CETC, Xi'an 710068, China
[3]Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau

Corresponding author: Yufeng Chen (chyf01@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61673309 and Grant 61374068, and in part by the Science and Technology Development Fund, MSAR, under Grant 011/2017/A and Grant 0012/2019/A1.

ABSTRACT Petri nets are an important and popular tool to model and analyze deadlocks in flexible manufacturing systems. The state space of a Petri net model can be divided into two disjoint parts: a live-zone and a dead-zone. Reachability graph analysis plays an important role in the modeling and control of Petri nets. Most existing studies have to fully enumerate the reachable markings of a Petri net to obtain the first-met bad markings (FBMs), which exacerbates the computational overheads. In this paper, a computationally efficient method to find dead markings in Petri nets is presented. We first introduce an algorithm to find dead markings by solving an integer linear programming problem. Then, the set of markings in the dead-zone is calculated, including the set of dead markings and the set of bad markings. Then we can find all the FBMs. By using a vector covering approach, the minimal covered set of FBMs is computed. The proposed approach can obtain the dead markings and FBMs by searching only a part of a reachability graph. Finally, examples are provided to demonstrate the proposed method.

INDEX TERMS Petri net, dead marking, first-met bad marking, deadlock control.

I. INTRODUCTION

Flexible manufacturing systems (FMSs) can automatically finish various kinds of jobs by using shared resources such as robots, machines, and automated guided vehicles. The survivability and competitiveness of an automated manufacturing system largely depends on whether it can produce low-cost, high-quality products of different varieties in a short development cycle [38], [42]. This makes the flexibility of the system more and more desirable to the survival of the system. In order to shorten product cycle and reduce product cost while ensuring product quality, FMSs came into being. An FMS is usually defined as a computer control system composed of a computer numerical control machine tool and a material transfer system, and can efficiently produce small and medium batch products. Due to the competition of limited resources, deadlocks may occur in such systems [1], [11], [14], [15], [37]. Deadlocks in a system usually mean that the whole system or a part of it is blocked [9], [10], [13], [41], resulting in a reduction in productivity and major economic losses, in some cases even catastrophic consequences. The description, analysis, control, and solution of deadlocks in FMSs are undoubtedly essential to the realization and normal operation of system control. Therefore, it is necessary to handle deadlocks in these systems.

Several tools are used to deal with deadlocks in FMSs: graph theory, automata [27], and Petri nets [7], [8], [12], [16], [17], [21]–[23], [28], [35], [39], [40]. As a mathematical modeling and analysis tool, Petri nets can accurately and effectively model, analyze, and control FMSs. For a system, if we can construct its Petri net model and analyze it, we can reveal many important information about the structure and dynamic behavior of the described system. These information can be used to evaluate the performance of the system or to make suggestions for improving the design of the system. As a system model, Petri nets can describe the dynamic behavior of the system, for example, the state change of the system. Petri nets have intuitive graphical representations, and many mathematical methods can be introduced to analyze their properties. An FMS to be controlled is first modeled with Petri nets. Then, the deadlocks in the net model are analyzed and a supervisor is designed to prevent the occurrence of the deadlocks. A Petri net supervisor usually consists of control places, arcs, and transitions. The supervisor can enforce some conditions on the request of resources to ensure that deadlocks never occur. Petri nets are adopted for

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

detecting deadlocks of a system and developing a policy to prevent their occurrences. Many researchers use Petri nets as a formalism to deal with deadlock problems.

For Petri nets, there are mainly two analysis techniques to deal with deadlock problems: structural analysis [18]–[20], [24], [25], [30], [31] and reachability graph analysis [5], [32]–[34]. Reachability graph analysis is an important technique to deal with the deadlock problem in Petri nets. The reachability graph of a net model can completely reflect the behavior and the evolution of a system. Uzam and Zhou [33], [34] classify a reachability graph into two parts: a live-zone (LZ) and a deadlock-zone (DZ), where DZ contains all illegal markings (deadlocks, livelocks, and bad markings that inevitably lead to deadlocks and livelocks), and the LZ includes all legal markings. An FBM is a node in the DZ, representing the very first entry from the LZ to the DZ. Once all FBMs are forbidden, the controlled system is live since it cannot enter the DZ anymore.

The concept of FBMs is widely studied in the literature. In [3], a vector covering approach is proposed to reduce the number of legal markings and FBMs that need to be considered. Then, an integer linear programming problem (ILPP) is formulated to design a control place to forbid a selected FBM. Meanwhile, constraints are designed to ensure that all legal markings are not prohibited. Finally, a maximally permissive Petri net supervisor can be obtained. In [4], the objective function of an ILPP is used to minimize the number of control places and constraints are designed to make all FBMs unreachable but no legal marking forbidden. Hence, a compact supervisor can be obtained which is optimized in both behavioral permissiveness and structural complexity. The work in [6] focuses on the design of a control place to forbid as many FBMs as possible. Then, an iterative approach is developed to obtain a supervisor to forbid all FBMs. This study can reduce the number of constraints in the ILPPs and lead to a maximally permissive supervisor with a small number of control places. In [2], by combining FBMs and structural analysis, a suboptimal Petri net supervisor can be obtained.

All the aforementioned papers deal with deadlocks in Petri nets by using the concept of FBMs. It is not efficient to find FBMs by generating the whole reachability graph since the number of reachable markings increases exponentially with the size of a net model. The traditional methods always need to enumerate all reachability graph, which exacerbates the computational overheads. Different from traditional methods, we propose a computationally efficient method to obtain the dead markings and FBMs in Petri nets by searching only a part of a reachability graph. We first find all generalized deadlock markings of a net model. A generalized deadlock marking is one satisfying the state equation and there exist some transitions cannot be enabled anymore. Through the generalized deadlock markings, we explore all the bad markings that will inevitably lead to some generalized deadlock markings. Once all bad markings are found, then we can define the boundary markings of DZ and LZ as the set of FBMs. By using the vector covering approach, a minimal covered set of FBMs are computed. Experimental results show that the obtained minimal covering set of FBMs are the same as the one obtained by the whole reachability graph analysis.

The rest of this paper is organized as follows. Section II briefly outlines the basics of Petri nets, structural analysis, and the reachability graph analysis used throughout this paper. An approach to enumerate all generalized deadlock markings is proposed in Section III. Section IV introduces an algorithm to compute the set of all generalized bad markings and the generalized FBMs. A number of Petri net examples are presented in Section V to show the experimental results. Finally, we conclude this paper in Section VI.

II. PRELIMINARY

This section recalls some basics of Petri nets [26], [36].

A. PETRI NETS (PNs)

A Petri net is a four-tuple $N = (P, T, F, W)$, where P and T are finite and nonempty sets. P is a set of places and T is a set of transitions with $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ if $(x, y) \in F$, and $W(x, y) = 0$, otherwise, where $x, y \in P \cup T$ and $\mathbb{N}$ is the set of non-negative integers. If $\forall f \in F$, $W(f) = 1$, a Petri net N is called an ordinary net; otherwise it is called a general net. ${}^{\cdot}x = \{y \in P \cup T | (y, x) \in F\}$ is called the preset of x and $x^{\cdot} = \{y \in P \cup T | (x, y) \in F\}$ is called the postset of x. $M(p)$ denotes the number of tokens in place p. The pair (N, M_0) is called a marked Petri net or a net system. A net is pure if $\forall (x, y) \in (P \times T) \cup (T \times P)$; $W(x, y) > 0$ implies $W(y, x) = 0$. Incidence matrix $[N]$ of pure net N is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. A place p is said to be $k-$bounded ($k \in \mathbb{N}$) if $\forall M \in R(N, M_0)$, $\forall p \in P$, $M(p) \leq k$. A net is k-bounded if every place is k-bounded. A net is bounded if it is k-bounded for some k.

A transition $t \in T$ is enabled at marking M if $\forall p \in {}^{\cdot}t$, $M(p) \geq W(p, t)$. This fact is denoted as $M[t\rangle$. Once a transition t fires, it yields a new marking M', denoted as $M[t\rangle M'$, where $M'(p) = M(p) - W(p, t) + W(t, p)$. $M[t\rangle$ is the set of all markings reachable from M by firing any possible sequence of transitions. $M_0[t\rangle$ is called the set of reachable markings of net N with initial marking M_0, often denoted by $R(N, M_0)$. It can be graphically expressed by a reachability graph. The reachability graph of a net (N, M_0), denoted as $G(N, M_0)$, is a directed graph whose nodes are markings in $R(N, M_0)$ and arcs are labeled by the transitions of N.

Let (N, M_0) be a net system with $N = (P, T, F, W)$. A transition $t \in T$ is live at M_0 if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$ such that $M'[t\rangle$. (N, M_0) is live if $\forall t \in T$, and t is live at M_0. It is dead at M_0 if $\nexists t \in T$ such that $M_0[t\rangle$.

Let $N = (P, T, F, W)$ be a Petri Net and σ be a finite transition sequence. The Parikh vector $\vec{\sigma}$ is defined as $\vec{\sigma}$:

$T \to \mathbb{N}$ that assigns the number of occurrences of transition t in σ, denoted as $\vec{\sigma}(t)$. For axample, $\vec{t}_1 = (10\cdots0)^T$, $\vec{t}_2 = (010\cdots0)^T, \ldots, \vec{t}_k = (000\cdots01)^T$, $k = |T|$.

For a transition t, we have $[N](\cdot,t) = [N]\vec{t}$. $M' = M + [N](\cdot,t)$ if $M[t\rangle M'$. So if $M[t\rangle M'$, we have $M' = M + [N]\vec{t}$. For finite transition sequences σ in Petri net (N, M_0), if $M_0[\sigma\rangle M'$, we have:

$$M' = M_0 + [N]\vec{\sigma} \tag{1}$$

Equation (1) is called the state equation of Petri nets. It is only a necessary condition to judge the reachability of a marking. This means that any reachable marking satisfies the state equation, but the reverse is not true.

B. STRUCTURAL ANALYSIS

Structural analysis mainly studies the special structures in Petri nets such as places invariants, siphons, and resource loops. In a Petri net (N, M_0), a P-vector refers to such a column vector $I: P \to \mathbb{Z}$ with place as a sequence mark. Similarly, a T-vector refers to such a column vector $J: T \to \mathbb{Z}$ with transition as a sequence mark. P-vector I is called P-invariant (Place invariant, PI for short) if $I \neq 0$, and $I^T[N] = 0^T$. T-vector J is called T-invariant (Place invariant)if $J \neq 0$, and $[N]J = 0$.

If there exist a P-invariant I in a Petri Net (N, M_0) and $M \in R(N, M_0)$, we have:

$$I^T M = I^T M_0 \tag{2}$$

A P-invariant indicates a marking-invariant relationship, which holds at any reachable marking (state).

Definition 1: A set S is a Siphon if $S \subseteq P$ and ${}^{\cdot}S \subseteq S^{\cdot}$.

(1) S is a minimal siphon if the true subset of S does not contain any other siphons.

(2) S is a strict siphon if ${}^{\cdot}S \subseteq S^{\cdot}$ but ${}^{\cdot}S \neq S^{\cdot}$.

(3) S is a strict minimal siphon (SMS) if it is strict and minimal.

C. ANALYSIS OF A REACHABILITY GRAPH

A reachability graph can be partitioned into a deadlock-zone (DZ) and a live-zone (LZ) [33], [34]. The DZ contains deadlocks and critical bad markings that inevitably lead to deadlocks. The LZ contains all the legal markings, where the set of legal markings $\mathcal{M}_L$ is the maximal set of reachable markings, from which it is possible to reach initial marking M_0 without leaving $\mathcal{M}_L$. Generally, the set of legal markings of a Petri net system (N, M_0) can be defined as follows:

$$\mathcal{M}_L = \{M | M \in R(N, M_0) \wedge M_0 \in R(N, M) \tag{3}$$

An FBM is the one within the DZ, representing the very first entry from the LZ to the DZ. Hence, we provide a mathematical form of the set of FBMs as follows:

$$\mathcal{M}_{FBM} = \{M | M \in DZ, \exists M' \in LZ, t \in T, s.t. M'[t\rangle M\} \tag{4}$$

The set of dangerous markings, denoted as $\mathcal{M}_D$, can be defined as followings:

$$\mathcal{M}_D = \{M | M \in LZ, \exists M' \in DZ, t \in T, s.t. M[t\rangle M'\} \tag{5}$$

From Eqs. (2–5), we can see that the markings in $\mathcal{M}_D$ may cause the appearance of bad or dead markings. Removing all dangerous markings from $\mathcal{M}_L$, the rest are called good markings. Therefore, the set of good markings $\mathcal{M}_G$ can be defined as follows:

$$\mathcal{M}_G = \mathcal{M}_L - \mathcal{M}_D \tag{6}$$

III. EFFICIENT COMPUTATION OF DEADLOCK-ZONE MARKINGS

For optimal deadlock control purpose, reachability graph analysis is generally used to find a Petri net supervisor. The main idea is to design a set of control places to forbid all FBMs. Then, all markings in the DZ cannot be reached. However, it is not efficient to find FBMs by generating the whole reachability graph. By combining siphon and state equation, we propose an approach to enumerate all FBMs by generating a part of a reachability graph.

A. GENERALIZATIONS OF REACHABILITY ANALYSIS

Let M and M' be two markings of $R(N, M_0)$. $M' = M + [N](\cdot,t)$ if $M[t\rangle M'$. Hence, we have $M = M' - [N](\cdot,t)$. This equation means that we can get a pre-marking of M' by reversing $[N]$. The reverse transition enabled rules can be defined as follows:

Definition 2: (1)A transition t is conversely enabled at a marking M if $\forall p \in t^{\cdot}: M(p) \geq W(t,p)$. This fact is denoted by $M[-t\rangle$.

(2)If $M[-t\rangle$, conversely firing it yields a new marking M', denoted by $M[-t\rangle M'$. The new marking M' is defined as

$$M'(p) \begin{cases} M(p) + W(p,t), & p \in {}^{\cdot}t \backslash t^{\cdot} \\ M(p) - W(t,p), & p \in t^{\cdot} \backslash {}^{\cdot}t \\ M(p) + W(p,t) - W(t,p), & p \in {}^{\cdot}t \cap t^{\cdot} \\ M(p), & otherwise \end{cases} \tag{7}$$

Definition 3: For a marking M, M' is called the pre-marking of M if there exists $t \in T$, $M[-t\rangle M'$. The set of pre-markings of M is denoted as $\mathcal{M}_M{}^{pre}$.

Definition 4: For a marking M, M'' is called the post-marking of M if there exists $t \in T$, $M[t\rangle M''$. The set of post-markings of M is denoted as $\mathcal{M}_M{}^{post}$.

Let $SetDZ$ denote a set of dead-zone markings. Then for any marking M, if $\mathcal{M}_M^{post} \subseteq SetDZ$ is true, M can join $SetDZ$; otherwise, M belongs to the set of undetermined markings.

If the structure and deadlock markings of a net model are known, we can get the bad markings and dangerous markings by converse transition enabled rules. We can find all deadlock-zone markings by searching a part of rather than the whole reachability graph.

We first provide an example to illustrate the idea of explore deadlock-zone markings from deadlock markings. Fig. 1 shows the reachability graph of a simple Petri net where M_{13} and M_{14} are dead markings.

The proposed converse searching process is described as follows. The deadlock-zone contains two markings M_{13} and M_{14}, denoted as $SetDZ = \{M_{13}, M_{14}\}$. The first step is to determine the pre-markings of M_{13} and M_{14}, which are M_6,

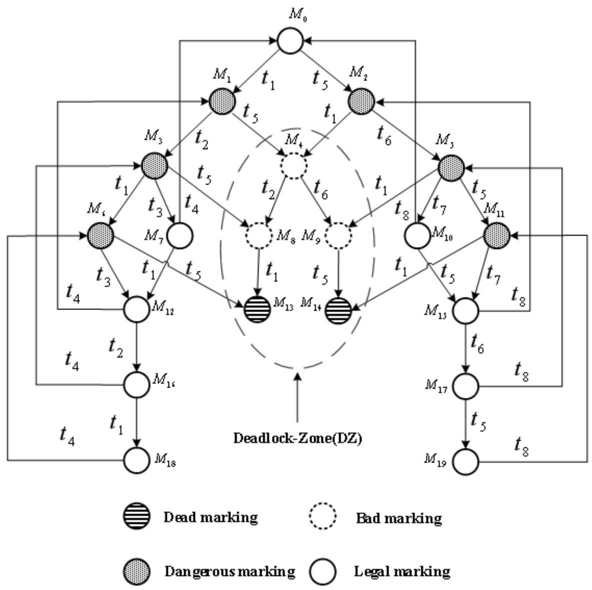

FIGURE 1. Reachability graph of a PN (N, M_0).

M_8, M_9, and M_{11}. M_{12} and M_{13} are post-markings of M_6, where M_{12} is not in *SetDZ*. Therefore, we temporarily add M_6 to the set of undetermined marking *MDD*, that is $MDD = \{M_6\}$. As M_8 has only one post-marking M_{13} that is included in *SetDZ*, we add M_8 to *SetDZ*. Add M_3 and M_4, which are the pre-makings of M_8, to *MDD*. As M_9 also has only one post-marking M_{14} that is included in *SetDZ*, we add M_9 to *SetDZ*. Add M_4 and M_5, which is the pre-making of M_9, to *MDD* ($MDD = \{M_3, M_4, M_5, M_6\}$). As M_{11} has two post-markings M_{14} and M_{15}, and M_{15} does not belong to *SetDZ*, we temporarily add M_{11} to *MDD*.

The second step is to check the markings in *MDD* ($MDD = \{M_3, M_4, M_5, M_6, M_{11}\}$), in order to determine whether there is any marking in *MDD* that can be added to *SetDZ*. Repeat this process until the markings in the *MDD* do not change. Finally, we can find all bad markings and dangerous markings. For this example, the final results are $SetDZ = \{M_4, M_8, M_9, M_{13}\}$ and $MDD = \{M_1, M_2, M_3, M_5, M_6, M_{11}\}$. This strategy only calculates a part of the reachable graph, which can reduce the computational cost.

Definition 5: Let (N, M_0) be a Petri net with n places and m transitions, and Π the set of strict minimal siphons. A marking $M = M_0 + [N]Y$ is called a generalized deadlock marking if $\exists S \in \Pi, M(S) = 0$, where Y is any $m \times 1$ vector with $Y \geq 0$. M is called a critical marking of S and S is called a critical siphon of M. The set of generalized dead markings of (N, M_0) is defined as $\mathcal{M}_{GD}$.

Definition 6: Let S be a siphon of a Petri net (N, M_0). The set of critical markings of S is defined as $\mathcal{C}(S) = \{M | M = M_0 + [N]Y, Y \geq 0, M(S) = 0\}$.

It is obvious that $\mathcal{M}_{GD} = \bigcup_{S \in \prod} \mathcal{C}(S)$.

Corollary 1: Let S be a siphon and M a critical marking of S. Then $\forall t \in S^{\cdot}, \forall M' \in R(N, M_0), M' \not[t\rangle$.

Proof: Since M is a critical marking of S, we have $M(S) = 0$. Hence, for any $p \in S$, we have $M(p) = 0$. Then, for any $t \in S^{\cdot}, M \not[t\rangle$. Since S is a siphon, $^{\cdot}S \subseteq S^{\cdot}$. Hence, for any $t \in^{\bullet} S$, $t \in S^{\cdot}$ and $M \not[t\rangle$. That is to say, no transition in $S^{\cdot}$ can fire at M. Therefore, for any $M'' \in \mathcal{M}_M^{post}, M''(S) = 0$. Hence, we have for any $t \in S^{\cdot}, M'' \not[t\rangle$. Finally, we have for any $t \in S^{\cdot}$, for any $M' \in R(N, M), M' \not[t\rangle$.

It can be seen that for any generalized deadlock marking, there exist some transitions that cannot be enabled anymore. For deadlock control purpose, the generalized deadlock markings should not be reachable.

Definition 7: Let (N, M_0) be a Petri net and $\mathcal{M}_{GD}$ the set of generalized deadlock markings of (N, M_0). The set $\mathcal{M}_{GB}$ of generalized bad markings is recursively defined as

1) $\mathcal{M}_{GD} \subseteq \mathcal{M}_{GB}$,
2) $\forall M \in \mathcal{M}_{GB}, \forall M' \in M_0 + [N]Y, M' \in \mathcal{M}_{GB}$ if $\mathcal{M}_M{}^{post} \subseteq \mathcal{M}_{GB}$.

By Definition 7, any marking in $\mathcal{M}_{GB}$ is either a generalized deadlock marking or a marking that inevitably leads to some generalized deadlock markings.

Definition 8: The set of generalized FBMs (GFBMs) is defined as

$$\mathcal{M}_{GF} = \{M | M \in \mathcal{M}_{GB}, \exists M' \notin \mathcal{M}_{GB}, \exists t \in T, M'[t\rangle M\}.$$

By Definition 8, if all markings in $\mathcal{M}_{GF}$ are forbidden, then no marking in $\mathcal{M}_{GB}$ is reachable. Similarly, we can define the set of generalized dangerous markings as follows.

Definition 9: The set of generalized dangerous markings is defined as

$$\mathcal{M}_{GDG} = \{M | M \notin \mathcal{M}_{GB}, \exists M' \in \mathcal{M}_{GB}, \exists t \in T, M[t\rangle M'\}.$$

B. COMPUTATION OF GENERALIZED DEADLOCK MARKINGS

This section presents an approach to find all generalized deadlock markings by P-invariants and strict minimal siphons of a net model. We formulate an integer linear programming problem (ILPP), namely, Find Dead-zone Marking (FDM), to calculate all the generalized dead markings of a Petri net. Let S_j be a strict minimal siphon. An ILPP, namely, FDM(S_j) is formulated to find the critical markings of S_j.

The objective function of the FDM(S_j) is defined as follows:

$$\min f = \sum_{p_i \in P} M(p_i) \tag{8}$$

In Eq. (8), M can be any marking that satisfies the state equation. Hence, $M(p_i)$ is considered as a nonnegative integer variable of the ILPP. Since any finite integer can be represented by a set of binary variables, for a place $p_i \in P$, if $M(p_i) \leq 2^{n_i} - 1$, we can use $2^0 y_{i,1} + 2^1 y_{i,2} + 2^2 y_{i,3} + \ldots + 2^{n_i - 1} y_{i,n_i}$ to replace $M(p_i)$, where $y_{i,1}, y_{i,2}, \ldots, y_{i,n_i}$ are binary variables.

For example, when $M(p_1)$ is not greater than one in a Petri net (N, M_0), $M(p_1)$ is replaced by $2^0 y_{1,1}$. When $M(p_1)$ is not greater than three, $M(p_1)$ is replaced by $2^0 y_{1,1} + 2^1 y_{1,2}$.

In this way, the objective function is converted into the following form:

$$\min f = \sum_{p_i \in P} 2^0 y_{i,1} + 2^1 y_{i,2} + \ldots + 2^{n_i - 1} y_{i,n_i} \tag{9}$$

Let (N, M_0) be a Petri net model with n_p P-invariants $I_1, I_2, \ldots, I_{n_p}$, and n_s strict minimal siphons $S_1, S_2, \ldots, S_{n_s}$. In order to find all the generalized deadlock markings, by Definition 6, we have the following constraints:

$$I_J^T \cdot M = I_J^T \cdot M_0, \forall J \in \{1, 2, \ldots, m\} \tag{10}$$

$$M\left(S_j\right) = 0 \tag{11}$$

By combining Eqs. (9)–(11), we have the following ILPP, namely FDM(S_j):

$$\min f = \sum_{p_i \in P} 2^0 y_{i,1} + 2^1 y_{i,2} + \ldots + 2^{n_i - 1} y_{i,n_i}$$

subject to:

$$I_J^T \cdot M = I_J^T \cdot M_0, \forall J \in \{1, 2, \ldots, m\} \tag{12}$$

$$M(S_j) = 0 \tag{13}$$

FDM(S_j) can be used to generate a critical marking of a siphon. In order to find all critical markings, we need to develop an iterative approach. At each iteration, a critical marking is computed by solving an ILPP and some constraints are added to eliminate the obtained siphons in the previous iterations.

Let $y^*_{i,k}$ be an optimal solution of FDM(S_j). In the next iteration, we need to exclude the current solution to find a different critical marking. Let n_i denote the number of variables $y_{i,k}$'s for place p_i. Then an additive constraint is shown as follows:

$$\sum_{y^*_{i,k}=1} y_{i,k} + \sum_{y^*_{i,l}=0} \left(1 - y_{i,l}\right) \leq \sum_{p_i \in P} n_i - 1 \tag{14}$$

.

Corollary 2: No other feasible solution of FDM(S_j) is excluded by Eq. (14).

Proof: Let $y'_{i,k}$ be a feasible solution to FDM(S_j) and there exists $k \in \{1, 2, \ldots, n_i\}$, $y'_{i,k} \neq y^*_{i,k}$. Then, we show that Eq. (14) does not exclude $y'_{i,k}$. Since there exists $k \in \{1, 2, \ldots, n_i\}$, $y'_{i,k} \neq y^*_{i,k}$, we have $\sum_{y^*_{i,k}=1} y'_{i,k} + \sum_{y^*_{i,l}=0}(1 - y'_{i,l}) \leq \sum_{p_i \in P} n_i - 1$. That is to say, the solution $y'_{i,k}$ satisfies Eq. (14). By $\sum_{y^*_{i,k}=1} y^*_{i,k} + \sum_{y^*_{i,l}=0}(1 - y^*_{i,l}) = \sum_{p_i \in P} n_i$, the solution $y^*_{i,k}$ does not satisfy Eq. (14). The conclusion holds.

Now, we can develop an algorithm to generate all generalized deadlock markings, as shown in Algorithm 1.

Corollary 3: Algorithm 1 can find all generalized deadlock markings.

Proof: Initially, FDM(S_j) can find a marking M with $M(S_i) = 0$. Hence, M is a generalized deadlock marking. In the following iteration, Eq. (14) is added to FDM(S_j) to exclude the current solution representing M and no other feasible solution is excluded. Hence, we can find a new critical marking different from M. The process is carried out until FDM(S_j) has no solution. Hence, all critical markings of S_j can be computed. Once all strict minimal siphons are dealt with, all generalized deadlock markings can be obtained.

Algorithm 1 Computation of the Set of Generalized Deadlock Markings $\mathcal{M}_{GD}$

Input: n_p P-invariant of (N, M_0) $I_1, I_2, \ldots, I_{n_p}$ and n_s strict minimal siphons $S_1, S_2, \ldots, S_{n_s}$
Output:the set of initial dead marking *SetDL*
1. $j = 1$, $\mathcal{M}_{GD} = \emptyset$
2. **while** $j < k + 1$ **do**
3. Formulate FDM(S_j) and solve it.
4. **while** FDM(S_j) has a solution **do**
5. Let M be the critical marking obtained from the solution of FDM(S_j).
6. $\mathcal{M}_{GD} = \mathcal{M}_{GD} \cup \{M\}$, $i = i + 1$.
7. **endwhile**
8. **endwhile**
9. end

FIGURE 2. The converse searching process.

IV. EXPLORATION OF DEADLOCK-ZONE MARKINGS

This section presents an algorithm to compute the set of generalized bad markings and the set of GFBMs.

A. COMPUTATION OF GENERALIZED BAD MARKINGS

The converse searching process consists of two steps. First, initially, let $SetDZ = \mathcal{M}_{GD}$ and we compute the pre-markings of *SetDZ*. Then, by checking whether the post-markings of the new generated markings are generalized bad ones or not, we can decide the set of markings to be added to *SetDZ*. The other new generated markings are added to the set of undetermined markings, denoted as *MDD*. Second, for the undetermined markings, find the pre-markings of *MDD* and add the markings that satisfy the conditions for generalized bad markings *SetDZ*. Then, the rest new generated markings are added to *MDD*. This process is carried out until the markings in the *MDD* do not change. Fig. 2 shows the process to find all generalized bad markings from the set of generalized deadlock markings.

Algorithm 2 is developed to compute the set of generalized bad markings.

In Algorithm 2, *SetDZ* stores all deadlock-zone markings, *SetPre* stores the pre-set of *SetDZ*, and *MDD* stores the undetermined markings. By applying Algorithm 2, it is possible to obtain the set of all generalized dead markings *SetDZ*. Finally, all the undetermined markings in *MDD* are generalized dangerous markings.

Algorithm 2 Computation of the Set of All Dead Markings *SetDZ*

Input: the set $\mathcal{M}_{GD}$ and the incidence matrix $[N]$
Output: the set of all generalized deadlock markings $SetDZ$ and the set of generalized dangerous marking $SetMD$
1. $SetDZ = \emptyset$, $MDD = \emptyset$, $SetMD = \emptyset$.
2. $SetNew = \mathcal{M}_{GD}$.
3. **while** $M_{new}! = \emptyset$ **do**
    $SetPre = \{M' | M \in \mathcal{M}_M^{pre}, \forall M \in SetNew\}$.
    $SetNew = \emptyset$.
    foreach $M \in SetPre$ **do**
        if $\mathcal{M}_M^{post} \subseteq SetDZ$
            $SetDZ = SetDZ \cup \{M\}$ and $SetNew = SetNew \cup \{M\}$.
        else $MDD = MDD \cup \{M\}$
        endif
    endforeach
    foreach $M \in MDD$ **do**
        if $\mathcal{M}_M^{post} \subseteq SetDZ$
            $SetDZ = SetDZ \cup \{M\}$ and $SetNew = SetNew \cup \{M\}$.
        endif
        endforeach
    endwhile
4. $SetMD = MDD$
5. end

B. COMPUTATION OF GENERALIZED FIRST-MET BAD MARKINGS

According to the set of generalized bad markings, we can compute the set of GFBMs as follows:

$$\mathcal{M}_{GF} = \left\{M \left| M \in SetDZ, \exists M' \in MDD, t \in T, M' [t\rangle M \right.\right\}$$

From the viewpoint of deadlock control, once all GFBMs are forbidden, then no marking in $\mathcal{M}_{GB}$ is reachable. When designing a Petri net supervisor to forbid GFBMs, a vector covering approach [3] can be used to reduce the number of markings to be considered, which can improve the efficiency of the deadlock control policies.

Definition 10: Let M and M' be two markings in $\mathcal{M}_{GF}$. M A-coveres M' (or M' is A-covered by M) if $\forall p \in P_A$, $M(p) \geq M'(p)$, which is denoted as $M \geq_A M'$ (or $M' \leq_A M$).

Definition 11: Let $\mathcal{M}_{GF}^*$ be a subset of $\mathcal{M}_{GF}$. $\mathcal{M}_{GF}^*$ is called a minimal covered set of GFBMs if the following two conditions are satisfied:

(1) $\forall M \in \mathcal{M}_{GF}, \exists M' \in \mathcal{M}_{GF}^*, s.t. M \geq_A M'$;

(2) $\forall M \in \mathcal{M}_{GF}^*$, there is no $M' \in \mathcal{M}_{GF}^*$ $s.t. M \geq_A M'$ and $M \neq M'$.

Algorithm 3 is developed to find the set of GFBMs $SetFBM$ and the minimal covered set of GFBMs $\mathcal{M}_{GF}^*$ of a Petri net (N, M_0).

Algorithm 3 Computation of the Set of GFBMs and the Minimal Covered Set of GFBMs

Input: the set of dangerous markings $SetMD$ and the set of dead markings $SetDZ$
Output:the first-met bad markings $SetFBM$ and the minimal covered set $SetFBM^*$
1. $SetPost = \emptyset$ and $SetFBM^* = \emptyset$.
2. **foreach** $M \in SetMD$ **do**
    $SetPost = SetPost \cup \mathcal{M}_M^{post}$.
    $SetFBM = SetPost \cap SetDZ$.
  endforeach
3. **while** $SetFBM \neq \emptyset$ **do**
    Select a marking $M \in SetFBM$.
    $SetFBM = SetFBM \setminus \{M\}$.
    foreach $M' \in SetFBM$ **do**
        if $M' \geq_A M$
            $SetFBM = SetFBM \setminus \{M'\}$.
        elseif $M' \leq_A M$
            Goto Step 2.
        endif
    endforeach
    $SetFBM^* = SetFBM^* \cup \{M\}$.
  endwhile
4. end

The proposed algorithms calculate the generalized deadlock markings through P-invariants and strict minimal siphons. We first find all the generalized deadlock markings. Then, by using a converse firing approach, we can find all generalized bad markings. Then, two kinds of border markings, generalized dangerous markings and GFBMs are obtained. Finally, by using a vector covering approach, we find the minimal covered set of GFBMs.

We consider the dead-zone markings as a multi-layered sets of markings. The upper layer consists of the union of the set of the pre-markings of the marking contained in its lower layer. The bottom layer is the set of deadlock markings, and there are no post-markings for them. The top layer set is the set of dangerous markings. The flowchart of the proposed algorithms is shown in Fig. 3.

V. EXPERIMENTAL RESULTS

This section provides some experimental results for the application of the proposed approach. We also do comparison on the efficiency between the proposed approach and the reachability graph analysis.

A. AN ILLUSTRATIVE EXAMPLE

Fig. 4 shows a Petri net with 11 places and 8 transitions. The set of idle places is $P^0 = \{p_1, p_5\}$, the set operation places is $P_A = \{p_2, p_3, p_4, p_6, p_7, p_8\}$, and the set of resource places is $P_R = \{p_9, p_{10}, p_{11}\}$. The initial marking is M_0=[3, 0, 0, 0, 3, 0, 0, 0, 1, 1, 1]^T.

For this example, the upper bound of tokens in p_1 and p_5 is three. Hence, we use two binary variables $y_{1,1}$ and $y_{1,2}$ to represent the number of tokens in p_1, i.e., $M(p_1) = y_{1,1} + 2y_{1,2}$. Similarly, we use two binary variables $y_{5,1}$ and $y_{5,2}$ to represent the number of tokens in p_5, i.e., $M(p_5) = y_{5,1} + 2y_{5,2}$. The maximum number of tokens held by the

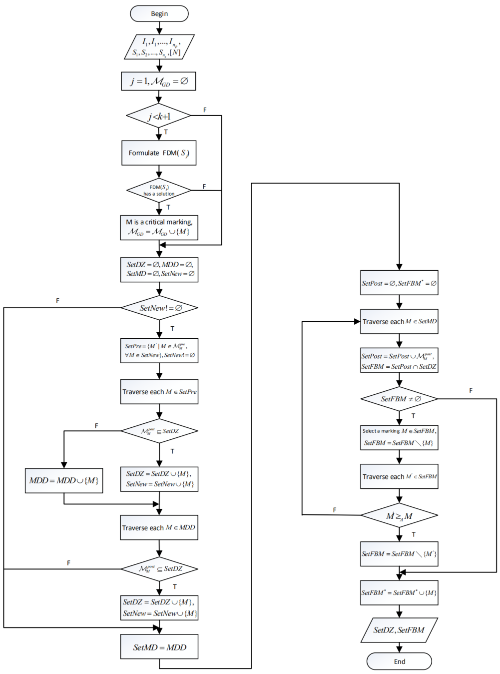

FIGURE 3. The flowchart of the proposed algorithms.

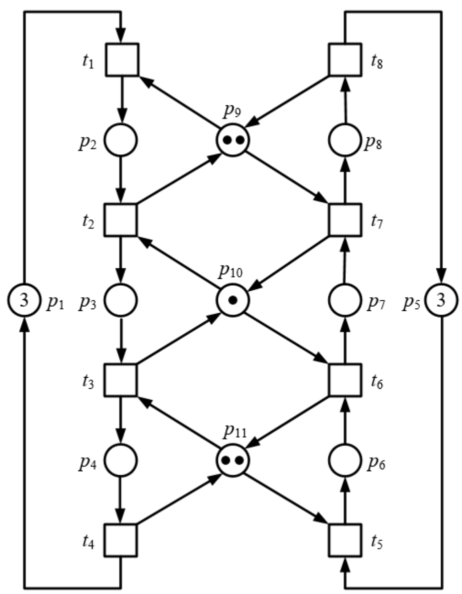

FIGURE 4. A Petri Net (N, M_0) with two processing processes FMS.

other places is one. Therefore, the number of tokens in p_i $(i \in \{2-4, 6-11\})$ is represented by a binary variable y_i.

The net model contains three strict minimal siphons:

$S_1 = \{p_4, p_8, p_9, p_{10}, p_{11}\}$,

$S_2 = \{p_3, p_8, p_9, p_{11}\}$,

$S_3 = \{p_4, p_7, p_{10}, p_{11}\}$.

And it contains five P-invariants, as shown below:

$M(p_2) + M(p_8) + M(p_9) = 1$,

$M(p_3) + M(p_7) + M(p_{10}) = 1$,

$M(p_4) + M(p_6) + M(p_{11}) = 1$,

$M(p_1) + M(p_2) + M(p_3) + M(p_4) = 3$,

$M(p_5) + M(p_6) + M(p_7) + M(p_8) = 3$.

For the five P-invariants, we have the following constraints to ensure that any obtained markings satisfy the P-invariant equations.

$y_2 + y_8 + y_9 = 1$,

$y_3 + y_7 + y_{10} = 1$,

$y_4 + y_6 + y_{11} = 1$,

$y_{1,1} + 2y_{1,2} + y_2 + y_3 + y_4 = 3$,

$y_{5,1} + 2y_{5,2} + y_6 + y_7 + y_8 = 3$.

For $S_1 = \{p_4, p_8, p_9, p_{10}, p_{11}\}$, the condition $M(S_1) = 0$ can be represented as:

$y_4 + y_8 + y_9 + y_{10} + y_{11} = 0$.

Finally, FDM(S_1) is formulated as follows:

$\min f = y_{1,1} + y_{1,2} + 2y_2 + y_3 + y_4 + y_{5,1} + y_{5,2} + y_6 + 2y_7 + y_8 + y_9 + y_{10} + y_{11}$

subject to

$y_4 + y_8 + y_9 + y_{10} + y_{11} = 0$.

$y_2 + y_8 + y_9 = 1$,

$y_3 + y_7 + y_{10} = 1$,

$y_4 + y_6 + y_{11} = 1$,

$y_{1,1} + y_{1,2} + y_2 + y_3 + y_4 = 3$,

$y_{5,1} + y_{5,2} + y_6 + y_7 + y_8 = 3$.

all variables are binary

By solving FDM(S_1), we have a solution as shown below: $\begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$.

The solution represents a generalized deadlock marking $M_1 = [1\ 1\ 1\ 0\ 2\ 1\ 0\ 0\ 0\ 0\ 0]^T$. In order to exclude M_1 from the feasible solutions of FDM(S_1), we add the following constraint to FDM(S_1): $y_{1,1} - y_{1,2} + y_2 + y_3 - y_4 - y_{5,1} + y_{5,2} + y_6 - y_7 - y_8 - y_9 - y_{10} - y_{11} \leq 4$.

We solve the ILPP again and obtain a new solution: $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$.

The solution represents a generalized deadlock marking $M_2 = [2\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 0]^T$. In order to exclude M from the feasible solutions of FDM(S_1), we add the following constraint to FDM(S_1): $-y_{1,1} + y_{1,2} + y_2 - y_3 - y_4 + y_{5,1} - y_{5,2} + y_6 + y_7 - y_8 - y_9 - y_{10} - y_{11} \leq 4$.

By solving FDM(S_1), there is no feasible solution. Hence, all critical markings of S_1 are generated. Next, we formulate FDM(S_2) to find the critical markings of S_2. To obtain FDM(S_2), we just need to replace $y_4+y_8+y_9+y_{10}+y_{11} = 0$ in FDM(S_1) by $y_3 + y_8 + y_9 + y_{11} = 0$. By solving FDM($S_2$), we get a new solution: $\begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$.

The solution represents a generalized deadlock marking $M_3 = [2\ 1\ 0\ 1\ 2\ 0\ 1\ 0\ 0\ 0\ 0]^T$. In order to exclude M_3 from the feasible solutions of FDM(S_2), we add the following constraint to FDM(S_2): $-y_{1,1} + y_{1,2} + y_2 - y_3 + y_4 - y_{5,1} + y_{5,2} - y_6 + y_7 - y_8 - y_9 - y_{10} - y_{11} \leq 4$.

Solve FDM(S_2) again and we have a new solution: $\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$.

The solution represents a generalized deadlock marking $M_3 = [2\ 1\ 0\ 0\ 2\ 0\ 1\ 0\ 0\ 0\ 1]^T$. In order to exclude M_4 from the feasible solutions of FDM(S_2), we add the following constraint to FDM(S_2):$-y_{1,1} + y_{1,2} + y_2 - y_3 - y_4 - y_{5,1} + y_{5,2} - y_6 + y_7 - y_8 - y_9 - y_{10} + y_{11} \leq 4$.

By solving FDM(S_2), there is no feasible solution. Hence, all critical markings of S_2 are generated.

Next, we formulate FDM(S_3) to find the critical markings of S_3. To obtain FDM(S_3), we just need to replace $y_3 + y_8 + y_9 + y_{11} = 0$ in FDM(S_2) by $y_4 + y_7 + y_{10} + y_{11} = 0$. By solving FDM($S_3$), we obtain a new solution: $\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$.

The solution represents a generalized deadlock marking $M_5 = [1\ 0\ 1\ 0\ 1\ 1\ 0\ 1\ 0\ 0\ 0]^T$. In order to exclude M_5 from the feasible solutions of FDM(S_3), we add the following constraint to FDM(S_3): $-y_{1,1} + y_{1,2} - y_2 + y_3 - y_4 + y_{5,1} - y_{5,2} + y_6 - y_7 + y_8 - y_9 - y_{10} - y_{11} \leq 4$. By solving FDM($S_3$) again, we obtain a new solution: $\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$.

The solution represents a generalized deadlock marking $M_6 = [2\ 0\ 1\ 0\ 2\ 1\ 0\ 0\ 1\ 0\ 0]^T$. In order to exclude M_6 from the feasible solutions of FDM(S_3), we add the following constraint to FDM(S_3):$-y_{1,1} + y_{1,2} - y_2 + y_3 - y_4 - y_{5,1} + y_{5,2} + y_6 - y_7 - y_8 + y_9 - y_{10} - y_{11} \leq 6$.

By solving FDM(S_3), there is no feasible solution. Hence, all critical markings of S_3 are generated. Finally, we find all the generalized deadlock markings.

Finally, we obtain the set of generalized deadlock markings *SetDZ*:

$$SetDZ = \begin{bmatrix} 1&1&1&0&2&1&0&0&0&0&0 \\ 2&1&0&0&1&1&1&0&0&0&0 \\ 1&1&0&1&2&0&1&0&0&0&0 \\ 2&1&0&0&2&0&1&0&0&0&1 \\ 2&0&1&0&1&1&0&1&0&0&0 \\ 2&0&1&0&2&1&0&0&1&0&0 \end{bmatrix}$$

It can be verified that *SetDZ* contains all the dead markings of the model. The set $\mathcal{M}_{Dead}$ of dead markings is

$$\mathcal{M}_{Dead} = \begin{bmatrix} 1&1&1&0&2&1&0&0&0&0&0 \\ 2&1&0&0&1&1&1&0&0&0&0 \end{bmatrix}$$

All markings in the *SetDZ* are the initial dead-zone markings obtained by Algorithm 1. By applying Algorithms 2 and 3, the final results are as follows:

$$SetDZ = \begin{bmatrix} 1&1&1&0&2&1&0&0&0&0&0 \\ 2&1&0&0&1&1&1&0&0&0&0 \\ 2&0&1&0&2&1&0&0&1&0&0 \\ 2&1&0&0&2&0&1&0&0&0&1 \\ 1&1&0&1&2&0&1&1&0&0&0 \\ 2&0&1&0&1&1&0&0&0&0&0 \\ 2&1&0&0&2&1&0&0&0&1&0 \end{bmatrix}$$

$$MDD = \begin{bmatrix} 1&1&1&0&3&0&0&0&0&0&1 \\ 2&0&1&0&3&0&0&0&1&0&1 \\ 3&0&0&0&1&1&1&0&1&0&0 \\ 3&0&0&0&2&0&1&0&1&0&1 \\ 3&0&0&0&2&1&0&0&1&1&0 \\ 2&1&0&0&3&0&0&0&0&1&1 \\ 2&0&1&1&2&0&0&1&0&0&1 \\ 2&0&0&0&2&0&1&0&1&0&0 \\ 3&0&0&0&2&1&0&0&1&1&0 \\ 2&1&0&0&3&0&0&0&0&1&1 \end{bmatrix}$$

$$\mathcal{M}_{GF} = \begin{bmatrix} 1&1&0&1&2&0&1&0&0&0&0 \\ 1&1&1&0&2&1&0&0&0&0&0 \\ 2&0&1&0&1&1&0&1&0&0&0 \\ 2&0&1&0&2&1&0&0&1&0&0 \\ 2&1&0&0&1&1&1&0&0&0&0 \\ 2&1&0&0&2&0&1&0&0&0&1 \\ 2&1&0&0&2&1&0&0&0&1&0 \end{bmatrix}$$

$$\mathcal{M}^{\star}_{GF} = \begin{bmatrix} 2&0&1&0&2&1&0&0&1&0&0 \\ 2&1&0&0&2&0&1&0&0&0&1 \\ 2&1&0&0&2&1&0&0&0&1&0 \end{bmatrix}$$

SetDZ contains seven generalized deadlock markings, and *SetMD* contains ten generalized dangerous markings. $\mathcal{M}_{GF}$ contains seven GFBMs. By using the vector covering approach, $\mathcal{M}^{*}_{FBM}$ contains three markings.

In order to demonstrate the advantage of the proposed method, we generate the reachability graph of the net model. The sets of illegal markings and FBMs, and the minimal covered set of FBMs are presented as follows:

$$\mathcal{M}_{L} = \begin{bmatrix} 1&1&1&0&2&0&0&0&0&0&0 \\ 2&1&0&0&1&1&1&0&0&0&0 \\ 2&0&1&0&2&1&0&0&1&0&0 \\ 2&1&0&0&2&0&1&0&0&0&1 \\ 2&1&0&0&2&1&0&0&0&1&0 \end{bmatrix}$$

$$\mathcal{M}_{FBM} = \begin{bmatrix} 1&1&1&0&2&1&0&0&0&0&0 \\ 2&0&1&0&2&1&0&0&1&0&0 \\ 2&1&0&0&1&1&1&0&0&0&0 \\ 2&1&0&0&2&0&1&0&0&0&1 \\ 2&1&0&0&2&1&0&0&0&1&0 \end{bmatrix}$$

$$\mathcal{M}^{\star}_{FBM} = \begin{bmatrix} 2&0&1&0&2&1&0&0&1&0&0 \\ 2&1&0&0&2&0&1&0&0&0&1 \\ 2&1&0&0&2&1&0&0&0&1&0 \end{bmatrix}$$

It can be seen that the net model has five illegal markings and five FBMs. However, we generate seven generalized bad markings that are two more than the number of illegal markings. The reason is that there are two unreachable markings generated by the proposed method. However, it can be verified that $\mathcal{M}^{\star}_{GF}$ is equal to $\mathcal{M}^{\star}_{FBM}$. Hence, from the point view of deadlock control, the proposed results are equivalent to the enumeration of all reachable markings.

B. EXPERIMENTAL RESULTS

In this section, we provide some experimental results to demonstrate the proposed approach. We develop software to implement the proposed algorithms in a 64-bit Windows operating system with a 2.60GHz CPU and 4GB RAM.

First, we apply the proposed approach to a well-studied example from the literature as shown in Fig. 5. The net model has 19 places and 14 transitions, where the places have the following partitions:$P^0 = \{p_1, p_8\}$, $P_A = \{p_2, p_3, p_4, p_5, p_6, p_7, p_9, p_{10}, p_{11}, p_{12}, p_{13}\}$, and $P_R = \{p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19}\}$. In order to show the results by generating the whole reachability graph, we use INA [29] as a tool to generate all the reachable markings.

The net model shown in Fig. 5 has only 251 reachable markings. Hence, we show more results by varying the initial

TABLE 1. Experimental results for the net in Figure 5.

p_1, p_8, p_{15}, p_{18}, and p_{19}	$\lvert\mathcal{M}_{GB}\rvert$	τ_{FDM}/s	τ_{INA}/s	$\lvert\mathcal{M}_{GB}\rvert/\lvert R(N, M_0)\rvert\%$
6,6,1,1,1	96	0.07	< 1	27.3%
7,6,2,1,1	129	0.11	< 1	19.3%
9,8,2,2,2	343	0.65	1.5	8.5%
12,11,3,3,3	959	8.96	20.25	3.5%
15,14,4,4,4	2137	145.64	430.59	1.5%
18,17,5,5,5	4282	2637.56	5170.26	0.8%

TABLE 2. Experimental results for randomly generated Petri nets.

models	$\lvert\mathcal{M}_{GB}\rvert$	τ_{FDM}/s	τ_{INA}/s	$\lvert\mathcal{M}_{GB}\rvert/\lvert R(N, M_0)\rvert\%$
$\lvert P\rvert = 11, \lvert T\rvert = 8$	9	0.01	< 1	20%
$\lvert P\rvert = 19, \lvert T\rvert = 14$	84	0.07	< 1	27.3%
$\lvert P\rvert = 29, \lvert T\rvert = 20$	5335	10.24	15.65	19.3%
$\lvert P\rvert = 42, \lvert T\rvert = 31$	2622	112.72	173.21	3.5%
$\lvert P\rvert = 59, \lvert T\rvert = 42$	7559	1051.73	1877.73	2.4%
$\lvert P\rvert = 19, \lvert T\rvert = 14$	4282	2637.56	5170.26	0.8%
$\lvert P\rvert = 91, \lvert T\rvert = 84$	11614	19635.51	84240.86	0.9%

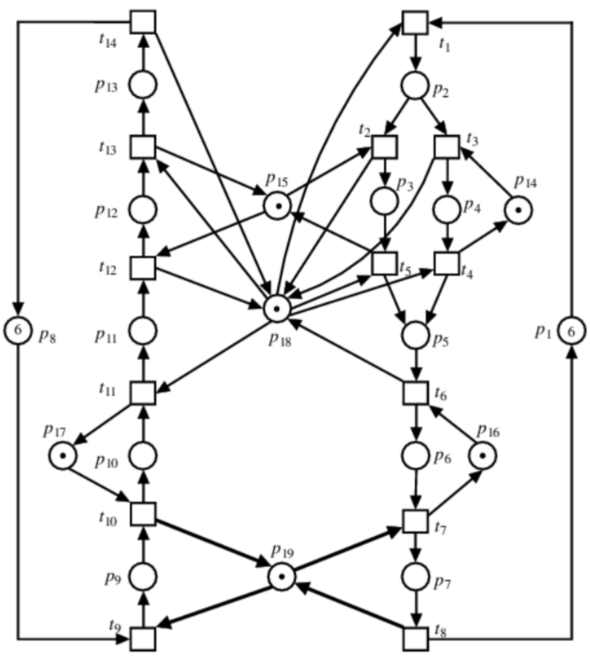

FIGURE 5. A Petri Net (N, M_0) with 19 places and 14 transitions.

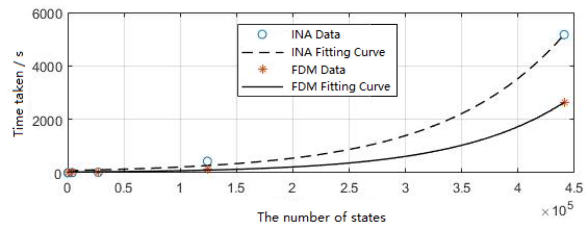

FIGURE 6. The experimental data from Table 1.

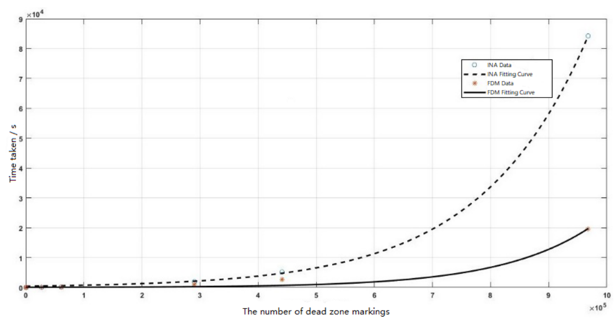

FIGURE 7. The experimental data from Table 2.

tokens in p_1, p_8, p_{15}, p_{18}, and p_{19}. The experimental results are shown in Table 1, where the first column indicates the number of tokens in the places p_1, p_8, p_{15}, p_{18}, and p_{19} at the initial marking, $|\mathcal{M}_{GB}|$ represents the number of generalized bad markings, τ_{FDM} indicates the time for the proposed approach, τ_{INA} represent the time to generate the reachability graph by INA, $|R(N, M_0)|$ indicates the number of reachable markings, and $|\mathcal{M}_{GB}|/|R(N, M_0)|\%$ represents the proportion of the number of generalized bad markings to that of reachable markings.

Fig. 6 shows the results by the data in Table 1. It can be seen that, the proposed method only needs to generate a part of the reachable markings to find all generalized bad markings and GFBMs. Hence, the proposed approach is more efficient than the enumeration of all reachable markings.

Next, the proposed approach is tested by using randomly generated Petri nets of different sizes. The results are shown in Table 2, where the first column shows the size of the generated Petri nets represented by the numbers of places and transitions, $|P|$ and $|T|$, respectively.

Fig. 7 shows the experimental results of Table 2, which can compare the efficiency between the proposed approach and the whole reachability graph based method. The following conclusions are obtained from the experimental results:

1. The proposed approach is more efficient than enumeration of the whole reachable graph since it only generates a part of the reachability graph.

2. Experimental results show that the proposed method can obtain the same minimal covered set of FBMs as the reachability graph analysis.

VI. CONCLUSION

In this paper, we first propose an algorithm to find all generalized deadlock markings of a Petri net by solving ILPPs. Then, we work out an algorithm to generate all generalized bad markings. Finally, we can compute all GFBMs and the minimal covered set of GFBMs. Experimental results show that the minimal covered covered set of GFBMs is the same as the minimal covered set of FBMs obtained by reachability graph analysis. Since the proposed approach only generates a part of the reachability graph, it is more efficient than the enumeration of the whole reachability graph. A number of examples are provided to show the efficiency of the proposed approach.

Although the proposed method can find the minimal covered set of FBMs, it cannot find the minimal covering set of legal markings. Hence, we cannot ensure the reachability of all legal markings when designing Petri net supervisors. In the future, we will consider to design a maximally permissive Petri net supervisor only by the minimal covered set of GFBMs.

REFERENCES

[1] Z. A. Banaszak and B. H. Krogh, "Deadlock avoidance in flexible manufacturing systems with concurrently competing process flows," *IEEE Trans. Robot. Autom.*, vol. 6, no. 6, pp. 724–734, Dec. 1990.

[2] D. Y. Chao, "Improvement of suboptimal siphon- and FBM-based control model of a well-known S^3PR," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 404–411, Apr. 2011.

[3] Y. Chen, Z. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 374–393, Apr. 2011.

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

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

[6] 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, May 2012.

[7] Y. F. Chen and Z. W. Li, *Optimal Supervisory Control of Automated Manufacturing Systems*. New York, NY, USA: CRC Press, Taylor & Francis Group, 2013.

[8] 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.

[9] Y. Chen, Z. Li, and M. Zhou, "Optimal supervisory control of flexible manufacturing systems by Petri nets: A set classification approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 549–563, Apr. 2014.

[10] Y. Chen, Z. Li, K. Barkaoui, and A. Giua, "On the enforcement of a class of nonlinear constraints on Petri nets," *Automatica*, vol. 55, pp. 116–124, May 2015.

[11] 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.

[12] 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.

[13] E. G. Coffman, M. J. Elphick, and A. Shoshani, "System deadlocks," *ACM Comput. Surv.*, vol. 3, no. 2, pp. 67–78, Jun. 1971.

[14] J. Ezpeleta, F. Tricas, F. Garcia-Valles, and J. M. Colom, "A banker's solution for deadlock avoidance in FMS with flexible routing and multiresource states," *IEEE Trans. Robot. Autom.*, vol. 18, no. 4, pp. 621–625, Aug. 2002.

[15] M. P. Fanti and M. Zhou, "Deadlock control methods in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 34, no. 1, pp. 5–22, Jan. 2004.

[16] 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. 137–142, Feb. 2003.

[17] H. Hu, M. Zhou, and Z. Li, "Algebraic synthesis of timed supervisor for automated manufacturing systems using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 549–557, Jul. 2010.

[18] Y.-S. Huang, M. Jeng, X. Xie, and D.-H. Chung, "Siphon-based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 36, no. 6, pp. 1248–1256, Nov. 2006.

[19] M. D. Jeng and X. L. Xie, "Deadlock detection and prevention of automated manufacturing systems using Petri nets and siphons," in *Deadlock Resolution in Computer-Integrated Systems*, M. C. Zhou and M. P. Fanti, Eds. New York, NY, USA: Marcel-Dekker Inc, 2005, pp. 233–281.

[20] Z. Li and M. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Ind. Informat.*, vol. 2, no. 4, pp. 313–325, Nov. 2006.

[21] Z. W. Li, H. S. Hu, and A. R. Wang, "Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 4, pp. 517–526, Jul. 2007.

[22] Z. Li and M. 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.

[23] Z. Li, G. Liu, H.-M. Hanisch, and M. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 42, no. 1, pp. 178–191, Jan. 2012.

[24] Z. W. Li, N. Q. Wu, and M. C. Zhou, "Deadlock control of automated manufacturing systems based on Petri nets—A literature review," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 4, pp. 437–462, Jun. 2012.

[25] D. Liu, Z. Li, and M. Zhou, "Hybrid liveness-enforcing policy for generalized Petri net models of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 43, no. 1, pp. 85–97, Jan. 2013.

[26] T. Murata, "Petri nets: Properties, analysis and application," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[27] A. Nazeem and S. Reveliotis, "Designing compact and maximally permissive deadlock avoidance policies for complex resource allocation systems through classification theory: The nonlinear case," *IEEE Trans. Autom. Control*, vol. 57, no. 7, pp. 1670–1684, Jul. 2012.

[28] 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, Jun. 2001.

[29] P. H. Starke. (2003). *INA: Integrated Net Analyzer*. [Online]. Available: http://www2.informatik.huberlin.de/starke /ina.html

[30] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpelata, "An iterative method for deadlock prevention in FMS," in *Discrete Event Systems: Analysis and Control*, G. Stremersch, Ed. Kluwer Academic: Boston, MA, 2000, pp. 139–148.

[31] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpeleta, "A Petri net structure–based deadlock prevention solution for sequential resource allocation systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 18–22, 2005, pp. 271–277.

[32] 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.

[33] M. Uzam and M. C. Zhou, "An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 44, no. 10, pp. 1987–2030, May 2006.

[34] M. Uzam and M. Zhou, "An iterative synthesis approach to Petri net-based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 37, no. 3, pp. 362–371, May 2007.

[35] N. Q. Wu and M. C. Zhou, *System Modeling and Control With Resource-Oriented Petri Nets*. New York, NY, USA: CRC Press, 2010.

[36] K. Yamalidou, J. Moody, M. Lemmon, and P. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, vol. 32, no. 1, pp. 15–28, Jan. 1996.

[37] M. Zhou and F. DiCesare, "Parallel and sequential mutual exclusions for Petri net modeling of manufacturing systems with shared resources," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 515–527, Aug. 1991.

[38] X. Guo, M. Zhou, S. Liu, and L. Qi, "Multiresource-constrained selective disassembly with maximal profit and minimal energy consumption," *IEEE Trans. Autom. Sci. Eng.*, early access, Jun. 19, 2020, doi: 10.1109/TASE.2020.2992220.

[39] X. Guo, S. Liu, M. Zhou, and G. Tian, "Disassembly sequence optimization for large-scale products with multiresource constraints using scatter search and Petri nets," *IEEE Trans. Cybern.*, vol. 46, no. 11, pp. 2435–2446, Nov. 2016.

[40] L. Qi, W. Luan, X. S. Lu, and X. Guo, "Shared P-type logic Petri net composition and property analysis: A vector computational method," *IEEE Access*, vol. 8, pp. 34644–34653, 2020.

[41] L. Qi, M. Zhou, and W. Luan, "A dynamic road incident information delivery strategy to reduce urban traffic congestion," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 5, pp. 934–945, Sep. 2018.

[42] W. Wang, G. Tian, M. Chen, F. Tao, C. Zhang, A. AI-Ahmari, Z. Li, and Z. Jiang, "Dual-objective program and improved artificial bee colony for the optimization of energy-conscious milling parameters subject to multiple constraints," *J. Cleaner Prod.*, vol. 245, Feb. 2020, Art. no. 118714.

MENGHUAN HU received the B.S. degree in measurement and control technology and instruments from Henan Polytechnic University, in 2017. She is currently pursuing the Ph.D. degree in control engineering with Xidian University, Xi'an, China. Her research interests include Petri nets and deadlock control of discrete event systems.

SHAOHUA YANG received the B.S. and M.S. degrees from Xidian University, Xi'an, China, in 2016 and 2019, respectively. He joined The 20th Research Institute, CETC, as a Researcher, in 2019. His research interests include Petri nets and deadlock control of discrete event systems.

YUFENG CHEN (Senior Member, IEEE) received the B.S. and Ph.D. degrees from Xidian University, Xi'an, China, in 2006 and 2011, respectively. He is currently with the Institute of Systems Engineering, Macau University of Science and Technology, Macau. He is the author or coauthor of over 20 publications. He is also a coauthor with Zhiwu Li, *Optimal Supervisory Control of Automated Manufacturing Systems* (CRC Press, Taylor and Francis Group, 2013). His research interests include Petri net theory and applications, supervisory control of discrete event systems, and scheduling flexible manufacturing systems.