IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 21 March 2023, accepted 23 April 2023, date of publication 2 May 2023, date of current version 12 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3272404

RESEARCH ARTICLE

Control Strategy of Discrete Event Systems Modeled by Labeled Petri Nets Based on Transition Priority

YEJIA LIU[1], **(Graduate Student Member, IEEE), XUNBO LI**[1], **AND ZHIWU LI**[2], **(Fellow, IEEE)**
[1]School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Institute of Systems Engineering, Macau University of Science and Technology, Macau, SAR, China

Corresponding author: Xunbo Li (xbli@uestc.edu.cn)

This work was supported by the Department of Industry and Information Technology of Guangxi, China under Grant Guigong2021–37.

ABSTRACT This paper deals with the supervisory control problem of discrete event systems modeled with labeled Petri nets. A transition priority matrix is proposed to control the firing of controllable transitions to prevent the system from entering illegal states. First, given a labeled Petri net system, an integer linear programming problem based on the pre-defined generalized mutual exclusion constraints and deadlocks is built to find out weakly illegal markings in its basis reachability graph. This approach is efficient since the exhaustive enumeration of the reachability space can be avoided. Second, since the firing of an uncontrollable transition sequence at a weakly illegal marking leading to an illegal state is inevitable, our goal is to prevent the system from entering weakly illegal states. A control algorithm is proposed to find a feasible transition priority matrix to avoid weakly illegal markings by controlling the firings of observable transitions. The dynamic transition priority matrix changes according to the current state of the system. Finally, two cases are studied to verify the control strategy. This control strategy does not complicate the structure of a system and can effectively avoid state-space explosion.

INDEX TERMS Control strategy, integer linear programming, labeled Petri net, transition priority matrix.

I. INTRODUCTION

Nowadays, manufacturing industry is in an environment where new technologies, materials, and products are constantly emerging. Market competition is becoming increasingly fierce. These factors lead to a dynamic market demand and a complex manufacturing process. To meet the requirements of multi-variety and small-batch production, flexible manufacturing systems (FMSs) are more and more widely used in manufacturing industry due to their efficiency and flexibility. To complete a production task, it is essential to guarantee the normal operation of the system. When the system fails (such as shut-down and buffer overflow), serious safety accidents and economic losses may occur. Therefore, regulating the system to ensure normal operation is an important prerequisite for production tasks. An FMS is a typical discrete event system (DES) characterized by a discrete state space and event-driven transition mechanism. In plain words, a DES is a dynamic system driven by a series of events, and the transformation of states is related to the occurrence of events [1]. Research frameworks for DESs mainly include automata and Petri nets. The latter has not only a rigorous mathematical theory basis but also a visual graphical representation. Various problems of DESs are analyzed with Petri nets, such as deadlock prevention, supervisor synthesis, opacity verification and enforcement, and fault detection. In this paper, we model DESs based on Petri nets, and focus on the theory of supervisory control problem of DESs.

The supervisory control theory of DESs originated from the pioneering works of Ramadge and Wonham in the 1980's [2], [3], [4]. In real-time systems, the supervisory control theory of DESs has been used for dynamic scheduling to deal with periodic and probabilistic tasks [5] and periodic tasks with multiple-periods [6]. The theory has also been used for reconfigurable coordination of DESs [7]. Deadlock control [8] is a typical supervisory control problem in resource

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

allocation systems, which has attracted much attention over the past decade [9], [10], where Petri nets are an effective tool [11]. A controlled Petri net model was first used to study the input-output behavior of DESs in [12]. The firing of transitions which is controlled by external conditions affects the evolution of the system. These executable control strategies must guarantee the system's deadlock-free operation. In particular, a number of deadlock avoidance and prevention strategies are proposed and reported by using invariant analysis [13], [14], [15], [16], [17], [18], structural analysis [19], [20], [21], [22], [23], [24], [25], and reachability analysis techniques [26], [27], [28].

For Petri nets where all transitions are controllable, Giua et al. [13] and Yamalidou et al. [14] proposed a supervisor synthesis method based on control places using place invariants. Similarly, given a Petri net prone to deadlock, Uzam and Zhou [15], [16] prevented the bad marking from reaching via a place invariant. Moody and John [17] extended the place invariant-based method to the forbidden state problem in Petri nets with uncontrollable transitions. Basile et al. [18] improved the method by introducing two parameters into the uncontrollable incidence matrix. Although the controller designed by this method has a higher behavior permissibility, it still cannot optimize all Petri nets.

A siphon is widely used for structural analysis in Petri nets [29]. The concepts of elementary siphons and independent siphons are first proposed by Li and Zhou in [19]. A mixed integer programming is used for siphon detection and complete siphon enumeration can be avoided. Monitors are only added for siphons that need to be controlled [20]. It is proven that the application of siphon-based deadlock control methods to FMS is possible. In their subsequent study [21], a monitor-based deadlock prevention policy which adds monitors for elementary siphons to a specific plant model called G-system has been shown, and initial tokens enforcing liveness to the system have been calculated by linear programing. Since finding an optimal supervisor is NP-hard, Li et al. [22] proposed a two-stage deadlock prevention method which can lower the computational cost significantly. Furthermore, for better balance between optimality and computational tractability, a near-optimal supervisor was introduced in [23] using the theory of regions and structural analysis.

A widely used method of designing a monitor is to add control places to the plant net. Feng et al. [24] have shown that a deadlock can be characterized by the saturation of a structural object in the systems of simple sequential processes with multiple resources (S^3PMR), called perfect resource transition-circuit (PRT-circuit). Each PRT-circuit is provided a control place with suitable control variables. Chen et al. dealt with the with the enforcement of nonlinear constraints on Petri nets [30], and proposed an optimal supervisor with data inhibitor arcs and only one control place [25]. Different from adding control places, Huang et al. [26] developed a new control policy by additional immediate transitions for a subclass of generalized stochastic Petri nets (GSPNs) using reachability analysis. Transition-based controllers for deadlock problems are also introduced in [31], [32], and [33].

A forbidden state problem is to design a supervisor for a Petri net to prohibit the firing of transitions to prevent the system from reaching illegal markings including deadlocks. When uncontrollable transitions exist in a Petri net system, the forbidden state problem becomes complicated since some legal markings of the system may reach illegal markings through an uncontrollable transition sequence. Based on the theory of regions, Ghaffari et al. [34] designed a maximally permissive controller for Petri nets with uncontrollable transitions. In addition, a necessary and sufficient conditions for the existence of control places realizing the maximum permissive control are given. Chen [35] proposed the definition of an uncontrollable influence subnet based on an uncontrollable path, and proved that the supervisor synthesis problem of the Petri net is only related to the uncontrollable influence subnet. To reduce the complexity of supervisor synthesis, Luo et al. [36] introduced a series of supervisor synthesis algorithms for several types of Petri nets by combining constraint transformations and Petri net simplifications. Since a control strategy based on the theory of regions usually needs to enumerate all the reachable states in the Petri net system, it will inevitably encounter the problem of state explosion. To overcome this limitation, [27] and [28] reported a strategy to build a basis reachability graph (BRG) to improve the related computational efficiency. This strategy represents the reachable markings of a Petri net with a set of states, called basis markings. In this paper, we investigate the supervisory control policy based on basis markings for labeled Petri nets.

Generally, existing supervisory control strategies result in a more structurally complex net supervisor than the plant net model, especially when the plant net is large in size. Supervisory control policies based on logic maintain the plant net structure and are convenient for online control. In real-world systems, the concept of priority is typically related to the arrangement of task processes. In the study of Petri nets, priority was first mentioned in Hack's study [37]. Investigating the semantics of priority Petri nets has been a research topic. By utilizing ordinary Petri nets, Best and Koutny [38] aimed to provide formal semantics to bounded priority Petri nets so that the constructed Petri net can retain as much of the concurrence semantics as possible without violating the priority constraints. The work in [39] showed that the relation of priority is dynamic and it is influenced by the current marking of the Petri net. A study on the boundedness, reachability, and existence of home states for priority conflict-free Petri nets was conducted in [40]. Priority can control the firing of transitions that affects the state of a net without adding additional control places. Therefore, the proposed control strategy is based on transition priority.

In this paper, we study the supervisory control problem based on transition priority in labeled Petri nets. This work has the following main contributions:

- We present an approach for identifying weakly illegal markings in a labeled Petri net system under the possibility that deadlocks and token-overflow markings may appear in the reachability graph (RG). We address the problem by using integer linear programming (ILP) based on generalized mutual exclusion constraints (GMECs) and deadlocks.
- A control algorithm is proposed to prevent the system from entering illegal states. A transition priority matrix is constructed to control the firings of transitions to prevent the system from reaching weakly illegal states. We assume that all observable transitions are controllable.

The remainder of this paper is organized as follows. Section II provides a brief introduction to Petri nets and some preliminaries. In Section III, we give the formalization of the problem considered in this paper. Section IV introduces the notion of the basis reachability graph. Section V proposes an integer linear programming method for weakly illegal markings. A control algorithm based on transition priority is presented in Section VI. Then two cases are presented in Section VII. Finally, we conclude the study in Section VIII.

II. PRELIMINARIES

The formalism and preliminary results of this study are recalled in this section. To learn more about Petri nets, we refer the reader to [11] and [41].

A. BASICS OF PETRI NETS

A Petri net is a four-tuple $N = (P, T, Pre, Post)$, where P is a set of m places, represented graphically by circles, T is a set of n transitions, represented graphically by bars. Pre: $P \times T \to \mathbb{N}$ and $Post$: $P \times T \to \mathbb{N}$ are the pre-incidence and post-incidence matrices that specify the arcs directed from places to transitions, and vice versa. The incidence matrix of a Petri net is denoted by $C = Post\text{-}\ Pre$. In this work, we denote the set of non-negative integers, integers, and real numbers as $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{R}$, respectively.

A marking of a Petri net is a vector M: $P \to \mathbb{N}$ that assigns a non-negative integer number of tokens to each place, pictorially represented by black dots. $M(p)$ indicates the number of tokens in place p at a marking M. A marking M is also denoted as $M = \Sigma_{p\in P} M(p) \cdot p$. The Petri net system $\langle N, M_0\rangle$ is a Petri net N with an initial marking M_0.

The input and output sets of a node $x \in P \cup T$ are denoted by $\cdot x$ and $x\cdot$, respectively. A string $x_1 \ldots x_r$ is called a path of net N if $x_i \in x_{i-1}\cdot$, $x_i \in P \cup T$ holds for $i = 1, \ldots, r$, $r \in N$. A circuit is a path where $x_1 = x_r$ and the other nodes are all different. A Petri net is said to be acyclic if it has no circuit.

A transition t is enabled at marking M if for all $p \in \cdot t_i$, $M(p) \geq Pre(p, t_i)$, which is denoted by $M[t_i\rangle$. A marking M' yields when an enabled transition t_i fires at M such that $M' = M + C(\cdot, t_i)$. Given a transition sequence $\sigma \in T^*$ and a marking M, $M[\sigma\rangle\ M'$ denotes that σ is enabled at marking M and a marking M' is reachable from M after firing σ. All markings reachable from the initial marking M_0 comprise the reachability set of a Petri net, denoted by $R(N, M_0)$. The reachability graph is constructed by $R(N, M_0)$ with related arcs. It fully exploits the properties of the flow relation of the net. A Petri net is bounded if there is a non-negative integer $k \in \mathbb{N}$ such that for all $p \in P$, for all $M \in R(N, M_0)$, it holds $M(p) \leq k$.

An n-dimensional column vector y: $T \to \mathbb{N}$ is defined as a firing vector. For a transition sequence σ, $y(t) = k$ indicates that transition t fires k times in σ. For $M_0[\sigma\rangle M$, we have $M = M_0 + C \cdot y$. It shows that if M is reachable from M_0, there exists a non-negative integer vector y satisfying the above equation. It is a necessary but insufficient condition. It is necessary and sufficient for acyclic nets.

Given a Petri net system $\langle N, M_0\rangle$, a transition $t \in T$ is live at the initial marking M_0 when

$$\forall M \in R(N, M_0), \exists M' \in R(N, M), M'[t\rangle$$

A Petri net system $\langle N, M_0\rangle$ is said to be live if for all $t \in T$, t is live at the initial marking M_0. The system is defined as dead if there does not exist $t \in T$ such that $M_0[t\rangle$. If for all $M \in R(N, M_0)$, there exists $t \in T$ such that $M[t\rangle$, the net system $\langle N, M_0\rangle$ is deadlock-free.

B. LABELED PETRI NET

A labeled Petri net (LPN) is a four-tuple $G = (N, M_0, E, \ell)$ where $\langle N, M_0\rangle$ is a Petri net system, E is the set of labels, and ℓ: $T \to E \cup \{\varepsilon\}$ is the labeling function that assigns to each transition $t \in T$ either a symbol from E or the empty word ε. Therefore, the set of transitions can be divided into two disjoint sets $T = T_o \cup T_u$, where $T_o = \{t \in T | \ \ell(t) \in E\}$ is the set of observable transitions and $T_u = \{t \in T | \ \ell(t) \in \varepsilon\}$ is the set of unobservable transitions. No information is generated after the transitions in T_u fire. Here we define $n_u = |T_u|$ as the cardinality of T_u and $n_o = |T_o|$ as the cardinality of T_o.

Given a sequence $\sigma_u \in T_u^*$, we define an n_u-dimensional vector y_u: $T_u \to \mathbb{N}$ as the unobservable firing vector and $y_u(t) = k$ if $t \in T_u$ is contained k times in y_u. Analogously, for a sequence $\sigma_o \in T_o^*$, an observable firing vector is defined as y_o: $T_o \to \mathbb{N}$.

The labeling function can be extended to a firing sequence $\sigma \in T^*$ and it is denoted by $\ell(\sigma)$. Given an LPN $G = (N, M_0, E, \ell)$ and a marking $M \in R(N, M_0)$, the set of observed words generated from M is defined as

$$\mathcal{L}(N, M) = \{w \in E^* \ |\exists\sigma \in T^* : M[\sigma\rangle, \ell(\sigma) = w\}$$

A string belonging to $\mathcal{L}(N, M_0)$ is known as an observation. Let w be an observation of an LPN G, we define

$$\mathcal{C}(w) = \{M \in \mathbb{N}^{\mathrm{m}} |\exists\sigma \in T^* : M_0\ [\sigma\rangle M, \ell(\sigma) = w\}$$

as the set of markings consistent with w.

An evolution of G from M is defined as a transition-marking sequence

$$s(M) = M t_{\alpha 1} M_1 t_{\alpha 2} M_2 \cdots t_{\alpha L} M_L$$

satisfying

$$M[t_{\alpha 1}\rangle M_1[t_{\alpha 2}\rangle M_2 \cdots [t_{\alpha L}\rangle M_L$$

where $\alpha_i \in\{1,2,\ldots,n\}$, $t_{\alpha i} \in T$, $M_i \in R(N, M_0)$, $i = 1,\ldots,L$, $L \in\{1,2,\ldots\}$. Let $\sigma[s(M)]$ denote the transition sequence $t_{\alpha 1}\ t_{\alpha 2}\ldots t_{\alpha L}$ in $s(M)$. The set of possible evolutions from M is defined as

$$\mathcal{S}(N, M) = \{s(M)|\sigma[s(M)] \in \mathcal{L}(N, M)\}$$

Given an LPN $G = (N, M_0, E, \ell)$ and the set of unobservable transitions T_u, the unobservable subnet $N_u = (P, T_u, Pre_u, Post_u)$ of G is the net resulting by removing all observable transitions from N, where Pre' and $Post'$ are the restrictions of Pre and $Post$ to T_u, respectively. The incidence matrix of the unobservable subnet is denoted by $C_u = Post_u$-Pre_u.

C. GENERALIZED MUTUAL EXCLUSION CONSTRAINT

Mutual exclusion constraints are used to express and analyze the concurrent use of finite resources among multiple processes. In the perspective of Petri nets, a generalized mutual exclusion constraint is a condition that limits the weighted sum of tokens in a subset of places, and it divides the set of system states into two disjoint sets: a legal marking set and a forbidden marking set [13].

A mutual exclusion constraint is a two-tuple $(\vec{w}, k)$ that defines a set of legal markings

$$M(\vec{w}, k) = \{M \in \mathbb{N}^m|\vec{w}^T \cdot M \leq k\}$$

where $\vec{w}$ is a non-negative integer weight vector, and k is a positive integer [42]. Markings violate the mutual exclusion constraint are defined as forbidden markings, i.e., illegal markings.

A generalized mutual exclusion constraint is denoted by a two-tuple $(W, \vec{k})$. Based on the GMEC $(W, \vec{k})$, a set of legal markings is defined as

$$M(W, \vec{k}) = \{M \in \mathbb{N}^m|W^T \cdot M \leq \vec{k}\}$$

where $W= [\vec{w}_1,\ldots, \vec{w}_m]$ and $\vec{k} = [k_1,\ldots, k_m]^T$.

Uncontrollable transitions may be observed, but they cannot be prevented from firing by external control conditions. Since the firings of unobservable transitions is undetectable, unobservable transitions are uncontrollable. Observable transitions can be controllable or uncontrollable [17]. Therefore, the set of transitions of an LPN G can also be divided into three disjoint sets $T = T_{co} \cup T_{uo} \cup T_{uu}$, where T_{co} is the set of controllable and observable transitions, T_{uo} is that of uncontrollable and observable transitions, and T_{uu} is that of uncontrollable and unobservable transitions. We assume that all observable transitions are controllable, i.e., $T_{co} = T_o$, $T_{uo} = \emptyset$, $T_{uu} = T_u$, leading to $T = T_{co} \cup T_{uu}$. The presence of uncontrollable transitions enhances the complexity of supervisory control problem. In this study, the GMEC helps us to distinguish illegal markings, and pave the way for the following sections.

D. TRANSITION PRIORITY

For Petri nets, a conventional method for defining priority relation is to impose it on a set of transitions. Since priority is a type of external conditions that can affect the evolution process, we only apply priority to controllable transitions. We denote $\rho(t)$ as the priority level for transition t. A generalized matrix of relation $\mathcal{P} = [r_{ij}]_{n\times n}$ to present the priority relation for transitions is proposed, named as transition priority matrix. Given a Petri net N with priority, the component of a transition priority matrix of N is given by (1):

$$\mathcal{P}(i,j) = \begin{cases} 1, & \rho(t_i) > \rho(t_j) \\ -1, & \rho(t_i) < \rho(t_j) \\ 0, & \rho(t_i) = \rho(t_j) \\ \varphi, & \text{otherwise} \end{cases} \tag{1}$$

where element 1 for r_{ij} denotes that transition t_i has priority over transition t_j. When the priority of t_i is lower than t_j, r_{ij} equals -1. If two transitions share the same priority, the element is defined as 0. For two transitions without strict priority constraints, r_{ij} is marked as φ.

Here we summarize some properties of the transition priority. Obviously, the priority relation is transitive, i.e., for $r_{ij} = 1$, $r_{jk} = 1$, $i, j, k \in\{1,\ldots,n\}$, $i \neq j, j \neq k, i \neq k$, we can infer that $r_{ik} = 1$. This inference can be extended to the situation where $r_{ij} = -1$, $r_{jk} = -1$, or $r_{ij} = 0$, $r_{jk} = 0$. For transition within different priority levels, there is $r_{ij} =$-$r_{ji} \in\{-1,1\}$, $i, j \in\{1,\ldots,n\}$, $i \neq j$. The diagonal elements of the transition priority matrix are always equal to zero, i.e., $r_{ii} = 0$, $i \in\{1,\ldots,n\}$.

Petri nets with priority have the following firing rules. Transitions in higher priority levels are always preferred to fire over transitions with lower priorities. A transition $t_i \in T$ is ρ-enabled at a marking M, denoted by $M[t_i\rangle_\rho$, if transition t_i is enabled at marking M and no transition with higher priority than t_i is enabled at M, i.e., $M[t_i\rangle$, $\forall t_j \in T$, $M[t_j\rangle$, $r_{ji} \neq 1$.

If enabled transitions are at the same priority level or with no constraints, the probability $U(t_i, M)$ of the firing of the transition t_i satisfies the uniform distribution given by (2):

$$U(t_i, M) = \begin{cases} 0, & t_i \notin T_e \\ \dfrac{1}{|T_e|}, & t_i \in T_e, \end{cases} \quad T_e = \{t|M[t\rangle\} \tag{2}$$

where $U(t_i, M)$ is the probability of firing t_i at marking M. T_e is the set that contains all enabled transitions at marking M. $|T_e|$ denotes the cardinality of T_e.

The priority relation affects the firing order of transitions. Therefore, priorities are only given to controllable transitions. Once an uncontrollable transition is enabled, it is fired immediately without any restriction.

A priority Petri net (PPN) is a three-tuple $R = (N, M_0, P)$, where $\langle N, M_0\rangle$ is a Petri net system, and P is the transition priority matrix.

Example 1: Consider the PPN in Fig. 1, where $M_0 = p_1 + p_4$. Transition t_2 has the lowest priority. Transitions t_1, t_3, *and*

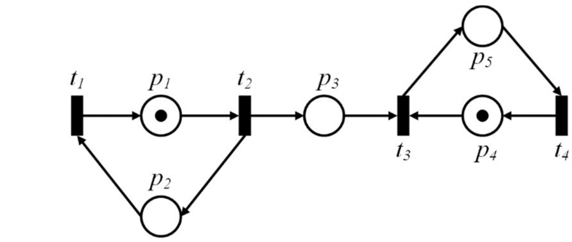

FIGURE 1. A Petri net with priority.

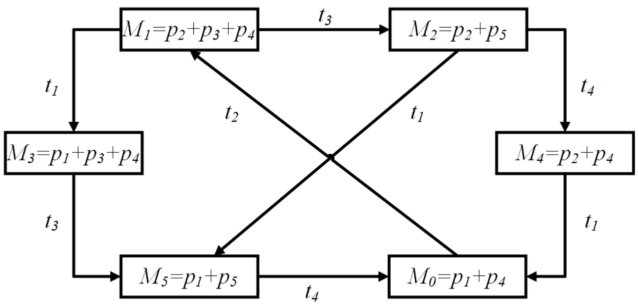

FIGURE 2. The reachability graph of the PPN in Fig. 1.

t_4 share the same priority level which is higher than $\rho(t_2)$. The corresponding priority matrix P is expressed as follows:

$$\mathcal{P}=\begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

The reachability graph of the PPN in Fig. 1 is shown in Fig. 2. The transition sequence $\sigma = t_2t_1t_3t_4$ is feasible at the initial marking M_0 and leads to itself. Since $\rho(t_2) < \rho(t_3)$, i.e., $r_{23} = -1$, transition t_2 cannot be fired at marking $M_3 = p_1 + p_3 + p_4$, and the transition sequence $\sigma' = t_2t_1t_2$ is not available from M_0.

The number of reachable markings of a PPN is less than or equal to that of the corresponding Petri net without priority. The firing sequences of a PPN are restricted by the ρ-enabling rules. If all transitions in a Petri net system have the same priority, the net system satisfies the maximally permissive behavior. If transitions are of different priorities, some markings in $R(N, M_0)$ may be forbidden. Consequently, the size of the reachability graph of a PPN is reduced compared to that of the Petri net without priority.

III. PROBLEM STATEMENT

This paper addresses the control strategy design problem based on transition priority in a labeled Petri net system that can be used to model DESs. Let us consider an LPN system $G = (N, M_0, E, \ell)$ where the set of transitions T is divided into two disjoint subnets, T_o and T_u. The system states evolve according to the firings of transitions. Unexpected illegal states, including deadlocks and token-overflow markings, may appear in $R(N, M_0)$. Thus, transition priority is needed to prevent the system from reaching the illegal states as a logical control method.

As known, the firing of unobservable transitions is unable to detect or control. Observable transitions can be either controllable or uncontrollable due to our setting. In this paper, we assume that all observable transitions are controllable. In addition, the following assumptions are made:

1) The labeled Petri net G is bounded.
2) The T_u-induced subnet is acyclic.
3) All observable transitions are controllable and all unobservable transitions are uncontrollable.

Assumption 1 ensures that the nodes in the basis reachability graph of an LPN are finite. Assumption 2 avoids the situation in which the system evolves in a circuit that only unobservable transitions are contained, such that no available output will be observed at all times. Assumption 3 simplifies the situation. We formalize the control strategy problem as follows:

Problem: Given a labeled Petri net system $G = (N, M_0, E, \ell)$ with $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$ under assumptions 1 to 3, the problem consists in calculating a transition priority matrix $\mathcal{P}$ such that it controls the firing of controllable transitions to avoid pre-defined illegal states.

IV. BASIS REACHABILITY GRAPH

Based on the assumption that the considered net system is bounded, the basis reachability graph is defined using the notion of basis markings. It is a deterministic graph and the number of nodes in the BRG equals the number of possible basis markings [27]. Before providing the BRG construction algorithm, we recall the following related definitions.

Definition 1: Given a marking M and an observable transition $t \in T_o$, we define

$$\Sigma(M, t) = \{\sigma \in T_u^* | M[\sigma\rangle M', M' \geq Pre(\cdot, t)\}$$

as a set of explanations of t at M and let the corresponding set of firing vectors called e-vectors be $Y(M, t) = \pi(\Sigma(M, t))$. The set $\Sigma(M, t)$ contains unobservable sequences whose firing at M enables t. Here we only focus on the minimal explanations, i.e., their firing vector is minimal.

Definition 2: Given a marking M and a transition $t \in T_o$, we define

$$\Sigma_{\min}(M, t) = \{\sigma \in \Sigma(M, t) | \nexists \sigma' \in \Sigma(M, t) : \pi(\sigma') < \pi(\sigma)\}$$

as the set of minimal explanations of t at M and define

$$Y_{\min}(M, t) = \pi(\Sigma_{\min}(M, t))$$

as the corresponding set of minimal e-vectors.

We adopt the approach proposed by Cabasino et al. [43] to compute $Y_{min}(M, t)$ based on the assumption that the unobservable subnet is acyclic.

The definition of the set of basis markings is constructed from the notion of explanations. A basis marking is reachable from the initial state M_0 by firing an observation w together with a sequence of unobservable transitions whose firing is strictly necessary to enable w.

Definition 3: Given an LPN $G = (N, M_0, E, \ell)$, we define $\mathcal{M}_B$ as the set of basis markings of G such that:

1) $M_0 \in \mathcal{M}_B$.
2) $\forall M \in \mathcal{M}_B, \forall t \in T_o, \forall y_u \in Y_{min}(M, t)$, it holds $M' \in \mathcal{M}_B$, where $M' = M + C(\cdot,t) + C_u \cdot y_u$.

According to the definition, the initial marking M_0 is one of the basis markings. If a marking can be reached from M_0 by firing observable transitions and their corresponding minimal explanations, the marking is added to the set of basis markings. Markings that are reachable by firing unobservable transitions only are ignored. Therefore, $\mathcal{M}_B$ is a subset of $R(N, M_0)$, whose size is generally much smaller than the original state space in practice.

Algorithm 1 shows the main steps of the BRG construction. The algorithm iteratively updates the set of basis markings and constructs the BRG based on the definition of basis markings. We denote the BRG as $B = (\mathcal{M}_B, E, \Delta, M_0)$, where $\mathcal{M}_B$ is the set of basis markings of the LPN, E is the alphabet of events, $\Delta \subseteq \mathcal{M}_B \times E \times \mathcal{M}_B$ is the corresponding transition relation for basis markings, and M_0 denotes the initial marking.

Algorithm 1 Construction of the BRG [27]

Input: A bounded labeled Petri net $G = (N, M_0, E, \ell)$.
Output: A BRG $\mathcal{B} = (\mathcal{M}_B, E, \Delta, M_0)$
1: Let $\mathcal{M}_B = \{M_0\}$, $\Delta = \{\}$. Assign no tag to M_0.
2: while markings with no tag exist
3: select a marking $M \in \mathcal{M}_B$ with no tag;
4: for all $t \in T_o$ and $Y_{min}(M, t) \neq \emptyset$
5: for all $y_u \in Y_{min}(M, t)$
6: $M' = M_0 + C_u \cdot y_u + C(\cdot, t)$;
7: if $M' \notin \mathcal{M}_B$
8: $\mathcal{M}_B = \mathcal{M}_B \cup M'$;
9: Assign no tag to M' ;
10: end if
11: $\Delta = \Delta \cup \{(M \ell(t), M')\}$;
12: end for
13: tag node M “old”
14: end for
15: end while
16: Remove all tags.

We concisely give an explanation for Algorithm 1. First, the algorithm computes basis markings from the initial marking M_0. The set of basis markings as $\mathcal{M}_B = \{M_0\}$ is initialized. For every observable transition t, we check if its corresponding set of minimal e-vectors $Y_{min}(M, t)$ is an empty set. If not, a new basis marking is computed by the equation in Step 6. A marking in $\mathcal{M}_B$ without the “old” tag has not yet been studied. The process is iterative until all markings in $\mathcal{M}_B$ are marked “old”. Since the LPN is bounded, the computation of the BRG is finite. The point of the algorithm is to explore the firing vectors of observable transitions and their minimal explanations rather than to calculate every new marking by firing a transition. Therefore, the approach brings advantages from the perspective of the computational effort. Note that when the set of unobservable transitions is empty, the size of the BRG is equivalent to its corresponding reachability graph.

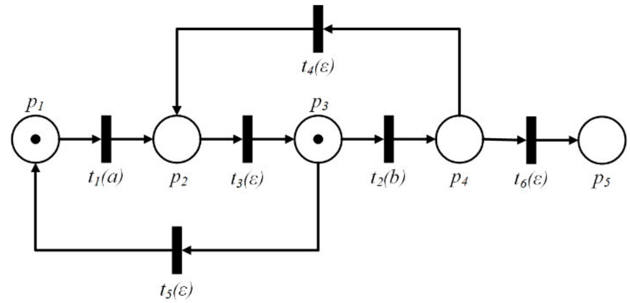

FIGURE 3. A bounded LPN with acyclic unobservable subnet.

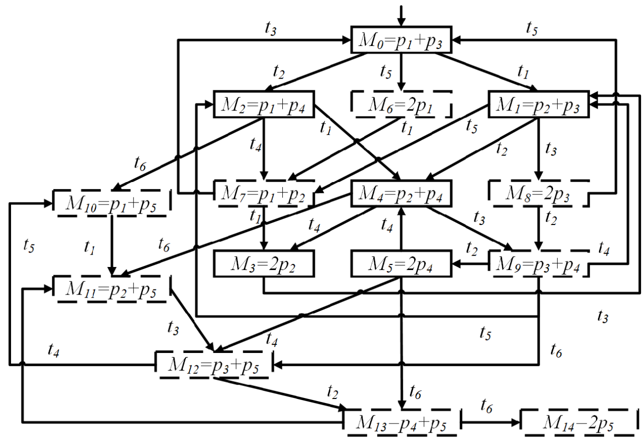

FIGURE 4. The reachability graph of the LPN in Fig. 3.

Example 2: Here we consider the LPN in Fig. 3, where $M_0 = p_1 + p_3$. The system capacity is 2. It has two observable transitions $T_o = \{t_1, t_2\}$ with $\ell(t_1) = a$, $\ell(t_2) = b$, and four unobservable transitions $T_u = \{t_3, t_4, t_5, t_6\}$. The reachability graph of the LPN in Fig. 3 contains 15 reachable markings, as illustrated in Fig. 4. Nodes with dotted lines are not included in the BRG of the given LPN in Fig. 3. The BRG is displayed in Fig. 5, which has only six basis markings.

The complexity of the BRG construction algorithm is lower than that of constructing the reachability graph. At the very worst, i.e., $T_o = T$, $T_u = \emptyset$, the BRG has the same size (complexity) as the reachability graph. The study in [44] shows that in parallel acyclic workflow cases, the cardinality of the reachability set grows exponentially with workflows and polynomially with system capacity and places. The size of the BRG can be an order of magnitude smaller than that of the reachability graph. Therefore, the construction of the BRG can effectively avoid the explosion in the state space. Table 1 presents that the state space of the RG enlarges significantly with respect to the system capacity, whereas the

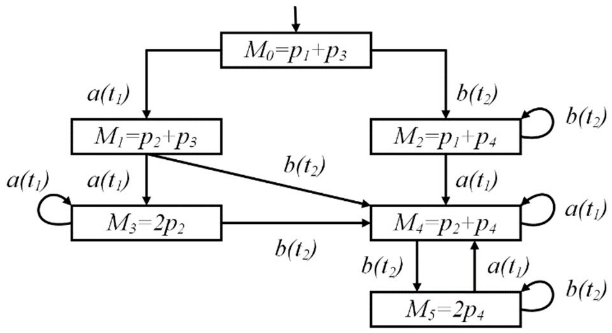

FIGURE 5. **The BRG of the LPN in Fig. 3.**

TABLE 1. **Size of the RG and The BRG of Fig. 3 with different system capacities.**

Capacity	M_0	|RG|	|BRG|	|BRG|/|RG|
2	p_1+p_3	15	6	40.0%
4	$2p_1$+$2p_3$	70	15	21.4%
8	$4p_1$+$4p_3$	495	45	9.1%
16	$8p_1$+$8p_3$	4845	153	3.2%
32	$16p_1$+$16p_3$	overtime	561	–

number of basic markings grows slowly. The BRG computation is based on an Intel(R) Xeon(R) CPU E5-2650 v2 with a basic frequency of 2.60 GHz and internal storage of 128 GB. The operating system is Windows Server 2008 R2 Enterprise with 64 bit. For the reachability graph with a calculation time of more than 8 hours, we mark its size |RG| as "overtime".

Definition 4: Given an LPN G that contains both controllable and uncontrollable transitions, i.e., $T = T_{co} \cup T_{uu}$, we define

$$R_u(M) = \{M'|M[t_1t_2\ldots t_s\rangle M', t_1, t_2, \ldots, t_s \in T_{uu}\}$$

as the set of markings reachable from uncontrollable transitions.

Proposition 5: Let $\mathcal{B}$ be the BRG of LPN G. If a path M_0–(y_1, t_1) – M_1– (y_2, t_2) – …– (y_s, t_s) – M_s exists in $\mathcal{B}$, there is $M \in R(N, M_0)$ for all $M \in R_u(M_s)$.

Proof: According to the construction process of the BRG, y_1 is the minimal e-vector from M_0 to M_1. Since the uncontrollable subnet is acyclic, if there exists $y \in Y_{min}(M, t)$, $y \geq 0$, satisfying the state equation $M + C \cdot y \geq 0$, the corresponding firing sequence σ is enabled at M. There exists corresponding firing sequence σ_1 satisfying $M_0[\sigma_1\rangle M'_0[t_1\rangle M_1$. Similarly, a conclusion is carried out that for any M_i, there exists a corresponding firing sequence σ_i satisfying $M_i[\sigma_{i+1}\rangle M'_i\ [t_{i+1}\rangle\ M_{i+1}$. Therefore, M_s and all markings in $R_u(M_s)$ are reachable from M_0.

Proposition 6: Let $\mathcal{B}$ be the BRG of the LPN G. If there exists a sequence σ and a marking M such that $M_0[\sigma\rangle M$ and $\sigma_{Tco} = t_1t_2\ldots t_s$, there exists a path M_0– (y_1, t_1) – M_1– (y_2, t_2) – …– (y_s, t_s) – M_s in $\mathcal{B}$ such that $M \in R_u(M_s)$.

Proof: Let $\sigma = \sigma_1t_1\sigma_2t_2\ldots\sigma_st_s\sigma_{s+1}$, where $\sigma_1\ldots\sigma_{k+1} \in T^*_{uu}$. The firing process from M_0 to M is shown as follows:

$$M_0[\sigma_1t_1\rangle M_1[\sigma_2t_2\rangle M_2\ldots M_{s-1}[\sigma_st_s\rangle M_s[\sigma_{s+1}\rangle M$$

If the firing vector y_1 of σ_1 is the minimal e-vector at M_0, M_1 is a basis marking. If y_1 is not the minimal one, there necessarily exists a firing vector y'_1 which is smaller than y_1, and y'_1 is the minimal e-vector, i.e., $M_0[\sigma'_1t_1\rangle M'_0$. Let $y''_1 = y_1 - y'_1$. We can prove that there exists the corresponding firing sequence σ''_1 of y''_1 satisfying $M'_0[\sigma''_1\rangle M_1$. Since $M'_0 + C\cdot y'' = M_1 \geq 0$, and the uncontrollable subnets in LPN are acyclic, σ''_1 necessarily exists. The former formula can be rewritten as

$$M_0[\sigma'_1t_1\rangle M'_1[\sigma''_1\sigma_2t_2\rangle M_2\ldots M_{s-1}[\sigma_st_s\rangle M_s[\sigma_{s+1}\rangle M$$

where M'_1 denotes a basis marking. The above logic can be applied to M_2,…,M_s recursively. Finally, we realign the firing order of transitions and obtain

$$M_0[\sigma'_1t_1\rangle M'_1[\sigma'_2t_2\rangle M'_2\ldots M'_{s-1}[\sigma'_st_s\rangle M'_s[\sigma'_{s+1}\rangle M$$

where M_0, M'_1, …, M'_s are basis markings. Consequently, there exists M_0– (y_1, t_1) – M_1– (y_2, t_2) – …– (y_s, t_s) – M_s in $\mathcal{B}$ and $M \in R_u(M_s)$, which completes the proof.

V. INTEGER LINEAR PROGRAMMING BASED ON GMEC AND DEADLOCKS

In real-world systems, some states are undesirable, such as buffer overflow and device shut-down. These states are represented by deadlock and token-overflow markings in the corresponding Petri net system. These markings that the system avoids reaching are defined as illegal markings. In this paper, we define weakly illegal markings, which are also a class of illegal markings. In the research of control strategy based on the BRG, weakly illegal markings are closely related to deadlock and token-overflow markings. Therefore, finding the weakly illegal markings is the key to subsequent research.

In this section, we put forward an approach to find weakly illegal markings by integer linear programming. First, we propose an ILP method based on a generalized mutual exclusion constraint to identify weakly illegal markings associated with the token-overflow states. Then, the ILP method for finding weakly illegal markings that may reach a deadlock state through an uncontrollable transition sequence is explained in the following subsection.

A. WEAKLY ILLEGAL MARKINGS

Definition 7: Let F be the set of illegal markings. If $M \in F$ and $R_u(M) \cap F \neq \emptyset$, marking M is a weakly illegal marking. We denote M_w as the set of weakly illegal markings.

According to the above definition, weakly illegal markings are included in the set of illegal markings. If the firing of an uncontrollable transition sequence makes the system evolve from M to an illegal marking, marking M is a weakly illegal marking. Since the assumption that observable transitions are all controllable is given, in the subsequent study, we develop control methods based on priority for observable transitions to avert the system from getting into illegal states.

Proposition 8: Let $\mathcal{B}$ be the BRG of the LPN G. If $M_0[\sigma\rangle M$, $\sigma_{Tco} = t_1t_2\ldots t_s$, and M is an illegal marking, there exists a path M_0– (y_1, t_1) – M_1– (y_2, t_2) – …– (y_s, t_s) – M_s, $M \in R_u(M_s)$, and M_s is a weakly illegal marking.

Proof: If M is an illegal marking, based on Proposition 6, there exists a path M_0– (y_1, t_1) – M_1– (y_2, t_2) – …– (y_s, t_s) – M_s in the BRG and $M \in R_u(M_s)$. In addition, according to Definition 7, M_s is a weakly illegal marking.

Proposition 9: Let $\mathcal{B}$ be the BRG of LPN G. For any firing sequence σ that satisfies $\sigma_{Tc} = t_1t_2\ldots t_s$, all markings satisfying $M_0[\sigma\rangle M$ are legal markings. For all paths M_0– $(\cdot, t_1)$ – M_1– $(\cdot, t_2)$ – …– $(\cdot, t_s)$ – M_s in $\mathcal{B}$, M_s is not a weakly illegal marking.

Proof: By contradiction, suppose that there is a path M_0–(y_1, t_1)–M_1–(y_2, t_2)–…–(y_s, t_s)–M_s in the BRG, and M_s is an illegal marking. As claimed by Proposition 5, there exists a firing sequence σ satisfying $M_0[\sigma\rangle M_s$ and $\sigma_{Tc} = t_1t_2\ldots t_s$. Since M_s is also a weakly illegal marking, the illegal marking $M \in R_u(M_s)$ exists. Therefore, $M_s[\sigma'\rangle M$, $\sigma\sigma'$ is a firing sequence satisfying $M_0[\sigma\sigma'\rangle M$ and $\sigma\sigma'_{Tc} = t_1t_2\ldots t_s$, which contradicts with the assumption. We conclude that M_s cannot be an illegal marking.

Note that based on Definition 7, the set of weakly illegal markings $\mathcal{M}_B$ is not included in the set of basis markings $\mathcal{M}_B$. Under the assumption that $T_u = T_{uu}$, $T_o = T_{co}$, given an LPN G, for any illegal marking $M_f \in$ F, a basis marking $M_b \in \mathcal{M}_B$ exists. An unobservable sequence fires at M_b and reaches M_f such that M_b is a weakly illegal marking. Therefore, as long as the system is prevented from reaching the weakly illegal markings in $\mathcal{M}_B$, it can avoid entering the corresponding illegal state.

B. THE GMEC-ILP METHOD

Generally, to find out the deadlocks and token-overflow markings, we need to filter them out from the reachability graph according to their characteristics. It is time-consuming to explore a path that a weakly illegal marking is reached from M_0 according to the reachability graph. In this subsection, we propose a GMEC-based ILP method that can quickly find weakly illegal markings that may reach the token-overflow markings, which is more efficient than the ergodic way.

For token-overflow related weakly illegal markings, feasible solutions to the ILP problem given by (3) exist when M is a weakly illegal marking:

$$\begin{cases} M + C_u \cdot y_u \geq 0 \\ W^T(M + C_u \cdot y_u) > k \\ y_u \geq 0, \quad y_u \in \mathcal{N} \end{cases} \tag{3}$$

where M defines a basis marking, C_u is the unobservable incidence matrix, and y_u defines the unobservable transitions' firing vector. If M belongs to the set of weakly illegal markings, M has its subsequent node such that the first inequality has feasible solutions. $M' = M + C_u \cdot y_u$ is an illegal marking that violates the GMEC. If the firing sequence y_u exists, elements in y must be natural numbers.

TABLE 2. **Weakly illegal marking and corresponding token-overflow marking.**

Weakly illegal marking	y_u	Token-overflow marking
M_1=$2p_1$+$2p_3$	$[0,0,1,0]^T$	$3p_1$+p_3
	$[0,0,2,0]^T$	$4p_1$
M_2=p_1+p_2+$2p_3$	$[0,0,2,0]^T$	$3p_1$+p_2
	$[1,0,3,0]^T$	$4p_1$
M_3=$2p_1$+p_3+p_4	$[0,0,1,0]^T$	$3p_1$+p_4
	$[1,1,2,0]^T$	$4p_1$
	$[0,0,1,1]^T$	$3p_1$+p_5

Example 3: As the same as the previous example, we consider the LPN in Fig. 3. Let $M_0 = 2p_1$+$2p_3$ and $P_f = \{p_1\}$. For place p_1, the GMEC is $W^T \cdot M(p_1) \leq 2$. The set of basis markings $\mathcal{M}_B$ contains 15 nodes. To ascertain the set of weaky illegal markings M_w, for each $M \in \mathcal{M}_B$, we have

$$\begin{cases} M + C_u \cdot y_u \geq 0 \\ W^T(M + C_u \cdot y_u) > 2 \\ y_u \geq 0, \quad y_u \in \mathcal{N} \end{cases}$$

By solving the above ILP inequalities, three basis markings are identified as weakly illegal markings. Table 2 lists the weakly illegal markings and the corresponding illegal states by firing y_u.

C. THE DEADLOCK-ILP METHOD

A deadlock proposed in [8] is a situation in which one or more processes are blocked in the system due to requirements which are not able to be satisfied. In a Petri net system, no transition fires at a deadlock marking. Based on the fact that a deadlock marking has no subsequent nodes, an ILP method shown in (4) is proposed to find out weakly illegal markings leading to deadlocks.

$$\begin{cases} \begin{cases} M + C_u \cdot y_u \geq 0 \\ y_{u1} \geq 0, \quad y_{u1} \in N \end{cases} & \text{(a)} \\ \begin{cases} (M + C_u \cdot y_{u1}) + C_u \cdot y_{u2} \geq 0 \\ y_{u2} \geq 0, \quad y_{u2} \in \mathcal{N} \end{cases} & \text{(b)} \end{cases} \tag{4}$$

For each basis marking M, if inequality group (a) has a feasible solution, then this basis marking M has subsequent nodes. Therefore, it is not a deadlock marking. A new marking $M' = M + C_u \cdot y_{u1}$ is obtained. When inequality group (b) has no feasible solutions, M' is proven to be a deadlock marking reached from the weakly illegal marking M.

Example 4: Let us consider again the LPN G in Fig. 3, where $M_0 = p_1 + p_3$. By solving the above ILP, deadlock marking $M = 2p_5$ is reachable from the basis marking $M = 2\,p_4$ by $y_{u1} = [0,0,0,2]^T$, i.e., t_6 fires twice.

The set of weakly illegal markings is a union of the sets obtained by solving the above two groups of ILPs. The subsequent control strategy makes sense only when the set of weakly illegal markings is not empty. In the following sections, the definitions and algorithms described above are used to design a controller.

VI. CONTROL STRATEGY

It can be seen from the above analysis that in order to ensure system liveness, it is essential to find weakly illegal markings of the system and prevent the system from entering illegal states. The following algorithms control the system by setting the transition priority matrix and reasonably arranging the firing of transitions.

A. STATIC CONTROL STRATEGY BASED ON THE TRANSITION PRIORITY MATRIX

Algorithm 2 shows how to find an appropriate transition priority matrix to control the system behavior. It iteratively computes transition priority relations until all weakly illegal markings are forbidden.

We now explicate how Algorithm 2 works in brief. When Algorithm 2 starts to run, it first ensures that the initial marking is not a deadlock. Next, the set of basis markings $\mathcal{M}_B$ is obtained based on the BRG construction algorithm. We use the ILP-based methods mentioned in Section V to determine the set of weakly illegal markings M_w. If it is not empty, i.e., the ILP-based methods have feasible solutions, the system has at least one illegal status. The next steps are crucial to the control strategy. For a weakly illegal marking M, it is vital to find the basis marking M' that reaches M and the corresponding observable transition t_i. At the same time, other basis markings and paths that can be reached from M' need to be recorded. If other paths exist, the priority of t_i is set lower than the observable transitions on other paths. It should be noted that all the priority relations must be verified. If there is no conflict, the transition priority matrix P is obtained. Otherwise, there is no feasible transition priority matrix, which means that there is no appropriate priority for avoiding all illegal states.

An optimal controller is an agent that guarantees the reachability of all legal markings and meanwhile prohibits all transitions that may lead to an illegal marking. In the BRG of an LPN with static priority constraints, some legal markings cannot be reached due to the low-priority transitions that cannot be fired. In this case, the maximally permissive behavior cannot be satisfied. To this end, we propose a dynamic control method to address the problem.

B. DYNAMIC TRANSITION PRIORITY CONTROL STRATEGY

Due to the increase in transition priority constraints, some legal states may be prohibited and a system cannot achieve maximally permissive behavior. Therefore, we propose a dynamic transition priority matrix construction method to control a system. The transition priority matrix is recalculated when the firing of a transition is observed. This ensures that the system could meets the maximally permissive behavior without entering illegal states. The following pseudocode shows how an on-line control policy on the basis of the transition priority matrix works.

Algorithm 2 Static control strategy based on the transition priority matrix

Input: A bounded LPN $G = (N, M_0, E, \ell)$, k
Output: Transition priority matrix $P = [r_{ij}]_{n_o \times n_o}$
Part 1: Initialization.
Let $\mathcal{M}_w = \{\}$.
if $\forall p \in \cdot t_i$, $t_i \in T$, $i = 1,\ldots,n$
  $M_0(p) \geq Pre(\cdot, t_i)$;
  continue;
else
  break;
end if
Part 2: Computation of the set of weakly illegal markings.
Compute BRG $\mathcal{B}$ and the set of basis markings $\mathcal{M}_B$ in accordance with Algorithm 1;
for each $M \in \mathcal{M}_B$
  Solve the ILP (3) and (4);
  if a new weakly illegal marking exists
    $\mathcal{M}_w = \mathcal{M}_w \cup M$;
  end if
end for
Part 3: Transition priority matrix construction.
Initialize the transition priority matrix:

$$\mathcal{P}_0 = \begin{bmatrix} 0 & \varphi & \cdots & \varphi \\ \varphi & 0 & \cdots & \varphi \\ \vdots & \cdots & \ddots & \varphi \\ \varphi & \cdots & \varphi & 0 \end{bmatrix}_{n_o \times n_o}$$

if $\mathcal{M}_w = \emptyset$
  let $\mathcal{P} = \mathcal{P}_0$;
  break;
end if
for all $M \in \mathcal{M}_w$
  find the set of basis markings: $\mathcal{M} = \{M' \mid M' \in \mathcal{M}_B\, M' [\sigma_{u1} t_i\rangle M\}$;
  find the set of transitions: $T = \{t_j | t_j \in T_o,\, M' [\sigma_{u2} t_j\rangle,\, j \neq i\}$;
if $T \neq \emptyset$
  for all t_j
    $r'_{ji} = 1$;
    if $(r_{ji} = \phi) \cup (r_{ji} = r'_{ji})$
      set$r_{ji} = 1$, $r_{ij} = -1$;
    else
      Conflict exists, break;
    end if
  end for
  end if
end for

The dynamic algorithm is more flexible than the static control strategy. When a transition firing is observed, the transition priority matrix changes accordingly. Basically, the dynamic control strategy is the same as the static control

Algorithm 3 On-line control strategy based on the transition priority matrix

Input: A bounded LPN $G = (N, M_0, E, \ell)$, k.
Output: Dynamic transition priority matrix $\mathcal{P} = [r_{ij}]_{n_o \times n_o}$
Part 1: Initialization.
Let $\mathcal{M}_w = \{\}$;
if $\forall p \in \cdot t_i, t_i \in T i = 1,\ldots,n$
  $M_0(p) \geq Pre(\cdot, t_i)$;
  continue;
else
  break;
end if
Part 2: Computation of the set of weakly illegal markings.
Compute the BRG $\mathcal{B}$ and the set of basis markings $\mathcal{M}_B$ according to Algorithm 1;
for each $\mathcal{M} \in \mathcal{M}_B$
  Solve the ILP (3) and (4);
  if a new weakly illegal marking exists
    $\mathcal{M}_w = \mathcal{M}_w \cup M$;
  end if
end for
Part 3: Dynamic transition priority matrix construction.
while a new word is observed
  Update current marking $M_c = M_c + C_u \cdot y_u + C(\cdot, t)$;
  Initialize the transition priority matrix:

$$\mathcal{P}_0 = \begin{bmatrix} 0 & \varphi & \cdots & \varphi \\ \varphi & 0 & \cdots & \varphi \\ \vdots & \cdots & \ddots & \varphi \\ \varphi & \cdots & \varphi & 0 \end{bmatrix}_{n_o \times n_o}$$

if $\mathcal{M}_w = \emptyset$
  let $\mathcal{P} = \mathcal{P}_0$;
  break;
end if
let the current transition priority matrix be $\mathcal{P}_c = \mathcal{P}_0$;
for all $t_i \in T_o$
    if $M_c[\sigma_{u1} t_i\rangle M'$, $M' \in M_w$
      find the set of transitions: $T = \{t_j | t_j \in T_o, M_c[\sigma_{u2} t_j\rangle M', j \neq i\}$
      if $T \neq \emptyset$
        set $r_{ji} = 1, r_{ij} = -1$;
      end if
    end if
  end for
end while

strategy. The BRG needs to be built, and the set of weakly illegal markings needs to be found with the help of the ILP. In the initial state, there are no priority constraints between transitions. Whenever the firing of an observation w is observed, the path to the weakly illegal marking needs to be found under the current basis marking M_c. If it exists, the priority of the controllable transition t_i at M_c leading to a weakly illegal marking is set to be the lowest among the simultaneously enabled transitions. Since the current transition priority matrix changes dynamically with the firing of an observable transition, the priority settings do not conflict, and maximally permissive behavior is able to be satisfied.

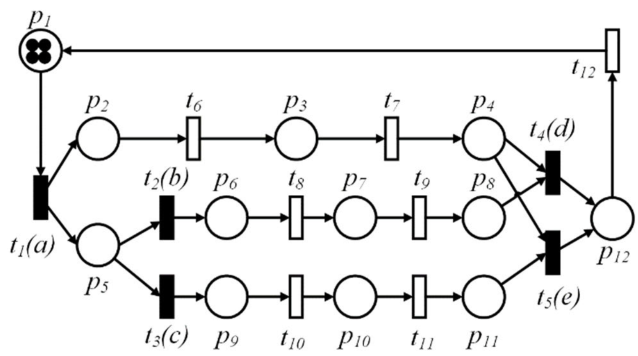

FIGURE 6. Plant.

The main computational overhead of the control algorithm includes the BRG construction mentioned in Algorithm 1 and the ILP shown in (3) and (4). The BRG construction algorithm takes exponential complexity in theory. In practice, the size of the BRG is significantly smaller than that of the reachability graph, since unobservable transitions are implicit, and the firings of which are omitted. The complexity of an ILP problem is NP-hard, and the number of constraints and variables influences the computational cost significantly. The ILP problem includes m integer variables for Min $\mathcal{M}_B$ and $7m$ constraints. In the control strategy, we check the legality of each basis marking in the BRG using (3) and (4). The cardinality of the BRG is denoted by k. As a result, we have totally $7mk$ constraints to solve. Note that the algorithm will not stop unless no observation is observed. Consequently, the complexity of the algorithm is related to the length of observation. Note that the BRG construction and the solution to the ILP can be done off-line.

VII. NUMERICAL EXAMPLE

Two cases are presented in this section to clarify how the transition priority matrix is constructed and how the control strategy works to prevent the system from entering illegal states. The reader can learn more details for manufacturing system modeling using Petri nets in [21].

A. CASE 1

Consider the net with 12 places and 12 transitions in Fig. 6 which models a producing plant. First, the plant system loads raw materials to the input (modeled by p_1). Then, each raw material is partitioned (transition t_1) into two pieces, which are put into buffers A and B (modeled by p_2 and p_5), respectively. The material in buffer A is delivered (transition t_6) to a processing machine (modeled by p_3) and then it is delivered (transition t_7) to another buffer C (modeled by p_4). There are two processing methods for materials in buffer B. Material in buffer B is delivered to processing machines (modeled

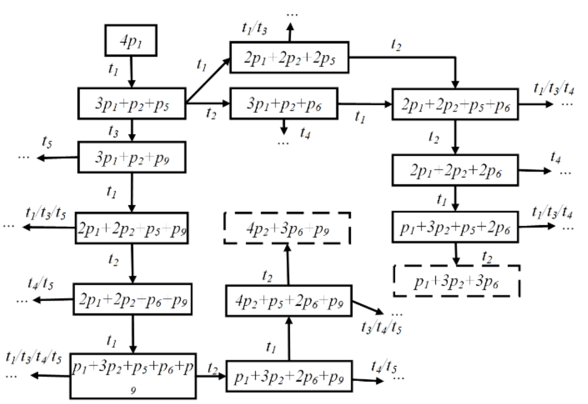

FIGURE 7. Part of the BRG of the LPN in Fig. 6.

TABLE 3. Meanings for places and transitions in Fig.6.

Places	Devices	Transitions	Behaviors
p_1	Input	t_1	Separation
p_2, p_4, p_5, p_8, p_{11}	Buffers	t_2, t_3, t_6, t_7, t_8, t_9, t_{10}, t_{11}	Delivery
p_3, p_6, p_7, p_9, p_{10}	Processing machines	t_4, t_5	Packing
p_{12}	Warehouse	t_{12}	Sending messages

by p_6 and p_7) and then delivered to buffer D (modeled by p_8). Similarly, the material in buffer B can also be processed and sent to buffer E (modeled by p_{11}). Finally, the products are packaged and delivered (transitions t_4 and t_5) to the warehouse (modeled by p_{12}). After warehousing, relevant information will be transmitted (transition t_{12}) to the input to start a new round of production. The specific meaning of each node is shown in Table 3. In Fig. 6, the set of observable transitions is $T_o = \{t_1, t_2, t_3, t_4, t_5\}$ with label $\ell = \{a, b, c, d, e\}$ and the set of unobservable transitions represented by white hollow bars is $T_u = \{t_6, t_7, t_8, t_9, t_{10}, t_{11}, t_{12}\}$. The capacity of p_8 is defined as 2, i.e., $W^T \cdot M(p_8) \leq 2$. Since the initial marking is $M_0 = [4,0,0,0,0,0,0,0,0,0,0,0]^{\mathrm{T}}$, the size of reachable markings is 5423. Part of the BRG of the plant net is illustrated in Fig. 7. It contains 70 basis markings.

For latent token-overflow markings and deadlocks, ILP problems are proposed to figure out the set of weakly illegal markings $\mathcal{M}_w$.

$$\begin{cases}\begin{cases} M + C_u \cdot y_{u1} \geq 0 \\ W^T(M + C_u \cdot y_{u1}) > 2 \\ y_{u1} \geq 0 \end{cases} \\ \begin{cases} M + C_u \cdot y_{u2} \geq 0 \\ y_{u2} \geq 0 \\ (M + C_u \cdot y_{u2}) + C_u \cdot y_{u3} \geq 0 \\ y_{u3} \geq 0 \end{cases}\end{cases}$$

Feasible solutions are found to identify weakly illegal markings which are shown in Table 4. There are only five

TABLE 4. Weakly illegal markings and illegal markings.

Weakly illegal markings	Uncontrollable e-vectors	Illegal markings
$[1,3,0,0,0,3,0,0,0,0,0,0]^{\mathrm{T}}$	$[0,0,3,3,0,0,0]^{\mathrm{T}}$	$[1,3,0,0,0,0,0,0,3,0,0,0]^{\mathrm{T}}$
$[0,4,0,0,1,3,0,0,0,0,0,0]^{\mathrm{T}}$	$[0,0,3,3,0,0,0]^{\mathrm{T}}$	$[0,4,0,0,1,0,0,0,3,0,0,0]^{\mathrm{T}}$
$[0,4,0,0,0,4,0,0,0,0,0,0]^{\mathrm{T}}$	$[0,0,3,3,0,0,0]^{\mathrm{T}}$	$[0,4,0,0,0,1,0,3,0,0,0,0]^{\mathrm{T}}$
$[0,4,0,0,0,3,0,0,1,0,0,0]^{\mathrm{T}}$	$[0,0,3,3,0,0,0]^{\mathrm{T}}$	$[0,4,0,0,0,0,0,3,1,0,0,0]^{\mathrm{T}}$
$[0,3,0,0,0,3,0,0,0,0,0,1]^{\mathrm{T}}$	$[0,0,3,3,0,0,0]^{\mathrm{T}}$	$[0,3,0,0,0,0,0,3,0,0,0,1]^{\mathrm{T}}$

weakly illegal markings in $\mathcal{M}_w$ leading to token-overflow markings.

Given an observation, the current marking of the plant can be calculated. As stated by the dynamic control strategy, we can reasonably arrange the firing order of transitions. Given an observation $w = ababa$, we can infer that the system may reach the basis marking $[1,3,0,0,1,2,0,0,0,0,0,0]^{\mathrm{T}}$. Transitions t_1, t_2, t_3, and t_4 are enabled at this state where the firing of t_2 will make the system reach a weakly illegal marking $[1,3,0,0,0,3,0,0,0,0,0,0]^{\mathrm{T}}$. Therefore, it is necessary to set the priority of transition t_2 to be lower than t_1, t_3, and t_4. The transition priority matrix is updated from P_o to P_{c1} as follows.

$$\mathcal{P}_{c1} = \begin{bmatrix} 0 & 1 & \varphi & \varphi & \varphi \\ -1 & 0 & -1 & -1 & \varphi \\ \varphi & 1 & 0 & \varphi & \varphi \\ \varphi & 1 & \varphi & 0 & \varphi \\ \varphi & \varphi & \varphi & \varphi & 0 \end{bmatrix}$$

Given an observation $w = acababa$, basis marking $M = [0,4,0,0,1,2,0,0,1,0,0,0]^{\mathrm{T}}$ may be reached. Since t_2, t_3, t_4 are enabled simultaneously at M and the firing of t_2 leads the system to a weakly illegal marking $[0,4,0,0,0,3,0,0,1,0,0,0]^{\mathrm{T}}$, we have $r_{32} = 1$, $r_{42} = 1$, $r_{52} = 1$. The current transition priority matrix P_{c2} is as follows.

$$\mathcal{P}_{c2} = \begin{bmatrix} 0 & \varphi & \varphi & \varphi & \varphi \\ \varphi & 0 & -1 & -1 & -1 \\ \varphi & 1 & 0 & \varphi & \varphi \\ \varphi & 1 & \varphi & 0 & \varphi \\ \varphi & 1 & \varphi & \varphi & 0 \end{bmatrix}$$

According to the observation, the transition priority can be flexibly changed to avoid illegal states. If we control the system by control places, one control place with two initial tokens is added. The control place is calculated to be the input place of t_2 and also the output place of t_4. The priority-based method does not require to add additional control places, which does not complicate the net structure. For different control requirements, it is easier to adjust the transition priority matrix than to reset the control places.

B. CASE 2

In this case, a manufacturing system is considered whose layout is illustrated in Fig. 8. The system contains two inputs, three exits, four machines, one buffer, four robots and two automated guided vehicles (AGVs). These three production lines yield distinct types of products.

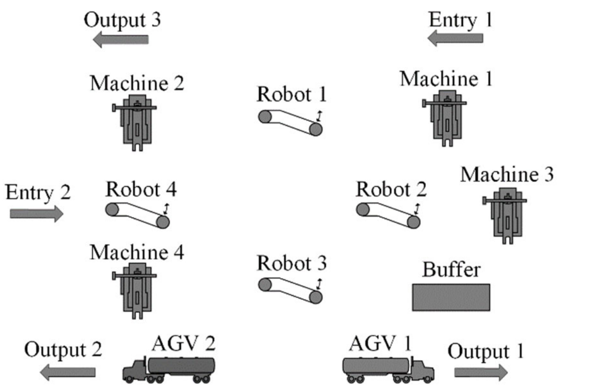

FIGURE 8. Layout of a manufacturing system with three production lines.

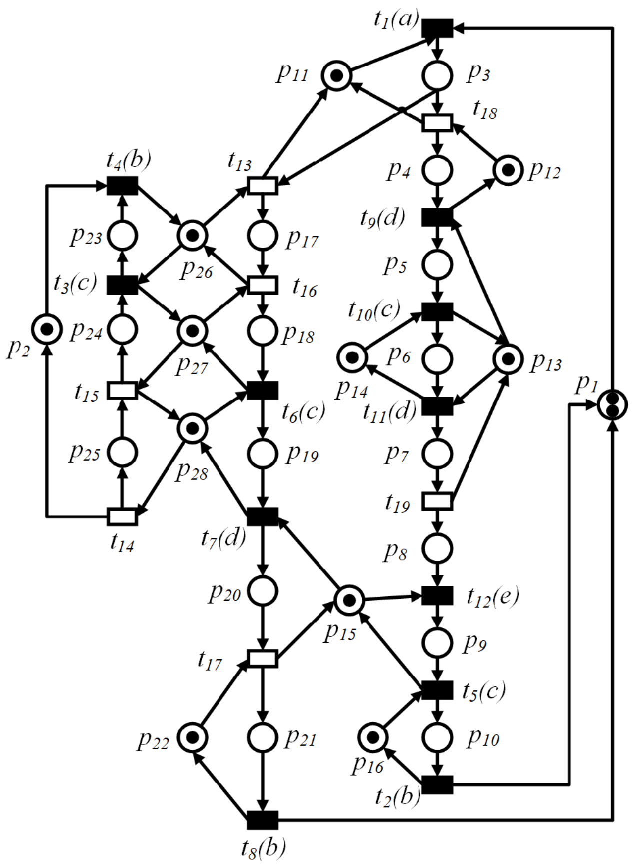

FIGURE 9. The LPN of Fig. 8.

Line 1: First, robot R1 brings raw materials from entry E1. Then, the materials are loaded into machine M1. The raw pieces are processed by M1 and are sent to machine M3 by robot R2. After processing by M3, robot R2 takes them to buffer B. Finally, robot R3 takes the products and loads the AGV1. It sends the end product to the output O1 and new production begins at E1.

Line 2: Robot R1 takes raw stuff from E1 and machine M2 is loaded. Intermediate products are generated by machine M2. Robot R4 carries them to machine M4. The final products are made by machine M4. Finally, AGV2 is loaded by robot R3 from M4. The products are sent to O2 and new raw materials are sent to E1.

Line 3: Machine M4 processes raw materials from entry E2. Robot R4 transfers intermediate parts from M4 to machine M2. Products are sent to exit O3 directly and new raw materials are sent to E2.

It is noteworthy that two production lines share robots R1 and R3. Robot R1 is shared by machines M1 and M2. Robot R3 is used by both machine M4 and buffer B. Part of production line 2 is the reverse of the processing steps of production line 3, and this structure may cause deadlocks.

The labeled Petri net model of Fig. 8 is shown in Fig. 9 with 28 places and 19 transitions. The set of unobservable transitions is $T_u = \{t_{13}, t_{14}, t_{15}, t_{16}, t_{17}, t_{18}, t_{19}\}$. The label set and its mapping are $E = \{`a';'b';'c';'d';'e'\}$ and $\ell = \{[t_1];[t_2, t_4, t_8];\ [t_3, t_5, t_6, t_{10}];[t_7, t_9, t_{11}];[t_{12}]\}$, respectively. The capacity of place p_8 is 1, i.e., $W^T \cdot M(p_8) \leq 1$. The initial marking $M_0 = 2p_1 + p_2 + p_{11} + p_{12} + p_{13} + p_{14} + p_{15} + p_{16} + p_{22} + p_{26} + p_{27} + p_{28}$. There are 286 nodes in the reachability graph and the size of the BRG is 121. Part of the evolutions in the BRG of Fig. 9 is shown in Table 5. The ILP is established to compute weakly illegal markings.

$$\left\{\begin{array}{l}\left\{\begin{array}{l}M + C_u \cdot y_{u1} \geq 0\\ W^T(M + C_u \cdot y_{u1}) > 1\\ y_{u1} \geq 0\end{array}\right.\\ \left\{\begin{array}{l}M + C_u \cdot y_{u2} \geq 0\\ y_{u2} \geq 0\end{array}\right.\\ \left\{\begin{array}{l}(M + C_u \cdot y_{u2}) + C_u \cdot y_{u3} \geq 0\\ y_{u3} \geq 0\end{array}\right.\end{array}\right.$$

Three weakly illegal markings reach five illegal markings, as detailed in Table 6. One deadlock and four token-overflow markings exist in $R(N, M_0)$. For different weakly illegal markings, the settings of transition priority are different. Given an observation $w = adcdadc$, the basis marking $M = [0,1,0,0,0,0,1,1,0,0,1,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^{\mathrm{T}}$ can be reached. Since t_{11}, t_3, and t_{12} are enabled simultaneously at M, and the firing of t_{11} leads the system to a weakly illegal marking, $r_{3,11} = 1$, $r_{12,11} = 1$. The current transition priority matrix P_c is expressed as:

$$P_c = \begin{bmatrix} 0 & \varphi & \varphi & \cdots & \varphi & \varphi \\ \varphi & 0 & \varphi & \cdots & \varphi & \varphi \\ \varphi & \varphi & 0 & \cdots & 1 & \varphi \\ \vdots & & & \ddots & & \vdots \\ \varphi & \varphi & -1 & \cdots & \varphi & -1 \\ \varphi & \varphi & \varphi & \cdots & 1 & 0 \end{bmatrix}_{12\times 12}$$

If additional control places are imposed to the plant net system, two control places and nine corresponding arcs are needed. In complex systems, a priority-based monitor has obvious advantages. It can be adjusted flexibly and does not

TABLE 5. Part of the evolutions in the BRG oF Fig. 9.

Index	Basis markings	Evolutions
M_0	$[2,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$	$M_0[\sigma_u t_1\rangle M_1$
M_1	$[1,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$	$M_1[\sigma_u t_9\rangle M_7$
M_7	$[1,1,0,0,1,0,0,0,0,0,1,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$	$M_7[\sigma_u t_{10}\rangle M_{17}$
M_{17}	$[1,1,0,0,0,1,0,0,0,0,1,1,1,0,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$	$M_{17}[\sigma_u t_{11}\rangle M_{30}$
M_{30}	$[1,1,0,0,0,0,1,0,0,0,1,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$	$M_{30}[\sigma_u t_1\rangle M_{41}$
M_{41}	$[0,1,1,0,0,0,1,0,0,0,0,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$	$M_{41}[\sigma_u t_9\rangle M_{56}$
M_{56}	$[0,1,0,0,1,0,0,1,0,0,1,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$	$M_{56}[\sigma_u t_{10}\rangle M_{70}$
M_{70}	$[0,1,0,0,0,1,0,1,0,0,1,1,1,0,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$	$M_{70}[\sigma_u t_{11}\rangle M_{84}$

TABLE 6. Weakly illegal markings and corresponding illegal markings.

Weakly illegal markings	Uncontrollable firing vectors	Illegal markings
$[0,1,1,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,1,0,0,0,0,1,1]^T$	$[1,1,0,1,0,0,0]^T$	$[0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,1,0,0,1,0,0,0]^T$
$[0,1,0,0,0,0,1,1,0,0,1,1,0,1,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$	$[0,1,0,0,0,0,1]^T$	$[0,0,0,0,0,0,0,2,0,0,1,1,1,1,1,1,0,0,0,0,0,1,0,0,1,1,1,0]^T$
	$[0,1,1,0,0,0,1]^T$	$[0,0,0,0,0,0,0,2,0,0,1,1,1,1,1,1,0,0,0,0,0,1,0,1,0,1,0,1]^T$
	$[0,0,0,0,0,0,1]^T$	$[0,1,0,0,0,0,0,2,0,0,1,1,1,1,1,1,0,0,0,0,0,1,0,0,0,1,1,1]^T$
$[0,0,0,0,0,0,1,1,0,0,1,1,0,1,1,1,0,0,0,0,0,1,1,0,0,0,1,1]^T$	$[0,0,0,0,0,0,1]^T$	$[0,0,0,0,0,0,0,2,0,0,1,1,1,1,1,1,0,0,0,0,0,1,1,0,0,0,1,1]^T$

complicate the net structure. It can ensure the maximally permissive behavior of the system, and it is easy to implement online control.

VIII. CONCLUSION AND FUTURE WORK

This paper addresses a control strategy based on transition priority in labeled Petri nets. First, we show that the notions of the BRG, deadlocks and GMECs can be used to identify weakly illegal markings with the help of ILP. In perspective of computational and space complexity, the proposed approach has great advantages. When token-overflow markings or deadlocks exist in an LPN, the corresponding ILP is proposed to identify weakly illegal markings. Second, under certain assumptions, we show that static and dynamic control algorithms based on the transition priority matrix can be used to prevent the system from entering illegal states. As the proposed method does not increase control places, it does not complicate the system structure. Our future research along this line will focus on dispatch in an FMS and fault detection for an LPN based on priority.

REFERENCES

[1] A. Giua, "Petri nets as discrete event models for supervisory control," Ph.D. dissertation, Dept. Comput. Sci., Rensselaer Polytechnic Inst., Troy, NY, USA, 1992.

[2] P. J. G. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. IEEE*, vol. 77, no. 1, pp. 81–98, Jan. 1989.

[3] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes," *SIAM J. Control Optim.*, vol. 25, no. 1, pp. 206–230, Jul. 1987.

[4] P. J. Ramadge and W. M. Wonham, "Modular feedback logic for discrete event systems," *SIAM J. Control Optim.*, vol. 25, no. 5, pp. 1202–1218, Sep. 1987.

[5] X. Wang, I. Khemaissia, M. Khalgui, Z. Li, O. Mosbahi, and M. Zhou, "Dynamic low-power reconfiguration of real-time systems with periodic and probabilistic tasks," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 258–271, Jan. 2015.

[6] X. Wang, Z. Li, and W. Wonham, "Dynamic multiple-period reconfiguration of real-time scheduling based on timed DES supervisory control," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 101–111, Feb. 2016.

[7] J. Zhang, M. Khalgui, Z. Li, G. Frey, O. Mosbahi, and H. Ben Salah, "Reconfigurable coordination of distributed discrete event control systems," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 323–330, Jan. 2015.

[8] R. C. Holt, "Some deadlock properties of computer systems," *ACM Comput. Surveys*, vol. 4, no. 3, pp. 179–196, Sep. 1972.

[9] M. A. Lawley, S. A. Reveliotis, and P. M. Ferreira, "A correct and scalable deadlock avoidance policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 14, no. 5, pp. 796–809, Oct. 1998, doi: 10.1109/70.720355.

[10] M. C. Zhou and K. Venkatesh, "Modeling FMS with Petri nets," in *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*, Singapore: World Scientific, 1998, ch. 5, pp. 91–146.

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

[12] A. Ichikawa and K. Hiraishi, "Analysis and control of discrete event systems represented by Petri nets," *Discrete Event Syst., Model Appl.*, vol. 103, pp. 115–134, Jan. 2006.

[13] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion contraints on nets with uncontrollable transitions," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 1992, pp. 974–979.

[14] 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, 1996.

[15] M. Uzam and M. Zhou, "Iterative synthesis of Petri net based deadlock prevention policy for flexible manufacturing systems," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 5. Hague, The Netherlands, Oct. 2004, pp. 4260–4265.

[16] 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, doi: 10.1109/TSMCA.2007.893484.

[17] J. O. Moody and P. J. Antsaklis, "Petri net supervisors for DES with uncontrollable and unobservable transitions," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 462–476, Mar. 2000.

[18] F. Basile, P. Chiacchio, and A. Giua, "Suboptimal supervisory control of Petri nets in presence of uncontrollable transitions via monitor places," *Automatica*, vol. 42, no. 6, pp. 995–1004, 2006.

[19] 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.

[20] Z. W. Li and M. C. 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, 2006, doi: 10.1109/TII.2006.885185.
[21] Z. Li and M. Zhao, "On controllability of dependent siphons for deadlock prevention in generalized Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 2, pp. 369–384, Mar. 2008.
[22] 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.
[23] Z. Li, G. Liu, 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] Y. Feng, K. Xing, M. Zhou, and H. Liu, "Liveness analysis and deadlock control for automated manufacturing systems with multiple resource requirements," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 2, pp. 525–538, Feb. 2020, doi: 10.1109/TSMC.2017.2767902.
[25] 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.
[26] 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.
[27] M. P. Cabasino, A. Giua, and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," *Automatica*, vol. 46, no. 9, pp. 1531–1539, Sep. 2010.
[28] M. P. Cabasino, C. N. Hadjicostis, and C. Seatzu, "Probabilistic marking estimation in labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 528–533, Feb. 2015.
[29] K. Barkaoui and J. F. Pradat-Peyre, "On liveness and controlled siphons in Petri nets," in *Proc. 17th Int. Conf. Appl. Theory Petri Nets*, vol. 1091. Berlin, Germany: Springer, Jun. 1996, pp. 57–72.
[30] 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.
[31] 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.
[32] M. Bashir, D. Liu, M. Uzam, N. 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, Jan. 2018, Art. no. 168781401775070.
[33] 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, Jul. 2018, Art. no. 168781401878740.
[34] 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.
[35] C. Haoxun, "Net structure and control logic synthesis of controlled Petri nets," *IEEE Trans. Autom. Control*, vol. 43, no. 10, pp. 1446–1450, Oct. 1998.
[36] J. Luo, H. Ni, W. Wu, S. Wang, and M. Zhou, "Simultaneous reduction of Petri nets and linear constraints for efficient supervisor synthesis," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 88–103, Jan. 2015.
[37] M. Hack, "Petri net languages," MIT, Cambridge, MA, USA, Tech. Rep. 159, Mar. 1976.
[38] E. Best and M. Koutny, "Petri net semantics of priority systems," *Theor. Comput. Sci.*, vol. 96, no. 1, pp. 175–215, Apr. 1992.
[39] F. Bause, "Analysis of Petri nets with a dynamic priority method," in *Proc. 18th Int. Conf. Appl. Theory Petri Nets*, vol. 1248. Toulouse, France, Jun. 1997, pp. 215–234.
[40] H.-C. Yen, "Priority conflict-free Petri nets," *Acta Inf.*, vol. 35, no. 8, pp. 673–688, Aug. 1998.
[41] C. G. Gassandras and S. Lafortune, "Petri nets," in *Introduction to Discrete Event Systems*, 2nd ed. New York, NY, USA: Springer, 2008, ch. 4, pp. 223–267.
[42] Z. W. Li and M. C. Zhou, "Monitor implementation of GMECs," in *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. London, U.K.: Springer, 2009, ch. 4, pp. 77–103.
[43] M. P. Cabasino, A. Giua, M. Pocci, and C. Seatzu, "Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems," *Control Eng. Pract.*, vol. 19, no. 9, pp. 989–1001, Sep. 2011.
[44] Z. Ma, Y. Tong, Z. Li, and A. Giua, "Basis marking representation of Petri net reachability spaces and its application to the reachability problem," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1078–1093, Mar. 2017.

YEJIA LIU (Graduate Student Member, IEEE) received the B.S. degree from Xiangtan University, Xiangtan, China, in 2016. She is currently pursuing the Ph.D. degree in mechanical engineering with the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu, China. Her current research interests include Petri nets, supervisory control, and fault diagnosis in discrete event systems.

XUNBO LI was born in Chengdu, China, in 1963. He received the B.S. degree in mechanical engineering, the M.S. degree in electronic engineering, and the Ph.D. degree in automatic control from the University of Electronic Science and Technology of China, Chengdu, China, in 1985, 1991, and 2001, respectively. He was a Visiting Professor with Victoria University, Melbourne, Australia, in 2006. He is currently a Professor with the School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China. Over the past decade, he has focused on research in intelligent industrial manufacturing. His current research interests include cyber-physical systems, machine vision, wireless sensor networks, intelligent scheduling, and fault diagnosis in flexible manufacturing systems.

ZHIWU LI (Fellow, IEEE) received the B.S. degree in mechanical engineering, the M.S. degree in automatic control, and the Ph.D. degree in manufacturing engineering from Xidian University, in 1989, 1992, and 1995, respectively.

Over the past decade, he held a visiting professorship with the University of Toronto, the Technion—Israel Institute of Technology, Martin-Luther University, King Saud University, the University of Cagliari, and Meliksah Universitesi. He is currently with the Institute of Systems Engineering, Macau University of Science and Technology, Macau. His current research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, system reconfiguration, game theory, and data and process mining. He was a member of the IFAC Technical Committee on Discrete Event and Hybrid Systems, from 2011 to 2014. He was a recipient of the Alexander von Humboldt Research Grant, Alexander von Humboldt Foundation, Germany. He is the Founding Chair of the Xi'an Chapter of the IEEE Systems, Man, and Cybernetics Society. He chairs the Discrete Event Systems Technical Committee of the IEEE Systems, Man, and Cybernetics Society. He is a frequent Reviewer of more than 100 international journals, including *Automatica* and a number of the IEEE TRANSACTIONS as well as many international conferences. He is listed in Marquis Who's Who in the World, 27th Edition, in 2010.