Hindawi
Journal of Control Science and Engineering
Volume 2019, Article ID 2895685, 11 pages
https://doi.org/10.1155/2019/2895685

Research Article

Model Identification of Unobservable Behavior of Discrete Event Systems Using Petri Nets

Guanghui Zhu,[1,2] Ya Wang ,[1] and Yajie Wang[3]

[1] *School of Information Engineering, Xuchang University, Xuchang 461000, China*
[2] *School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China*
[3] *School of Mathematics and Statistics, North China University of Water Resources and Electric Power, Zhengzhou 450046, China*

Correspondence should be addressed to Ya Wang; wangya302@xcu.edu.cn

Received 19 November 2018; Accepted 3 March 2019; Published 27 March 2019

Academic Editor: Paolo Mercorelli

This paper deals with the problem of identifying a Petri net that models the unobservable behavior of a system from the knowledge of its dynamical evolution. We assume that a partial Petri net model that represents the observable behavior of a system is given in which all the transitions are observable. An identifier monitors the system evolution and records the observed transition sequence (and possible corresponding markings). Some unobservable transitions modeling the unknown system behavior are identified from the transition sequence by formulating and solving integer linear programming problems. These identified unobservable transitions together with the given partial Petri net model characterize the whole system, including observable and unobservable behavior. Two different cases are considered. First, we assume that no place is observable. In such a case, a transition sequence is observed only during the evolution of the system. Second, we assume that a subset of places is observable; i.e., the observation contains not only the transition sequence but the corresponding markings as well. Hence an additional constraint should be imposed on the unobservable transition in the related programming problems according to the observed markings such that a more authentic unobservable transition can be found.

1. Introduction

Dynamic systems can always be modeled as continuous time systems or discrete event systems (DESs) [1]. In the framework of DESs, the dynamic behavior of a system is usually characterized by its output in the form of numbers, symbols, text, etc. An interesting issue is to infer a system model from the knowledge of system's output information such that its behavior can be verified, controlled [2, 3], or diagnosed [4–7]. The identification of a DES consists in determining a mathematical model from the data set of input-output signals of a system that describes all or part of the system behavior.

In the DES context, the model setting for the identification problem mainly includes automata and Petri nets. In the early work, Gold [8] shows that the problem of whether there exists an automaton that agrees with the given data is NP-complete. On the basis of an automaton model, Angluin [9] presents a family of efficient algorithms for inferring the k-reversible languages from finite positive samples.

Given some information on the evolution of a system, several studies discuss the identification problem of DES using Petri nets. Roughly speaking, the existing approaches can be divided into three categories: (a) given a finite language describing the system behavior, it consists in determining the structure and initial marking of a Petri net system whose language coincides with the given one [10–12], (b) given an event sequence (transition sequence) as well as the corresponding system states (markings), the goal is to determine a Petri net system such that the given event sequence is firable from its initial marking [13–15], and (c) assuming that the net to be identified has special structure (e.g., safe Petri nets) or properties (e.g., exhibiting cyclic behavior), the net is identified from finite data using the customized algorithm [16–19]. In general, an integer linear programming problem (ILPP) is defined to impose certain

constraints on the structure of the net to be identified in cases (a) and (b). It may require heavy computational costs to solve an ILPP; however this technique is very general. On the other hand, a customized algorithm is designed based on the knowledge of the structure and property of the net in case (c), and the algorithm often has polynomial time complexity. Nevertheless, it can only be applied to a special class of Petri nets.

Giua and Seatzu [10] present an integer programming technique for the identification of a free-labeled net, i.e., a net without an associated labelling function, from a finite prefix-closed language with the length of its longest word less than or equal to a given positive integer. In their approach, both positive examples and counterexamples can be derived from the given language, and the constraint set of the programming problem allows the positive examples but forbids the counterexamples. The approach is general, but it has high computational complexity. Cabasino et al. [11] extend this approach to the identification of a *deterministic* λ-free-labeled Petri net, i.e., a net where two transitions can share the same label (cannot be an empty string).

Unlike the studies in [10, 11] in which no information about places is known, Dotoli et al. [13] assume that a transition sequence and the corresponding reached marking after transition fires are observed. A net system is identified by solving an ILPP such that the transition sequence is firable from its initial marking and each marking after firing a transition in the sequence equals the observed one.

Hiraishi [16] proposes an approach which constructs a safe Petri net from the knowledge of firing sequences. Firstly, a finite state automaton is identified from the given firing sequences, and then the structure of the Petri net is estimated. The approach has polynomial time complexity, but it can only be applied to safe Petri nets. Bekrar et al. [19] provide an algorithm that identifies a safe Petri net from the data set of I/O signals of a system. They assume that the considered system is cyclic and all its cycles start from the same initial state. Meda et al. in [17, 18] discuss the identification problem by using Interpreted Petri nets, i.e., a Petri net associated with an input function and an output function, assuming that the dependency between transitions is known. All studies in [16–19] fall into case (c). These approaches are not general and can only be applied to Petri nets with special structure or behavior.

The problems of fault diagnosis and diagnosability have received a lot of attention since the work by Sampath et al. [20]. In the context of Petri nets, it is usually assumed that the faulty model describing not only the fault-free but faulty behavior of a system is known and that the fault events are modeled by unobservable transitions [4–7]. In other words, when we build the Petri net model of a system for diagnosis, both fault-free system behavior and faulty behavior should be considered simultaneously. However, it is not realistic to figure out all possible faulty behavior when one models a system [21]. Thus an issue arises: how to identify faulty (unobservable) behavior to improve the accuracy of the faulty model. In this issue, the fault-free model is assumed to be known. We have to decide whether some unobservable behaviors occur according to the observed event sequence. To the best of our knowledge, there are few studies focusing on the identification of unobservable transitions using Petri nets.

Cabasino et al. [12] propose an approach based on the work in [10] to identify the faulty (unobservable) transitions by assuming that the fault-free model is known and that the system contains at most one unobservable loop-free transition. The main idea is to define and solve an integer programming problem according to the positive examples and counterexamples that can be computed by comparing the given language and the observed language. The main drawbacks of this approach are the computational complexity and the limitation that only one unobservable transition can be identified.

Dotoli et al. [15] also assume that the nominal model without unobservable transitions is known. An online approach is proposed to identify recursively an unobservable subnet from the knowledge of an observed transition sequence and the corresponding reached markings.

In this paper, we extend the work in [15], where all places are assumed to be observable, to the case of partial observable places. Our work is motivated by the fact that in the practical application it is difficult to associate each place with a sensor because of the technical or financial consideration.

This paper deals with the identification of unobservable transitions from the knowledge of an observed transition sequence. An identifier monitors the system and, upon the observation of a transition, an integer programming problem is solved to decide whether an unobservable transition should be introduced. If it is true, an unobservable transition is identified by solving another integer programming problem. An online algorithm is also presented to identify an unobservable subnet consisting of identified unobservable transitions.

In contrast to the work [15], the proposed approach is more general since only part of places need to be observable (all places are assumed to be observable in [15]), and they have the same computational complexity. More specifically, the number τ of observable places in our approach satisfies $0 \leq \tau \leq m$, where m is the number of places in a net. For $\tau = 0$, i.e., no place is observable, the observed output is a transition sequence. For $0 < \tau < m$, i.e., a subset of places is observable, the observation is a transition-partial-marking sequence, where a *partial marking* is a marking of observable places. If $\tau = m$, then the problem to solve is the same as [15].

This paper is organized as follows. Section 2 presents basic definitions about Petri nets as well as some preliminary results. In Section 3, we first present some results that are theoretical basis of the subsequent identification algorithm, and then an online algorithm and its computational complexity are discussed. In Section 4, we extend the algorithm to the case that some of places are observable. Finally, we conclude this paper with the further work in Section 5.

2. Preliminary

This section recalls basic concepts of Petri nets and some preliminary results. In addition, several definitions used in this paper are given. For more details on Petri nets, the reader is referred to [1, 22].

2.1. Basics of Petri Nets. A Petri net is a four-tuple $PN = (P, T, Pre, Post)$, where P is the set of places, T is the set of transitions, $Pre : P \times T \longrightarrow \mathbb{N}$, and $Post : P \times T \longrightarrow \mathbb{N}$ are the pre- and postincidence matrices, respectively, which specify the structure of the Petri net. For $p \in P$, $t \in T$, and $x \in \mathbb{N}$, $Pre(p,t) = x$ if there exists an arc with weight x from p to t, 0 otherwise; $Post(p,t) = x$ if an arc with weight x goes from t to p, 0 otherwise. Here $\mathbb{N}$ is the set of nonnegative integers. We denote by $m = |P|$ the number of places and $n = |T|$ that of transitions. The incidence matrix of a net is denoted as $C = Post - Pre$. For $t \in T$, its *preset* is defined as ${}^{\bullet}t = \{p \in P \mid Pre(p,t) > 0\}$, and its *postset* is defined as $t^{\bullet} = \{p \in P \mid Post(p,t) > 0\}$. A transition t is said to be a *source transition* if it satisfies ${}^{\bullet}t = \emptyset$. A marking of a Petri net is a vector $M : P \longrightarrow \mathbb{N}$, and $M(p)$ indicates the number of tokens in place p. A net system $\langle PN, M_0 \rangle$ is a Petri net with an initial marking M_0.

A transition $t_j \in T$ is *enabled* at marking M if $\forall p \in {}^{\bullet}t_j, M(p) \geq Pre(p, t_j)$, which is denoted by $M[t_j\rangle$. An enabled transition t_j at M can fire yielding a new marking M' such that $M' = M + Post(\cdot, t_j) - Pre(\cdot, t_j)$, which is denoted as $M[t_j\rangle M'$. If $M[t_j\rangle M'$ holds, by the definition of incidence matrix C, we have $M' = M + C(\cdot, t_j)$, which can be rewritten as $M' = M + C \cdot \vec{t}_j$, where $\vec{t}_j$ is the n-dimensional canonical basic vector and its j-th entry is one.

Given a transition sequence σ and a marking M, $M[\sigma\rangle$ denotes that σ is enabled at marking M and $M[\sigma\rangle M'$ denotes that a new marking M' is *reachable* from M after firing σ. All the markings reachable from M_0 comprise the *reachability set* of a Petri net and it is denoted as $R(PN, M_0)$. The set of sequences enabled at the initial marking M_0 is denoted as

$$L(PN, M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}, \tag{1}$$

which is called the *language* of a Petri net.

We define an n-dimensional column vector $\vec{\sigma} : T \longrightarrow \mathbb{N}$ (called *firing vector*) for the corresponding transition sequence σ such that $\vec{\sigma}(t) = k$ if transition t occurs k times in σ. For an enabled transition sequence σ at M_0 such that $M_0[\sigma\rangle M$, we have the following state equation:

$$M = M_0 + C \cdot \vec{\sigma}, \tag{2}$$

which is called the *state equation* of a Petri net system.

2.2. Labeled Petri Net. Given a Petri net $PN = (P, T, Pre, Post)$ and the set of events E, a labeling function $\lambda : T \longrightarrow E \cup \{\varepsilon\}$ assigns to each transition $t \in T$ either a symbol from the event set E or an empty string ε. A transition with label ε is called an *unobservable transition* and the set of unobservable transitions is denoted as $T_u = \{t \in T \mid \lambda(t) = \varepsilon\}$ with $n_u = |T_u|$ being the cardinality of set T_u. All other transitions whose labels are not ε comprise the set of *observable transitions* $T_o = \{t \in T \mid \lambda(t) \neq \varepsilon\}$ and $n_o = |T_o|$. Thus the set T is divided into two disjoint subsets T_o and T_u with $T = T_o \cup T_u$. We assume that each label $e \in E$ can only be assigned to one transition, i.e., the set E is isomorphic to the set of observable transitions T_o, and thus without loss of generality we assume $E = T_o$. A place $p \in P$ is said to be *observable* if it is equipped with a sensor that allows observation of the number of its tokens.

For each $\sigma_o \in T_o^*$, we define an n_o-dimensional vector $\vec{\sigma}_o : T_o \longrightarrow \mathbb{N}$ and $\vec{\sigma}_o(t) = k$ if $t \in T_o$ occurs k times in σ_o. Analogously, for each $\sigma_u \in T_u^*$, an n_u-dimensional vector $\vec{\sigma}_u : T_u \longrightarrow \mathbb{N}$ is defined and $\vec{\sigma}_u(t) = k$ if $t \in T_u$ occurs k times in σ_u.

2.3. Linear Reformulation. An optimization problem with an absolute value is nonlinear. Absolute value functions are not continuously differentiable, and it is difficult to perform standard optimization procedure on them. However, we can convert an absolute value function into several linear constraints such that the optimization can be solved using the standard linear programming technique. We here only consider this situation: there is an absolute value in the objective function. Let us consider the following optimization problem:

$$\begin{aligned} \min \quad & \vec{a}^T \cdot \operatorname{abs}(\vec{x}) + \vec{b}^T \cdot \vec{y} \\ \text{s.t.} \quad & A \end{aligned} \tag{3}$$

where $\vec{x}$ and $\vec{y}$ are variables in $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, $\vec{a}$ and $\vec{b}$ are constant vectors in $(\mathbb{R}_{\geq 0})^n$ and $\mathbb{R}^m$, respectively, $\operatorname{abs}(\vec{x})$ denotes the absolute value of vector $\vec{x}$, and A denotes the linear constraints in this programming problem. Note that $\vec{a}$ is a column vector consisting of nonnegative real numbers.

Programming problem (3) can be rewritten in the form of a linear objective function as

$$\begin{aligned} \min \quad & \vec{a}^T \cdot \vec{z} + \vec{b}^T \cdot \vec{y} \\ \text{s.t.} \quad & A \\ & -\vec{x} \leq \vec{z} \\ & \vec{x} \leq \vec{z} \\ & \vec{z} \geq \vec{0} \end{aligned} \tag{4}$$

where $\vec{z}$ is a variable in $\mathbb{R}^n$; i.e., we add a variable $\vec{z} \geq \vec{0}$ and two additional constraints $-\vec{x} \leq \vec{z}$ and $\vec{x} \leq \vec{z}$.

In fact, for the real numbers x_1 and z_1 that are the first entries of vectors $\vec{x}$ and $\vec{z}$, respectively, there are three different cases.

(1) $x_1 > 0$. Constraint $-x_1 \leq z_1$ is always fulfilled. Because of the constraint $x_1 \leq z_1$, z_1 is necessarily at least as large as x_1. On the other hand, z_1 is in the objective function such that it trends to be as small as possible. Thus z_1 will be equal to x_1, i.e., $z_1 = \operatorname{abs}(x_1)$.

(2) $x_1 = 0$. In this case, constraints $-x_1 \leq z_1$ and $x_1 \leq z_1$ are always verified because of $z_1 \geq 0$. Moreover z_1 is in the objective function, and thus it trends to be zero, i.e., $z_1 = \operatorname{abs}(x_1)$.

(3) $x_1 < 0$. Being analogous to case (1), $z_1 = \text{abs}(x_1)$ is trivially verified.

Definition 1. Given a Petri net $PN = (P, T, Pre, Post)$ with $T = T_o \cup T_u$, PN can be partitioned into two components denoted by $PN = PN_o \otimes PN_u$. Here $PN_o = (P, T_o, Pre_o, Post_o)$ is the *observable subnet*, where P is the set of places, T_o is the set of observable transitions, and Pre_o and $Post_o$ are the restrictions of Pre and $Post$ to T_o, respectively; $PN_u = (P, T_u, Pre_u, Post_u)$ is the *unobservable subnet*, where T_u is the set of unobservable transitions and Pre_u and $Post_u$ are the restrictions of Pre and $Post$ to T_u, respectively.

Definition 2 (see [5]). Considering the net $PN = (P, T, Pre, Post)$ with $T = T_o \cup T_u$, we define the following two projection operations:

(i) $Q_o : T^* \longrightarrow T_o^*$, which is defined as (i) $Q_o(\varepsilon) = \varepsilon$; (ii) for all $\sigma \in T^*$ and $t \in T$, $Q_o(\sigma t) = Q_o(\sigma)t$ if $t \in T_o$, and $Q_o(\sigma t) = Q_o(\sigma)$ otherwise, where ε is the empty string.

(ii) $Q_u : T^* \longrightarrow T_u^*$, which is defined as (i) $Q_u(\varepsilon) = \varepsilon$; (ii) for all $\sigma \in T^*$ and $t \in T$, $Q_u(\sigma t) = Q_u(\sigma)t$ if $t \in T_u$, and $Q_u(\sigma t) = Q_u(\sigma)$ otherwise.

In plain words, given a transition sequence $\sigma \in T^*$, the operator Q_o computes the projection of σ on observable transition set T_o and the operator Q_u computes the projection of σ on unobservable transition set T_u. Note that Q_o^{-1} denotes the inverse of projection Q_o.

Given a sequence $\sigma \in L(PN, M_0)$, $w = Q_o(\sigma)$ is the observed sequence. The set of all observed sequences can be represented by

$$\begin{aligned}&\Omega(PN, M_0)\\&\quad = \{w \in T_o^* \mid \sigma \in L(PN, M_0), w = Q_o(\sigma)\}.\end{aligned} \tag{5}$$

A Petri net is said to be *acyclic* if it has no directed circuit. Equation (2) shows that there exists a nonnegative integer $\vec{x}$ such that $M = M_0 + C \cdot \vec{x}$ if M is reachable from M_0, which is a necessary but not sufficient condition. However, for an acyclic Petri net, it is necessary and sufficient, as shown by the following theorem.

Theorem 3 (see [22]). *In an acyclic Petri net, a marking $M \geq 0$ is reachable from M_0 if and only if (iff) there exists a nonnegative integer solution $\vec{x}$ satisfying $M = M_0 + C \cdot \vec{x}$.*

A net $PN = (P, T, Pre, Post)$ is *pure* if it has no self-loop; namely, $\forall t \in T$, $Pre(p, t) \times Post(p, t) = 0$ for each $p \in P$. Note that the structure of a pure net can be represented by its incidence matrix. Moreover, it is trivially verified that an acyclic net is pure.

Definition 4. Consider net system $\langle PN, M_0\rangle$ with $T = T_o \cup T_u$. Let $w \in \Omega(PN, M_0)$ be an observed sequence. The sequences of unobservable transitions which enable the occurrence of w are denoted by

$$\Sigma(w) = \left\{y \in T_u^* \mid y = Q_u\left(Q_o^{-1}(w)\right)\right\} \tag{6}$$

and the n_u-dimensional firing vectors of sequences of unobservable transitions associated with w are denoted by

$$Y(w) = \left\{\vec{y} \in \mathbb{N}^{n_u} \mid y \in \Sigma(w)\right\}. \tag{7}$$

Example 5. Figure 1 shows a net system with $T_o = \{t_1, t_2\}$, $T_u = \{\varepsilon_1, \varepsilon_2, \varepsilon_3\}$, and $M_0 = [1, 0, 0, 0]^T$. If an observed transition sequence is $w = t_1 t_2$, by Definition 4, we observe $Q_o^{-1}(w) = \{t_1\varepsilon_1\varepsilon_2 t_2, t_1\varepsilon_3 t_2\}$. Thus $\Sigma(w) = \{\varepsilon_1\varepsilon_2, \varepsilon_3\}$ and $Y(w) = \{[1, 1, 0]^T, [0, 0, 1]^T\}$.

3. Model Identification without Observable Places

3.1. Problem Statement. Assume that the observable subnet PN_o and the initial marking M_0 are known. We monitor the system evolution and record the occurred event. Upon the occurrence of an event, a transition $t \in T_o$ is observed and it is appended to the tail of the observed sequence. If the observed transition sequence is v, then two different situations are taken into account.

(i) $M_0[v\rangle$ holds, i.e., $v \in L(PN_o, M_0)$. In such a case the observable subnet is sufficient to describe the observed transition sequence.

(ii) $M_0[v\rangle$ does not hold. We infer that one or more unobservable transitions not included in PN_o occur.

We can infer a net $PN = (P, T, Pre, Post)$ with $PN = PN_o \otimes PN_u$ from the observed transition sequence v such that it can describe the observable and unobservable system behavior. To correctly identify the unobservable subnet, we make the following assumptions.

(A1) The unobservable behavior of a system can be characterized by an acyclic subnet, denoted by $PN_u = (P, T_u, Pre_u, Post_u)$.

Given an observable subnet PN_o that models the observable behavior of a system, Assumption (A1) means that we can always find a solution of an acyclic unobservable subnet PN_u such that the combined net $PN = PN_o \otimes PN_u$ describes the whole system. On the other hand, Assumption (A1) provides a necessary and sufficient condition for the reachability of the unobservable subnet PN_u.

To combine the known observable subnet PN_o and the identified unobservable subnet PN_u, the preincidence matrix of the whole net PN is computed by $Pre = [Pre_o \;\; Pre_u]$, and its postincidence matrix is represented by $Post = [Post_o \;\; Post_u]$. Correspondingly, we have $C = [C_o \;\; C_u]$.

Problem 6. Assume that an observable subset $PN_o = (P, T_o, Pre_o, Post_o)$ and the initial marking M_0 are given and that the observed transition sequence is $w \in T_o^*$. Under Assumption (A1), the problem consists in finding a Petri net $PN = (P, T, Pre, Post)$ such that $PN = PN_o \otimes PN_u$ and $w \in \Omega(PN, M_0)$.

The structure of a pure net can be represented by its incidence matrix. Thus, by Assumption (A1), we only need to compute the incidence matrix C_u of the unobservable

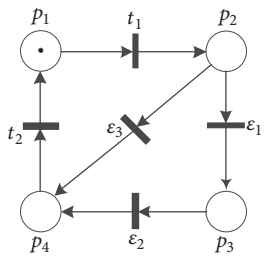

Figure 1: A Petri net system.

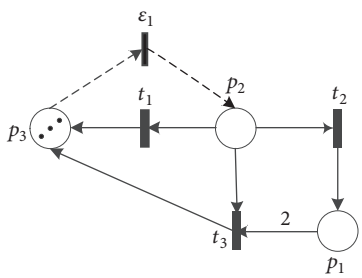

Figure 2: A Petri net with $T_u = \{\varepsilon_1\}$.

subnet PN_u to determine its structure. On the other hand, the observable subset PN_o is known, and thus we can obtain the net PN.

3.2. Main Results. Assume that the observable subnet PN_o and M_0 are given and that a sequence v has been observed. A Petri net PN is identified such that $PN = PN_o \otimes PN_u$ and $v \in \Omega(PN, M_0)$. Now a new transition $t \in T_o$ is observed. We can decide whether $vt \in \Omega(PN, M_0)$ holds thanks to the following theorem, where vt denotes the observed sequence after transition t is recorded.

Theorem 7. *Assume that Assumption (A1) is satisfied in a Petri net PN with $PN = PN_o \otimes PN_u$. Let $v \in \Omega(PN, M_0)$ be the current observed sequence and $\vec{y} \in Y(v)$ be a corresponding firing vector of unobservable transitions. A new transition t is observed. If (8) admits a solution, then $vt \in \Omega(PN, M_0)$.*

$$\begin{aligned} M_0 + C_o \cdot \vec{v} + C_u \cdot \vec{y} + C_u \cdot \vec{\sigma}_u &\geq Pre_o(\cdot, t) \\ \vec{\sigma}_u &\in \mathbb{N}^{n_u} \end{aligned} \tag{8}$$

Proof. Let $M' = M_0 + C_o \cdot \vec{v} + C_u \cdot \vec{y}$ be a marking of PN after sequence v fires. Clearly, $M' \in R(PN, M_0)$ holds. If (8) admits a solution, then there exists an nonnegative integer vector $\vec{\sigma}_u$ such that $M' + C_u \cdot \vec{\sigma}_u \geq Pre_o(\cdot, t) \geq \vec{0}$. Let $M'' = M' + C_u \cdot \vec{\sigma}_u$. By Assumption (A1) and Theorem 3, there exists an unobservable transition sequence σ_u such that $M'' \in R(PN, M_0)$ and $M'[\sigma_u\rangle M''$. Since $M'' \geq Pre_o(\cdot, t)$, $M'[\sigma_u\rangle M''[t\rangle$ holds. Thus the transition t can be observed after v by firing σ_u, i.e., $vt \in \Omega(PN, M_0)$. □

The solution to (8) is not unique, and we can choose a performance index to select a solution among the set of admissible ones such that it is most suitable for the current situation. In particular, we can consider the performance index $\vec{1}_{n_u}^T \cdot \vec{\sigma}_u$ which minimizes the sum of the firing times of unobservable transitions. We define the following ILPP:

$$\text{ILPP 1}: \begin{cases} \min & \vec{1}_{n_u}^T \cdot \vec{\sigma}_u \\ \text{s.t.} & Eq.\ (8) \end{cases} \tag{9}$$

where $\vec{1}_{n_u}^T$ is a row vector of dimension n_u with each element being 1.

Example 8. Consider the net system in Figure 2, where $P = \{p_1, p_2, p_3\}$, $T_o = \{t_1, t_2, t_3\}$, $T_u = \{\varepsilon_1\}$, and the dashed lines represent the arcs connected with unobservable transitions. Assume that the observed sequence is $v = t_1$ and that the corresponding firing vector of unobservable transitions is $\vec{y} = [1]^T$. If the transition t_2 is now observed, we find a solution $\vec{\sigma}_u = [1]^T$ by solving ILPP 1. According to Theorem 7, we have $v = t_1 t_2 \in \Omega(PN, M_0)$. In fact, there exists a firing sequence $\sigma = \varepsilon_1 t_1 \varepsilon_1 t_2$ such that $M_0[\sigma\rangle$.

If (8) has no feasible solution, we can only add a new unobservable transition to Petri net PN to obtain a new Petri net PN' such that $vt \in \Omega(PN', M_0)$. Assume that ε_z is the considered unobservable transition. Let $PN' = PN_o \otimes PN'_u$; i.e., Petri nets PN and PN' have the same observable subnet PN_o; however the unobservable subnet PN'_u is obtained by adding the unobservable transition ε_z to PN_u. By Assumption (A1), the structure of PN_u can be determined by its incidence matrix C_u. Assume that $\vec{\varepsilon}_z$ is the column in the incidence matrix associated with ε_z and then the incidence matrix of PN'_u is $C'_u = [C_u \ \vec{\varepsilon}_z]$; i.e., PN'_u retains the structure of PN_u but includes an additional transition ε_z. The following proposition provides a method to compute the unobservable transition.

Proposition 9. *Given a net system $\langle PN, M_0\rangle$ with $PN = PN_o \otimes PN_u$, let $v \in \Omega(PN, M_0)$ be the current observed sequence and $\vec{y} \in Y(v)$ be a corresponding firing vector of unobservable transitions. Assume that a new transition t is observed. If there exists a vector $\vec{\varepsilon}_z$ satisfying (10), then $vt \in \Omega(PN', M_0)$, where PN' is a Petri net obtained by adding an unobservable transition ε_z with the corresponding incidence matrix column being $\vec{\varepsilon}_z$ to PN.*

$$\begin{aligned} M_0 + C_o \cdot \vec{v} + C_u \cdot \vec{y} + \vec{\varepsilon}_z &\geq Pre_o(\cdot, t) \\ \vec{\varepsilon}_z &\in \mathbb{Z}^m \\ \vec{\varepsilon}_z &\neq \vec{0} \end{aligned} \tag{10}$$

where $\mathbb{Z}$ is the set of integers.

Proof. Let $M' = M_0 + C_o \cdot \vec{v} + C_u \cdot \vec{y}$ be a marking of PN after firing sequence v. Obviously, $M' \in R(PN', M_0)$ holds. If $\vec{\varepsilon}_z$ is a solution of (10), then $M' + \vec{\varepsilon}_z \geq Pre_o(\cdot, t)$ holds. Let $M'' = M' + \vec{\varepsilon}_z$. Since $C'_u = [C_u \ \vec{\varepsilon}_z]$, i.e., $\vec{\varepsilon}_z$

is the last column of incidence matrix C'_u, by Theorem 3, $M' + C'_u(\cdot, \varepsilon_z) = M'' \geq Pre_o(\cdot, t)$ holds, where ε_z denotes the new unobservable transition whose incidence matrix column is $\vec{\varepsilon}_z$. Thus $M'[\varepsilon_z\rangle M''[t\rangle$ holds, i.e., $vt \in \Omega(PN', M_0)$. □

Let us consider (10) again. When the cardinality of set T_u is zero, i.e., $|T_u| = 0$, the incidence matrix C_u is a 0×0 matrix and we can set $C_u \cdot \vec{y} = \vec{0}$. On the other hand, there may be more than one solution satisfying the linear constraint set represented by (10), and we can define a performance index to select a solution among those solutions. In particular, if $f = \vec{1}_m^T \cdot \text{abs}(\vec{\varepsilon}_z)$ is considered as a performance index which minimizes the sum of weights of the arcs connected with the unobservable transition ε_z, the following integer programming problem is formulated:

$$\begin{aligned} &\min \quad \vec{1}_m^T \cdot \text{abs}\left(\vec{\varepsilon}_z\right) \\ &\text{s.t.} \quad Eq.\ (10) \\ &\qquad \left.\begin{aligned} S \in \{0,1\}^m \\ \vec{1}_m^T \cdot S \leq m-1 \\ -K \cdot S + \vec{\varepsilon}_z \leq -\vec{1}_m \end{aligned}\right\} (a) \end{aligned} \tag{11}$$

where K is a constant that is large enough.

If the performance index $f = \vec{1}_m^T \cdot \text{abs}(\vec{\varepsilon}_z)$ is chosen, the solution of (11) is always $\vec{\varepsilon}_z \geq 0$; namely, we can always obtain a source transition. In practical applications, because of the limited resources, a source transition that models unobservable behavior of a system is not common. In order to forbid the solution containing source transitions, we impose the additional constraints (a) on the unobservable transition to guarantee that *at least* one component of the solution $\vec{\varepsilon}_z$ is a negative integer; i.e., the solution cannot be a source transition. Here, that *at least* a component of a vector is less than or equal to zero can be characterized by the logical OR operator which can be easily converted into some linear constraints [23], such as constraints (a) in (11).

On the other hand, the optimization problem represented by (11) is nonlinear because of the absolute value in the objective function. We have shown in Section 2.3 that the absolute value in a programming problem can be removed by adding additional variables and constraints. Thus (11) can be rewritten as follows:

$$\text{ILPP 2} \begin{cases} \min \quad \vec{1}_m^T \cdot \vec{r} \\ \text{s.t.} \quad Eq.\ (10) \\ \qquad \left.\begin{aligned} S \in \{0,1\}^m \\ \vec{1}_m^T \cdot S \leq m-1 \\ -K \cdot S + \vec{\varepsilon}_z \leq -\vec{1}_m \end{aligned}\right\} (a) \\ \qquad \left.\begin{aligned} -\vec{\varepsilon}_z \leq \vec{r} \\ \vec{\varepsilon}_z \leq \vec{r} \\ \vec{r} \in \mathbb{N}^m \end{aligned}\right\} (b) \end{cases} \tag{12}$$

where $\vec{r}$ and (b) are, respectively, the additional variables and constraints associated with $\text{abs}(\vec{\varepsilon}_z)$ in (11).

Example 10. Consider the net system $\langle PN, M_0\rangle$ in Figure 3(a), where $T_o = \{t_1, t_2, t_3\}$ and $T_u = \{\varepsilon_1\}$. Assume that the observed sequence is $v = t_1t_2$ satisfying $v \in \Omega(PN, M_0)$ and that the corresponding firing vector is $\vec{y} = [2]^T$. Suppose that t_3 is now observed, but ILPP 1 has no feasible solution; i.e., we should add a new unobservable transition to PN such that $t_1t_2t_3 \in \Omega(PN, M_0)$. The solution $\vec{\varepsilon}_2 = [1, 1, -1]^T$ is found by solving ILPP 2. Let $C_u = [C_u, \vec{\varepsilon}_2]$ and update the structure of PN_u by adding a new unobservable transition ε_2. Figure 3(b) shows the obtained Petri net.

Note that the net system in Figure 3(c) is also a solution of Example 10. Moreover, it has a smaller sum of weights of the arcs connected with ε_2. In fact, we can make use of the previously obtained unobservable transitions when we compute the new unobservable transition to obtain a simpler solution. The following proposition provides a technique to make use of the previously identified unobservable transitions.

Proposition 11. *Assume that the net system $\langle PN, M_0\rangle$ is given and $v \in \Omega(PN, M_0)$ is the current observed sequence with $\vec{y} \in Y(v)$. A new transition t is observed. If there exists a vector $\vec{\varepsilon}_z$ satisfying; (13a), (13b), (13c), and (13d), then $vt \in \Omega(PN', M_0)$, where PN' is a Petri net obtained by adding ε_z to PN.*

$$M_0 + C_o \cdot \vec{v} + C_u \cdot \left(\vec{y} + \vec{\sigma}_u\right) + \vec{\varepsilon}_z \geq Pre_o(\cdot, t) \tag{13a}$$

$$\vec{\sigma}_u \in \mathbb{N}^{n_u} \tag{13b}$$

$$\vec{\varepsilon}_z \in \mathbb{Z}^m \tag{13c}$$

$$\vec{\varepsilon}_z \neq \vec{0} \tag{13d}$$

Proof. Let $M' = M_0 + C_o \cdot \vec{v} + C_u \cdot \vec{y}$ be the marking after firing sequence v. $M' \in R(PN', M_0)$ holds. The constraint (13a) can be rewritten as

$$M' + C_u \cdot \vec{\sigma}_u + \vec{\varepsilon}_z \geq Pre_o(\cdot, t). \tag{14}$$

Since $C'_u = [C_u \ \ \vec{\varepsilon}_z]$, (14) is rewritten as

$$M' + C'_u \cdot \begin{bmatrix} \vec{\sigma}_u \\ 1 \end{bmatrix} \geq Pre_o(\cdot, t). \tag{15}$$

Let $M'' = M' + C'_u \cdot [\vec{\sigma}_u \ \ 1]^T$. By Theorem 3, there exists an unobservable transition sequence σ'_u such that $M'[\sigma'_u\rangle M''$. Since $M'' \geq Pre_o(\cdot, t)$, $M'[\sigma'_u\rangle M''[t\rangle$ holds, i.e., $vt \in \Omega(PN', M_0)$. □

As shown in (10), the solution of (13a), (13b), (13c), and (13d) is neither unique. We still consider the performance index $f = \vec{1}_m^T \cdot \text{abs}(\vec{\varepsilon}_z)$ and convert the absolute value in

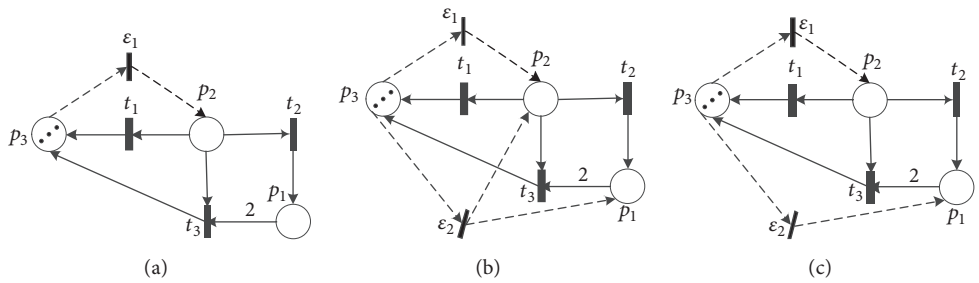

Figure 3: (a) A net satisfying $t_1t_2 \in \Omega(PN, M_0)$, (b) a net after ε_2 is added, and (c) a net with a simpler structure than (b).

objective function into linear constraints. The final integer linear programming problem is shown as follows:

$$\text{ILPP 3}: \begin{cases} \min & \vec{1}_m^T \cdot \vec{r} \\ \text{s.t.} & Eq.\,(13a), (13b), (13c), \text{and } (13d) \\ & \left.\begin{array}{c} S \in \{0,1\}^m \\ \vec{1}_m^T \cdot S \le m-1 \\ -K \cdot S + \vec{\varepsilon}_z \le -\vec{1}_m \end{array}\right\} (a) \\ & \left.\begin{array}{c} -\vec{\varepsilon}_z \le \vec{r} \\ \vec{\varepsilon}_z \le \vec{r} \\ \vec{r} \in \mathbb{N}^m \end{array}\right\} (b) \end{cases} \tag{16}$$

Example 12. Consider again the net system in Figure 3(a) and assume that $v = t_1t_2$ is the observed sequence and $\vec{y} = [2]^T$. Now the transition t_3 is observed and ILPP 1 has no feasible solution. We solve ILPP 3 to introduce a new unobservable transition. A solution $\vec{\varepsilon}_2 = [1, 0, -1]^T$ is found. Let $C_u = [C_u \ \vec{\varepsilon}_2]$. The resulting net system is shown in Figure 3(c) such that $t_1t_2t_3 \in \Omega(PN, M_0)$.

3.3. Online Identification Algorithm. There may be different approaches that can provide a solution to Problem 6; however we propose an online identification algorithm that recursively determines the structure of the unobservable subnet. Upon each observed transition t, we record t and decide whether a new unobservable transition should be introduced by solving ILPP 1. If it is the case, we obtain the unobservable transition by solving ILPP 3 and update the structure of the unobservable subnet. The identification procedure is shown by Algorithm 1, which monitors the system evolution and waits until a transition is observed.

The inputs of Algorithm 1 are the observable subnet PN_o and the initial marking M_0, i.e., Pre_o, $Post_o$, and M_0, and its output is the unobservable subnet PN_u such that we can obtain the whole net $PN = PN_o \otimes PN_u$.

In line (1), the algorithm initializes the variables. Initially, the unobservable subnet PN_u is null, i.e., $T_u = \emptyset$, C_u is a 0×0 matrix, and the vector $\vec{y}$ is also null. Note that the vector $\vec{y}$ is maintained and updated by Algorithm 1 when a transition is observed. In addition, n_u denotes the number of unobservable transitions, and it is the subscript when we name the newly identified transition ε_{n_u}.

In line (3), if n_u equals zero, i.e., no unobservable transition is identified, it is not necessary to solve ILPP 1. Instead, the condition of line (4) is checked.

In line (10), ILPP 1 has a feasible solution; then we only update the firing vector $\vec{y}$ and go back to line (2). In lines (14) and (15), a new unobservable transition is identified and we will update the corresponding variables. First, we add the solution $\vec{\varepsilon}_{n_u}$ to C_u as the last column, and the corresponding unobservable transition is called ε_{n_u}. Note that n_u increases by one when a new unobservable transition is identified such that the unobservable transitions are named ε_1, ε_2, ε_3,... We also obtain the value of variable $\vec{\sigma}_u$ that represents the firing vector of the previously identified unobservable transitions to enable the observed transition t. Hence we add $\vec{y}$ by $\vec{\sigma}_u$ and add the dimension of $\vec{y}$ by one, which means that the new identified unobservable transition ε_{n_u} fires once.

In line (19), it happens that ILPP 3 has no feasible solution, which probably stems from the necessity of a source transition or the inaccurate sensor reading.

Example 13. Consider the observable subnet in Figure 4(a), where $P = \{p_1, p_2, \ldots, p_7\}$, $T_o = \{t_1, t_2, \ldots, t_7\}$, and $M_0 = [3, 0, 1, 0, 0, 0, 0]^T$. Assume that the observed sequence is $v = t_1t_2t_2t_3t_6t_3t_6t_2$. Upon each observed transition, Algorithm 1 executes the identification procedure. We analyse the transitions in v in sequence.

Initially, the observed sequence v is an empty string, $T_u = \emptyset$, both C_u and $\vec{y}$ are null, and $n_u = 0$.

Occurrence of t_1: the relation $M_0 \ge Pre_o(\cdot, t_1)$ holds, and thus we do not have to solve ILPP 3 to find an unobservable transition. The algorithm sets $v = t_1$.

Occurrence of t_2: $M_0 + (Posto - Pre_o) \cdot \vec{v} = [2, 2, 1, 0, 0, 0, 0]^T \ge Pre_o(\cdot, t_2) = [0, 1, 1, 0, 0, 0, 0]^T$ holds. Thus the algorithm sets $v = t_1t_2$. *Occurrence of the second* t_2: since $M_0 + (Posto - Pre_o) \cdot \vec{v} \not\ge Pre_o(\cdot, t_2)$, ILPP 3 is solved and the solution $\vec{\varepsilon}_1 = [0, 0, 1, -1, 0, 0, 0]^T$ is found. Then the algorithm updates C_u, n_u, and $\vec{y}$ with $n_u = 1$, $\vec{y} = [1]^T$ and adds ε_1 to T_u. The Petri net $PN = PN_o \otimes PN_u$ is shown in Figure 4(b). Now $v = t_1t_2t_2$.

Input: A net system $\langle PN_o, M_0 \rangle$
Output: A Petri net PN with $PN = PN_o \otimes PN_u$
(1) $T_u = \emptyset$, $n_u = 0$, $C_u = [\,]$, $\overrightarrow{y} = [\,]$, $v = ''$;
(2) wait until transition t is observed;
(3) **if** $n_u == 0$ **then**
(4) **if** $M_0 + C_o \cdot \overrightarrow{v} \geq Pre_o(\cdot, t)$ **then**
(5) $v = vt$; goto 2;
(6) **else**
(7) goto 13;
(8) **end if**
(9) **else**
(10) **if** ILPP 1 admits a solution $\overrightarrow{\sigma}_u$ **then**
(11) $v = vt$; $\overrightarrow{y} = \overrightarrow{y} + \overrightarrow{\sigma}_u$; goto 2;
(12) **else**
(13) **if** ILPP 3 admits a solution $\overrightarrow{\varepsilon}_{n_u}$ and $\overrightarrow{\sigma}_u$ **then**
(14) $n_u = n_u + 1$; $C_u = [C_u, \overrightarrow{\varepsilon}_{n_u}]$; $T_u = T_u \cup \{\varepsilon_{n_u}\}$;
(15) $\overrightarrow{y} = \overrightarrow{y} + \overrightarrow{\sigma}_u$; $\overrightarrow{y} = \begin{bmatrix} \overrightarrow{y} \\ 1 \end{bmatrix}$;
(16) $v = vt$;
(17) goto 2;
(18) **else**
(19) **error:** cannot find a solution satisfying ILPP 3
(20) **return**
(21) **end if**
(22) **end if**
(23) **end if**

Algorithm 1: Identification of an unobservable subnet.

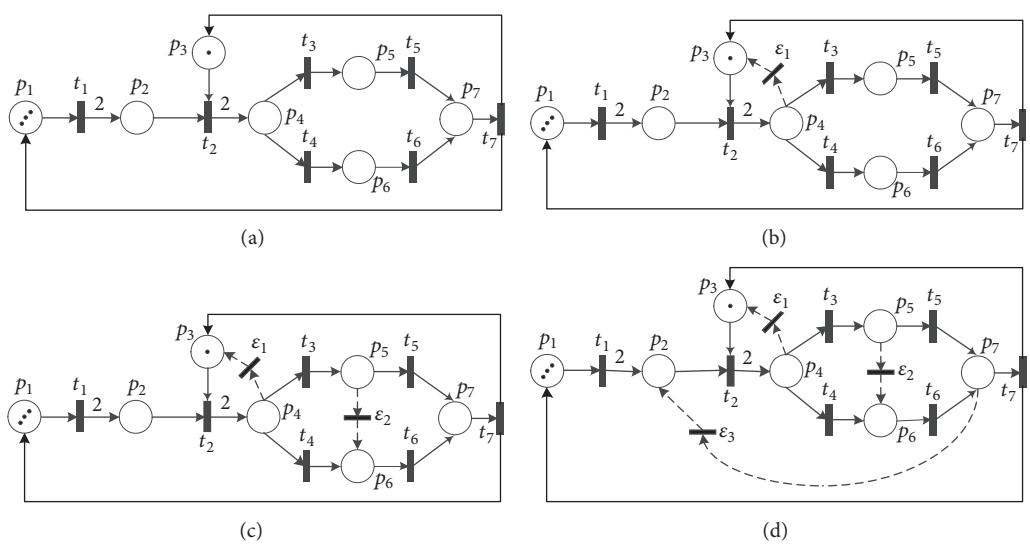

Figure 4: (a) The observable subnet PN_o, (b) the net identified after sequence $t_1t_2t_2$, (c) the net identified after sequence $t_1t_2t_2t_3t_6$, and (d) the net identified after sequence $t_1t_2t_2t_3t_6t_3t_6t_2$ by applying Algorithm 1.

Occurrence of t_3: since $n_u > 0$, we solve ILPP 1 and obtain the solution $\overrightarrow{\sigma}_u = \overrightarrow{0}$, implying that t_3 is enabled at the current marking, and $\overrightarrow{y} = \overrightarrow{y} + \overrightarrow{0}$ remains unchanged.

Occurrence of t_6: ILPP 1 has no feasible solution, i.e., a new unobservable transition should be introduced. Solving ILPP 3 and obtaining the solution $\overrightarrow{\varepsilon}_2 = [0, 0, 0, 0, -1, 1, 0]^T$,

we add ε_2 to T_u and update C_u. Now the net PN is shown in Figure 4(c) with $\overrightarrow{y} = [1, 1]^T$ and $v = t_1t_2t_2t_3t_6$.

Occurrence of the second t_3: ILPP 1 has a solution $\overrightarrow{\sigma}_u = \overrightarrow{0}$. Thus we just need to increase the observed sequence, i.e., $v = t_1t_2t_2t_3t_6t_3$.

Occurrence of the second t_6: A solution to ILPP 1 is $\overrightarrow{\sigma}_u = [0, 1]^T$, implying that t_6 is enabled after firing unobservable transition ε_2. The vector $\overrightarrow{y}$ is updated with $\overrightarrow{y} = \overrightarrow{y} + \overrightarrow{\sigma}_u = [1, 2]^T$ and let $v = t_1t_2t_2t_3t_6t_3t_6$.

Occurrence of the third t_2: ILPP 1 has no feasible solution and a solution $\overrightarrow{\varepsilon}_3 = [0, 1, 0, 0, 0, 0, -1]^T$ is found by solving ILPP 3. We add ε_3 to set T_u and update the structure of unobservable subnet PN_u. The net PN is shown in Figure 4(d). Furthermore, we have $\overrightarrow{y} = [1, 2, 1]^T$ and $v = t_1t_2t_2t_3t_6t_3t_6t_2$.

As shown in Example 13, an unobservable transition is introduced in some cases when a transition t is observed. An unobservable subnet PN_u is recursively identified by Algorithm 1 and the Petri net $PN = PN_o \otimes PN_u$ is finally provided. The following theorem proves that the output of Algorithm 1 is a solution to Problem 6.

Theorem 14. *Given the observable subnet PN_o and the initial marking M_0, under Assumption (A1), the identified Petri net $PN = PN_o \otimes PN_u$ by Algorithm 1 is a solution to Problem 6.*

Proof. The proof is made by induction.

Basic Step. In this case, the observed sequence v is an empty string and incidence matrix C_u is null. If transition t is now observed, then we check the relation $M_0 + C_o \cdot \overrightarrow{v} \geq Pre_o(\cdot, t)$, i.e., the relation $M_0 \geq Pre_o(\cdot, t)$. If it holds, let $w = vt = t$, and $w \in \Omega(PN, M_0)$ trivially holds. Otherwise, the solution $\overrightarrow{\varepsilon}_z$ is obtained by solving ILPP 3 and PN_u is updated by adding an unobservable transition ε_z. Let $w = t$ and $PN = PN_o \otimes PN_u$. By Proposition 11, $M_0[\varepsilon_z\rangle M'[t\rangle$ is trivially verified, namely, $w \in \Omega(PN, M_0)$.

Inductive Step. Assume that the observed sequence is v and the corresponding firing vector of unobservable transitions is $\overrightarrow{y} \in Y(v)$. If t is now observed, two different situations are considered.

(1) n_u equals 0, i.e., no unobservable transition is introduced during the period of observation v. If $M_0 + C_o \cdot \overrightarrow{v} \geq Pre_o(\cdot, t)$ is true, then t is enabled at the current marking, i.e., $w = vt \in \Omega(PN, M_0)$. Otherwise, a solution $\overrightarrow{\varepsilon}_z$ is found by solving ILPP 3 and the structure of PN is updated by adding a new unobservable transition ε_z whose incidence matrix column is $\overrightarrow{\varepsilon}_z$. Let $w = vt$. By Proposition 11, $M_0[v\rangle M'[\varepsilon_z\rangle M''[t\rangle$ holds, where $M', M'' \in R(PN, M_0)$, namely, $w \in \Omega(PN, M_0)$.

(2) n_u is greater than 0. In this situation, if ILPP 1 admits a solution, we update $\overrightarrow{y}$ and the observed sequence with $w = vt$. By Theorem 7, $w \in \Omega(PN, M_0)$ holds. On the other hand, if ILPP 1 has no feasible solution, the solution including $\overrightarrow{\varepsilon}_z$ and $\overrightarrow{\sigma}_u$ is obtained by solving ILPP 3. We update the related variables with $\overrightarrow{y} = \overrightarrow{y} + \overrightarrow{\sigma}_u$ and $C_u = [C_u \ \overrightarrow{\varepsilon}_z]$ and add a new unobservable transition ε_z to Petri net PN. Let $w = vt$, and then $w \in \Omega(PN, M_0)$ holds by Proposition 11.

Hence, we conclude that the Petri net PN identified by Algorithm 1 is a solution to Problem 6. □

3.4. Complexity of the Identification Algorithm. Obviously, the primary determinants of the computational difficulty of this algorithm are two integer linear programming problems ILPP 1 and ILPP 3. It is well known that an integer linear programming problem is NP-hard, but it can be solved easily if the number of variables is small. In Algorithm 1, if necessary, ILPP 1 and ILPP 3 are solved serially. In addition, the number of variables in ILPP 3 is greater than that in ILPP 1. Thus we analyse the complexity of ILPP 3 only.

We observe that $\overrightarrow{\sigma}_u$, $\overrightarrow{\varepsilon}_z$, and $\overrightarrow{r}$ are the integer variables in ILPP 3 and S is a binary variable. Then the number of integer variables is $u = n_u + 3m$, where n_u denotes the number of unobservable transitions and $3m$ denotes the sum of the dimensions of $\overrightarrow{\varepsilon}_z$, $\overrightarrow{r}$, and S.

In other words, the number of integer variables in ILPP 3 is linear (the coefficient is 1) with respect to that of unobservable transitions. Thus ILPP 3 is a simple integer programming problem that can be solved efficiently. For example, the unobservable subnet with three unobservable transitions in Example 13 is identified using LPSOLVE (a mixed integer linear programming solver [24]) in 0.0541 seconds on a notebook computer with 2 GHz processor and 8 GB RAM. In addition, we test the presented algorithm on an observable subnet with 43 places, 40 transitions, 140 arcs. In this setting, 15 unobservable transitions are found by solving 25 integer programming problems in 0.3698 seconds on the same computer using LPSOLVE solver.

4. Model Identification with Partial Observable Places

4.1. Problem Statement. In Section 3, we assume that no place is observable. In such a setting, we may not be able to determine an unobservable transition that describes exactly the unknown behavior of a system by solving ILPP 3. The main reasons are the limited knowledge of system evolution and the possibility that there is more than one solution that satisfies the considered linear constraint set even if we choose a performance index. To determine a more authentic unobservable transition, more information on system evolution is needed. In this section, we consider that some of places are observable.

Let us consider a net system $\langle PN, M_0\rangle$, where $PN = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Assume that $P_o = \{P_{\alpha_1}, P_{\alpha_2}, \ldots, P_{\alpha_h}\} \subseteq P$ with $h \leq m$ is the set of observable places. For each $M \in R(PN, M_0)$, the restriction of M to P_o is called a *partial marking* denoted as M^{po}, and $M|_{po} = M^{po}$ denotes that the restriction of M to P_o is M^{po}.

If some of places in a Petri net model are observable, the observed information during system evolution includes not only the observed transitions but also the corresponding

partial markings. We denote the output of the system with observable places by $w = t_{\beta_1} t_{\beta_2} \dots t_{\beta_k}$ with $k \geq 1$ and the corresponding reached partial markings by $M_1^{po} M_2^{po} \dots M_k^{po}$, where $t_{\beta_i} \in T_o$ and M_i^{po} is the observed partial marking after transition t_{β_i} for $i = 1, \dots, k$.

Example 15. Consider again the observable subnet shown in Figure 4(a). If the set of observable places is $P_o = \{p_{\alpha_1}, p_{\alpha_2}, p_{\alpha_3}\} = \{p_1, p_3, p_7\}$, for an observed sequence $w = t_{\beta_1} t_{\beta_2} t_{\beta_3} t_{\beta_4} = t_1 t_2 t_2 t_3$, the corresponding partial markings can be $M_1^{po} = [2, 1, 0]^T$, $M_2^{po} = [2, 0, 0]^T$, $M_3^{po} = [1, 0, 0]^T$, $M_4^{po} = [1, 0, 0]^T$; i.e., the observed transition-partial-marking sequence is $\sigma = M_0 t_1 M_1^{po} t_2 M_3^{po} t_2 M_3^{po} t_3 M_4^{po}$. It is clear that the sequence σ cannot be generated by the net in Figure 4(a) and thus some unobservable transitions must occur.

Problem 16. Assume that the observable subnet $PN_o = (P, T_o, Pre_o, Post_o)$ and the initial marking M_0 are given and $P_o = \{P_{\alpha_1}, P_{\alpha_2}, \dots, P_{\alpha_h}\} \subseteq P$ with $h \leq m$ is the set of observable places. The observed output includes the transition sequence $w = t_{\beta_1} t_{\beta_2} \dots t_{\beta_k}$ with $k \geq 1$ and the corresponding reached partial markings $M_1^{po} M_2^{po} \dots M_k^{po}$, where $t_{\beta_i} \in T_o$ and M_i^{po} is the observed partial marking after transition t_{β_i} for $i = 1, \dots, k$. The problem consists in determining a Petri net PN with $PN = PN_o \otimes PN_u$ such that $w \in \Omega(PN, M_0)$ and each marking $M_i \in R(PN, M_0)$ after t_{β_i} fires satisfies $M_i|_{po} = M_i^{po}$ for $i = 1, \dots, k$.

4.2. The Solution to Problem 16. This subsection discusses whether an unobservable transition has to be introduced or not under the assumption that some of places are observable. Moreover, we present a technique to show how the unobservable transitions can be introduced.

Theorem 17. *Consider net system $\langle PN, M_0 \rangle$. Assume that $P_o = \{P_{\alpha_1}, P_{\alpha_2}, \dots, P_{\alpha_h}\}$ with $h \leq m$ is the set of observable places and the observed transition sequence is v with $\vec{y}$ being the corresponding firing vector. Assume that a transition t_{β_k} and the corresponding partial markings M_k^{po} are observed. If (17a), (17b), and (17c) admit a solution, then $vt_{\beta_k} \in \Omega(PN, M_0)$ and the marking M_k after t_{β_k} fires satisfies $M_k|_{po} = M_k^{po}$.*

$$M_0 + C_o \cdot \vec{v} + C_u \cdot \vec{y} + C_u \cdot \vec{\sigma}_u \geq Pre_o\left(\cdot, t_{\beta_k}\right) \tag{17a}$$

$$A \cdot \left(M_0 + C_o \cdot \vec{v} + C_u \cdot \vec{y} + C_u \cdot \vec{\sigma}_u + C_o\left(\cdot, t_{\beta_k}\right)\right) = M_k^{po} \tag{17b}$$

$$\vec{\sigma}_u \in \mathbb{N}^{n_u} \tag{17c}$$

where A is an $h \times m$ matrix such that $A(i, \alpha_i) = 1$ for $i = 1 \dots h$ and 0 otherwise. The matrix A is dedicated to projecting a marking $M \in R(PN, M_0)$ onto a partial marking based on the set of observable places P_o.

Proof. If we do not consider the observable places and remove constraint (17b), then (17a), (17b), and (17c) are reduced to be (8). By Theorem 7, $vt_{\beta_k} \in \Omega(PN, M_0)$ is trivially verified. However, if constraint (17b) is removed, there may be numerous solutions that satisfy $vt_{\beta_k} \in \Omega(PN, M_0)$ but violate $M_k|_{po} = M_k^{po}$. When the set of observable places is taken into account, constraint (17b) guarantees that these solutions that satisfy $M_k|_{po} = M_k^{po}$ are chosen only. Thus this theorem is verified. □

The solution of (17a), (17b), and (17c) is also not a singleton. We again choose the performance index $\vec{1}_{n_u}^T \cdot \vec{\sigma}_u$ and obtain the following integer linear programming problem that minimizes the considered performance index:

$$\text{ILPP 4}: \begin{cases} \min & \vec{1}_{n_u}^T \cdot \vec{\sigma}_u \\ s.t. & \text{Eq. (17a), (17b), and (17c)} \end{cases} \tag{18}$$

Proposition 18. *Consider net system $\langle PN, M_0 \rangle$. Assume that P_o is the set of observable places and the observed transition sequence is v with $\vec{y}$ being the corresponding firing vector. A transition t_{β_k} and the corresponding partial marking M_k^{po} are observed. If there exists a vector $\vec{\varepsilon}_z$ satisfying (19a), (19b), (19c), (19d), and (19e), then $vt_{\beta_k} \in \Omega(PN', M_0)$ and the marking M_k after t_{β_k} fires satisfies $M_k|_{po} = M_k^{po}$, where PN' is a Petri net obtained by adding ε_z to PN.*

$$M_0 + C_o \cdot \vec{v} + C_u \cdot \left(\vec{y} + \vec{\sigma}_u\right) + \vec{\varepsilon}_z \geq Pre_o\left(\cdot, t_{\beta_k}\right) \tag{19a}$$

$$A \cdot \left(M_0 + C_o \cdot \vec{v} + C_u \cdot \left(\vec{y} + \vec{\sigma}_u\right) + \vec{\varepsilon}_z + C_o\left(\cdot, t_{\beta_k}\right)\right) = M_k^{po} \tag{19b}$$

$$\vec{\sigma}_u \in \mathbb{N}^{n_u} \tag{19c}$$

$$\vec{\varepsilon}_z \in \mathbb{Z}^m \tag{19d}$$

$$\vec{\varepsilon}_z \neq \vec{0} \tag{19e}$$

Proof. If constraint (19b) is removed, then (19a), (19b), (19c), (19d), and (19e) are reduced to be (13a), (13b), (13c), and (13d). By Proposition 11, $vt_{\beta_k} \in \Omega(PN', M_0)$ is trivially verified. On the other hand, constraint (19b) guarantees that the solutions satisfying $M_k|_{po} = M_k^{po}$ are chosen only. Thus, this proposition is proved. □

Although constraint (19b) is imposed to choose the solutions satisfying the observed partial markings, the solution of (19a), (19b), (19c), (19d), and (19e) is still not unique. As shown in ILPP 3, we choose $\vec{1}_m^T \cdot \text{abs}(\vec{\varepsilon}_z)$ as the performance index and obtain the following programming problem:

ILPP 5 :

$$\begin{cases} \min & \vec{1}_m^T \cdot \text{abs}\left(\vec{\varepsilon}_z\right) \\ s.t. & \text{Eq. (19a), (19b), (19c), (19d), and (19e)} \end{cases} \tag{20}$$

The solution to Problem 16 can be obtained by making slight changes to Algorithm 1, i.e., replacing ILPP 1 and ILPP 3 with ILPP 4 and ILPP 5, respectively.

5. Conclusion

The paper addresses the problem of model identification of unobservable system behavior by assuming that the observable subnet model is known. We consider two different cases: first, no place is observable; second, a subset of places is observable. Two problems are discussed in both cases and an online algorithm is proposed which computes the unobservable transitions recursively and provides solutions for these two problems. The computational complexity of the algorithm is analysed and it can be executed efficiently because of a small number of unknowns of the integer programming problem.

Further research will be twofold. First, we plan to develop an approach that provides a solution to the problem in this paper and minimizes the unobservable subset, e.g., minimizing the number of unobservable transitions or minimizing the sum of weights of the arcs connected with unobservable transitions. Second, we aim to explore the identification of unobservable transitions in the case that the given subnet includes unobservable transitions.

Data Availability

The authors confirm that the data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by Key Scientific Research Program of Higher Education of Henan Province of China (no. 19A520036) and in part by Key Technology R&D Program of Science and Technology Department of Henan Province of China (no. 182102210500).

References

[1] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Springer, New York, NY, USA, 2nd edition, 2008.

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

[3] 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, 2015.

[4] F. Basile, P. Chiacchio, and G. de Tommasi, "An efficient approach for online diagnosis of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 748–759, 2009.

[5] 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, 2010.

[6] M. Dotoli, M. P. Fanti, A. M. Mangini, and W. Ukovich, "On-line fault detection in discrete event systems by Petri nets and integer linear programming," *Automatica*, vol. 45, no. 11, pp. 2665–2672, 2009.

[7] Y. Ru and C. N. Hadjicostis, "Fault diagnosis in discrete event systems modeled by partially observed petri nets," *Discrete Event Dynamic Systems*, vol. 19, no. 4, pp. 551–575, 2009.

[8] E. M. Gold, "Complexity of automaton identification from given data," *Information and Control*, vol. 37, no. 3, pp. 302–320, 1978.

[9] D. Angluin, "Inference of reversible languages," *Journal of the ACM*, vol. 29, no. 3, pp. 741–765, 1982.

[10] A. Giua and C. Seatzu, "Identification of free-labeled Petri nets via integer programming," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 7639–7644, IEEE, Seville, Spain, December 2005.

[11] M. P. Cabasino, A. Giua, and C. Seatzu, "Identification of Petri nets from knowledge of their language," *Discrete Event Dynamic Systems*, vol. 17, no. 4, pp. 447–474, 2007.

[12] M. P. Cabasino, A. Giua, C. N. Hadjicostis, and C. Seatzu, "Fault model identification and synthesis in Petri nets," *Discrete Event Dynamic Systems*, vol. 25, no. 3, pp. 419–440, 2015.

[13] M. Dotoli, M. P. Fanti, and A. M. Mangini, "An optimization approach for identification of Petri Nets," in *Proceedings of the 8th International Workshop on Discrete Event Systems, WODES '06*, pp. 332–337, IEEE, Michigan, Mich, USA, July 2006.

[14] M. Dotoli, M. P. Fanti, and A. Mangini, "Real time identification of discrete event systems using Petri nets," *Automatica*, vol. 44, no. 5, pp. 1209–1219, 2008.

[15] M. Dotoli, M. Pia Fanti, A. M. Mangini, and W. Ukovich, "Identification of the unobservable behaviour of industrial automation systems by Petri nets," *Control Engineering Practice*, vol. 19, no. 9, pp. 958–966, 2011.

[16] K. Hiraishi, "Construction of a class of safe Petri nets by presenting firing sequences," in *Proceedings of the International Conference on Application and Theory of Petri Nets*, vol. 616, pp. 244–262, Sheffield, UK, 1992.

[17] M. E. Meda, A. Ramirez, and A. Malo, "Identification in discrete event systems," in *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 740–745, IEEE, California, Calif, USA, October 1998.

[18] M. Meda-Campaña and E. López-Mellado, "Identification of concurrent discrete event systems using Petri nets," in *Proceedings of the 17th IMACS World Congress on Computational and Applied Mathematics*, pp. 11–15, Paris, France, 2005.

[19] R. Bekrar, N. Messai, N. Essounbouli, A. Hamzaoui, and B. Riera, "Off-line identification for a class of discrete event systems using safe Petri nets," *IFAC Proceedings Volumes*, vol. 39, no. 17, pp. 221–226, 2006.

[20] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.

[21] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Annual Reviews in Control*, vol. 37, no. 2, pp. 308–320, 2013.

[22] T. Murata, "Petri nets: properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[23] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[24] LPSOLVE, https://sourceforge.net/projects/lpsolve/.