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

Guanghui Zhu,1,2 Ya Wang,1, and Yajie Wang3

1School of Information Engineering, Xuchang University, Xuchang 461000, China
2School of Electro-Mechanical Engineering, Xidian University, Xi’an 710071, China
3School 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
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 \( \tau \) 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 \( PN = (P, T, \text{Pre}, \text{Post}) \), where \( P \) is the set of places, \( T \) is the set of transitions, \( \text{Pre} : P \times T \rightarrow \mathbb{N} \), and \( \text{Post} : P \times T \rightarrow \mathbb{N} \) are the pre- and postincidence matrices, respectively, which specify the structure of the Petri net. For \( p \in P, t \in T, \) and \( x \in \mathbb{N} \), \( \text{Pre}(p, t) = x \) if there exists an arc with weight \( x \) from \( p \) to \( t \), 0 otherwise; \( \text{Post}(p, t) = x \) 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 \( C = \text{Post} - \text{Pre} \). For \( t \in T \), its preset is defined as \( \text{Pre}(p, t) > 0 \), and its postset is defined as \( \text{Post}(p, t) > 0 \). A transition \( t \) is said to be a source transition if it satisfies \( \text{Pre}(p, t) = 0 \). A marking of a Petri net is a vector \( M : P \rightarrow \mathbb{N} \), and \( M(p) \) indicates the number of tokens in place \( p \). A net system \( (PN, M_0) \) is a Petri net with an initial marking \( M_0 \).

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

Given a transition sequence \( \sigma \) and a marking \( M, M[\sigma] \) denotes that \( \sigma \) is enabled at marking \( M \) and \( M[\sigma]M' \) denotes that a new marking \( M' \) is reachable from \( M \) after firing \( \sigma \). All the markings reachable from \( M_0 \) comprise the \emph{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

\[
L(PN, M_0) = \{ \sigma \in T^* | M_0[\sigma]\} \tag{1}
\]

which is called the \emph{language} of a Petri net.

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

\[
M = M_0 + C \cdot \overrightarrow{\sigma} \tag{2}
\]

which is called the \emph{state equation} of a Petri net system.

2.2. Labeled Petri Net. Given a Petri net \( PN = (P, T, \text{Pre}, \text{Post}) \) and the set of events \( E \), a labeling function \( \lambda : T \rightarrow E \cup \{ \varepsilon \} \) assigns to each transition \( t \in T \) either a symbol from the event set \( E \) or an empty string \( \varepsilon \). A transition with label \( \varepsilon \) is called an \emph{unobservable transition} and the set of unobservable transitions is denoted as \( T_\varepsilon = \{ t \in T | \lambda(t) = \varepsilon \} \) with \( n_\varepsilon = |T_\varepsilon| \) being the cardinality of set \( T_\varepsilon \). All other transitions whose labels are not \( \varepsilon \) comprise the set of \emph{observable transitions} \( T_\sigma = \{ t \in T | \lambda(t) \neq \varepsilon \} \) and \( n_\sigma = |T_\sigma| \). Thus the set \( T \) is divided into two disjoint subsets \( T_\varepsilon \) and \( T_\sigma \) with \( T = T_\varepsilon \cup T_\sigma \). We assume that each label \( \varepsilon \in E \) can only be assigned to one transition, i.e., the set \( E \) is isomorphic to the set of observable transitions \( T_\sigma \), and thus without loss of generality we assume \( E = T_\sigma \). 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 \( \sigma_o \in T_\sigma^* \), we define an \( n_\sigma \)-dimensional vector \( \overrightarrow{\sigma_o} : T_\sigma \rightarrow \mathbb{N} \) and \( \overrightarrow{\sigma_o}(t) = k \) if \( t \in T_\sigma \) occurs \( k \) times in \( \sigma_o \). Analogously, for each \( \sigma_u \in T_u^* \), an \( n_u \)-dimensional vector \( \overrightarrow{\sigma_u} : T_u \rightarrow \mathbb{N} \) is defined and \( \overrightarrow{\sigma_u}(t) = k \) if \( t \in T_u \) occurs \( k \) times in \( \sigma_u \).

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:

\[
\min \quad \overrightarrow{a}^T \cdot \text{abs} (\overrightarrow{x}) + \overrightarrow{b}^T \cdot \overrightarrow{y} \tag{3}
\]

s.t. \( A \)

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

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

\[
\min \quad \overrightarrow{a}^T \cdot \overrightarrow{z} + \overrightarrow{b}^T \cdot \overrightarrow{y} \tag{4}
\]

s.t. \( A \)

\[
-\overrightarrow{z} \leq \overrightarrow{\rho} \]

\[
\overrightarrow{z} \leq \overrightarrow{\rho} \]

\[
\overrightarrow{z} \geq 0
\]

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

In fact, for the real numbers \( x_1 \) and \( x_2 \) that are the first entries of vectors \( \overrightarrow{x} \) and \( \overrightarrow{z} \), respectively, there are three different cases.

1. \( x_1 > 0 \). Constraint \( -x_1 \leq x_2 \) is always fulfilled. Because of the constraint \( x_1 \leq x_2 \), \( x_1 \) is necessarily at least as large as \( x_1 \). On the other hand, \( x_1 \) is in the objective function such that it tends to be as small as possible. Thus \( x_1 \) will be equal to \( x_1 \), i.e., \( x_1 = \text{abs}(x_1) \).

2. \( x_1 = 0 \). In this case, constraints \( -x_1 \leq x_2 \) and \( x_1 \leq x_2 \) are always verified because of \( x_1 \geq 0 \). Moreover \( x_2 \) is in the objective function, and thus it tends to be zero, i.e., \( x_2 = \text{abs}(x_1) \).
\( x_1 < 0 \). Being analogous to case (1), \( x_1 = \text{abs}(x_1) \) is trivially verified.

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

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

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

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

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

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

\[
\Omega(PN, M_0) = \{ w \in T_o^* \mid e \in L(PN, M_0), w = Q_o(e) \}.
\]

A Petri net is said to be acyclic if it has no directed circuit. Equation (2) shows that there exists a nonnegative integer \( x \) such that \( M = M_o + C \cdot x \), 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 \( x \) satisfying \( M = M_o + C \cdot x \).

A net \( PN = (P, T, \text{Pre}, \text{Post}) \) is pure if it has no self-loop; namely, \( \forall t \in T \), \( \text{Pre}(p, t) \times \text{Post}(p, t) = 0 \) 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

\[
\Sigma(w) = \{ y \in T_u^* \mid y = Q_u(Q_o^{-1}(w)) \}.
\]

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

\[
Y(w) = \{ y \in \mathbb{N}^n \mid y \in \Sigma(w) \}.
\]

**Example 5.** Figure 1 shows a net system with \( T_o = \{ t_1, t_2 \} \), \( T_u = \{ e_1, e_2, e_3 \} \), and \( M_0 = [1, 0, 0, 0]^T \). If an observed transition sequence is \( w = t_1t_2 \), by Definition 4, we observe \( Q_o^{-1}(w) = \{ t_1, t_2 \} \). Thus \( \Sigma(w) = \{ e_1, e_2, e_3 \} \) 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) \( M_0(v) \) 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) \( M_0(v) \) does not hold. We infer that one or more unobservable transitions not included in \( PN_o \) occur.

We can infer a net \( PN = (P, T, \text{Pre}, \text{Post}) \) 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 \( PN_u = (P, T_u, \text{Pre}_u, \text{Post}_u) \).

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 \( \text{Pre} = [\text{Pre}_o, \text{Pre}_u] \), and its postincidence matrix is represented by \( \text{Post} = [\text{Post}_o \ \text{Post}_u] \). Correspondingly, we have \( C = [C_o \ C_u] \).

**Problem 6.** Assume that an observable subnet \( PN_o = (P, T_o, \text{Pre}_o, \text{Post}_o) \) 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 \( PN = (P, T, \text{Pre}, \text{Post}) \) 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...
subnet \(PN_o\) 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 system with \(PN = PN_o \otimes PN_u\). Let \(v \in \Omega(PN, M_0)\) be the current observed sequence and \(\overrightarrow{y} \in Y(v)\) be a corresponding firing vector of unobservable transitions. A new transition \(t\) is observed. If (8) admits a solution, then \(vt \in \Omega(PN, M_0)\).

\[
M_0 + C_o \cdot \overrightarrow{v} + C_u \cdot \overrightarrow{y} + C_u \cdot \overrightarrow{\sigma_u} \geq \text{Pre}_o(\cdot, t)
\]

where \(\overrightarrow{\sigma_u} \in \mathbb{R}^{n_u}\).

**Proof.** Let \(M' = M_0 + C_o \cdot \overrightarrow{v} + C_u \cdot \overrightarrow{y}\) 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 \(\overrightarrow{\sigma_u}\) such that \(M' + C_u \cdot \overrightarrow{\sigma_u} \geq \text{Pre}_o(\cdot, t) \geq \overrightarrow{0}\). Let \(M'' = M' + C_u \cdot \overrightarrow{\sigma_u}\). By Assumption (A1) and Theorem 3, there exists an unobservable transition sequence \(\sigma_u\) such that \(M'' \in R(PN, M_0)\) and \(M'(\sigma_u)M''\). Since \(M'' \geq \text{Pre}_o(\cdot, t)\), \(M'(\sigma_u)M''(t)\) holds. Thus the transition \(t\) can be observed after \(v\) by firing \(\sigma_u\), 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 \(1^{-T} \overrightarrow{\sigma_u}\) which minimizes the sum of the firing times of unobservable transitions. We define the following ILPP:

\[
\text{ILPP } 1:\begin{align*}
\min & \ 1^{-T} \overrightarrow{\sigma_u} \\
\text{s.t.} & \ Eq. (8)
\end{align*}
\]

where \(1^{-T}\) is a row vector of dimension \(n_u\) with each element being 1.

**Example 8.** Consider the net system in Figure 2, where \(P = \{p_1, p_2, p_3\}\), \(T_o = \{t_1, t_2, t_3\}\), and \(T_u = \{\varepsilon_1\}\), 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 \(\overrightarrow{y} = [1]^T\). If the transition \(t_2\) is now observed, we find a solution \(\overrightarrow{\sigma_u} = [1]^T\) by solving ILPP 1. According to Theorem 7, we have \(v = t_1t_2 \in \Omega(PN, M_0)\). In fact, there exists a firing sequence \(\sigma = \varepsilon_1t_1\varepsilon_2\) such that \(M_0(\sigma)\).

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 \(\varepsilon_z\) 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 \(\varepsilon_z\) to \(PN_u\). By Assumption (A1), the structure of \(PN_u\) can be determined by its incidence matrix \(C_u\). Assume that \(\overrightarrow{e_z}\) is the column in the incidence matrix associated with \(\varepsilon_z\) and then the incidence matrix of \(PN_u'\) is \(C_u' = [C_u \ \overrightarrow{e_z}]\); i.e., \(PN_u'\) retains the structure of \(PN_u\) but includes an additional transition \(\varepsilon_z\). The following proposition provides a method to compute the unobservable transition.

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

\[
M_0 + C_o \cdot \overrightarrow{v} + C_u \cdot \overrightarrow{y} + \overrightarrow{e_z} \geq \text{Pre}_o(\cdot, t)
\]

where \(Z\) is the set of integers.

**Proof.** Let \(M' = M_0 + C_o \cdot \overrightarrow{v} + C_u \cdot \overrightarrow{y}\) be a marking of \(PN\) after firing sequence \(v\). Obviously, \(M' \in R(PN', M_0)\) holds. If \(\overrightarrow{e_z}\) is a solution of (10), then \(M' + \overrightarrow{e_z} \geq \text{Pre}_o(\cdot, t)\) holds. Let \(M'' = M' + \overrightarrow{e_z}\). Since \(C_u = [C_u \ \overrightarrow{e_z}]\), i.e., \(\overrightarrow{e_z}\)
is the last column of incidence matrix $C_u$, by Theorem 3, $M' + C_u' + \epsilon_z = M'' \geq \text{Pre}_e(\cdot, t)$ holds, where $\epsilon_z$ denotes the new unobservable transition whose incidence matrix column is $\epsilon_z$. Thus $M'(\epsilon_z)M''(\epsilon_z)^T$ holds, i.e., $\forall t \in \Omega(PN', M_0)$. \hfill $\square$

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 $0 \times 0$ matrix and we can set $C_u = \epsilon = 0$. 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 $f = \frac{1}{m} \cdot \text{abs}(\epsilon_z)$ is considered as a performance index which minimizes the sum of weights of the arcs connected with the unobservable transition $\epsilon_z$, the following integer programming problem is formulated:

$$\min \frac{1}{m} \cdot \text{abs}(\epsilon_z)$$

s.t. Eq. (10) \hfill (11)

$$S \in \{0,1\}^m$$

$$\frac{1}{m} \cdot S \leq m - 1$$

$$-K \cdot S + \epsilon_z \leq -\frac{1}{m}$$

where $K$ is a constant that is large enough.

If the performance index $f = \frac{1}{m} \cdot \text{abs}(\epsilon_z)$ is chosen, the solution of (11) is always $\epsilon_z \geq 0$; 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 $\epsilon_z$ 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:

$$\begin{align*}
\min & \quad \frac{1}{m} T \cdot \overrightarrow{r} \\
\text{s.t.} & \quad \text{Eq. (10)} \\
& \quad S \in \{0,1\}^m \\
& \quad \frac{1}{m} \cdot S \leq m - 1 \\
& \quad -K \cdot S + \epsilon_z \leq -\frac{1}{m} \\
& \quad \epsilon_z \leq \overrightarrow{r} \\
& \quad \overrightarrow{r} \in \mathbb{N}^m \\
& \quad \overrightarrow{r} \in \mathbb{N}^m \\
& \quad (\overrightarrow{r} \geq \overrightarrow{r}) \quad (b)
\end{align*}$$

ILPP 2

where $\overrightarrow{r}$ and (b) are, respectively, the additional variables and constraints associated with $\text{abs}(\epsilon_z)$ 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 $T_u = \{\epsilon_1\}$. Assume that the observed sequence is $\nu = t_1t_2$ satisfying $\nu \in \Omega(PN, M_0)$ and that the corresponding firing vector is $\nu = [2]$. 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 $\epsilon_z = [1, 1, -1]^T$ is found by solving ILPP 2. Let $C_u = [C_u', \epsilon_z]$ and update the structure of $PN_u$ by adding a new unobservable transition $\epsilon_z$. 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 $\epsilon_z$. 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 $\nu \in \Omega(PN, M_0)$ is the current observed sequence with $\nu \in Y(\nu)$. A new transition $t$ is observed. If there exists a vector $\epsilon_z$ satisfying (13a), (13b), (13c), and (13d), then $\forall t \in \Omega(PN', M_0)$, where $PN'$ is a Petri net obtained by adding $\epsilon_z$ to $PN$. 

$$M_0 + C_o \cdot \overrightarrow{v} + C_u \cdot [\nu^T + \sigma_u^T] + \epsilon_z \geq \text{Pre}_e(\cdot, t)$$

$$\sigma_u \in \mathbb{N}^n$$

$$\epsilon_z \in \mathbb{Z}^m$$

$$\epsilon_z \neq \overrightarrow{0}$$

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

$$M' + C_u' \cdot \sigma_u + \epsilon_z \geq \text{Pre}_e(\cdot, t)$$

Since $C_u' = [C_u', \epsilon_z]$, (14) is rewritten as

$$M' + C_u' \cdot \sigma_u \geq \text{Pre}_e(\cdot, t).$$

Let $M'' = M' + C_u' \cdot [\sigma_u^T \cdot 1]^T$. By Theorem 3, there exists an unobservable transition sequence $\sigma_u'$ such that $M''[\sigma_u']M''$. Since $M'' \geq \text{Pre}_e(\cdot, t) \cdot M''[\sigma_u'M''(\sigma_u')^T]$, holds, i.e., $\forall t \in \Omega(PN', M_0)$. \hfill $\square$

As shown in (10), the solution of (13a), (13b), (13c), and (13d) is neither unique. We still consider the performance index $f = \frac{1}{m} \cdot \text{abs}(\epsilon_z)$ and convert the absolute value in
objective function into linear constraints. The final integer linear programming problem is shown as follows:

\[
\begin{align*}
\text{min} & \quad \mathbf{r}^T \mathbf{1}_m, \\
\text{s.t.} & \quad E \text{q. (13a), (13b), (13c) and (13d)} \\
\text{ILPP 3:} & \quad \begin{cases}
S \in \{0,1\}^m \\
-\mathbf{1}_m^T S \leq m - 1 \\
\mathbf{r}^T \mathbf{1}_m = \epsilon \\
K \cdot S + \mathbf{e}_\mathbf{z} \leq \mathbf{r} \\
\mathbf{e}_\mathbf{z} \leq \mathbf{r} \\
\mathbf{r} \in \mathbb{N}^m
\end{cases}
\end{align*}
\]

Example 12. Consider again the net system in Figure 3(a) and assume that \( \nu = t_1t_2 \) is the observed sequence and \( \mathbf{y} = [2]^T \). Now the transition \( t_1 \) is observed and ILPP 1 has no feasible solution. We solve ILPP 3 to introduce a new unobservable transition. A solution \( \mathbf{x}_2 = [1,0,-1]^T \) is found. Let \( C_0 = [C_{n_1} \mathbf{e}_2] \). 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., \( \text{Pre}_o, \text{Post}_o \), 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 \( 0 \times 0 \) matrix, and the vector \( \mathbf{y} \) is also null. Note that the vector \( \mathbf{y} \) 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 \( e_n \).

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 \( \mathbf{y} \) 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 \( \mathbf{e}_n \) to \( C_u \) as the last column, and the corresponding unobservable transition is called \( e_n \). Note that \( n_u \) increases by one when a new unobservable transition is identified such that the unobservable transitions are named \( e_1, e_2, e_3, \ldots \). We also obtain the value of variable \( \mathbf{e}_u \) that represents the firing vector of the previously identified unobservable transitions to enable the observed transition \( t \). Hence we add \( \mathbf{x} \) by \( \mathbf{e}_u \) and add the dimension of \( \mathbf{y} \) by one, which means that the new identified unobservable transition \( e_n \) 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 \( \nu = t_1t_2t_3t_6t_7t_2 \). Upon each observed transition, Algorithm 1 executes the identification procedure. We analyse the transitions in \( \nu \) in sequence.

Initially, the observed sequence \( \nu \) is an empty string, \( T_u = \emptyset \), both \( C_u \) and \( \mathbf{y} \) are null, and \( n_u = 0 \).

Occurrence of \( t_1 \): the relation \( M_0 \geq \text{Pre}_o(\cdot, t_1) \) holds, and thus we do not have to solve ILPP 3 to find an unobservable transition. The algorithm sets \( \nu = t_1 \).

Occurrence of \( t_2 \): \( M_0 + (\text{Post}_o - \text{Pre}_o) \cdot \mathbf{v} = [2, 2, 1, 0, 0, 0, 0]^T \geq \text{Pre}_o(\cdot, t_2) = [0, 1, 1, 0, 0, 0, 0]^T \) holds. Thus the algorithm sets \( \nu = t_1t_2 \). Occurrence of the second \( t_2 \): since \( M_0 + (\text{Post}_o - \text{Pre}_o) \cdot \mathbf{v} \not\geq \text{Pre}_o(\cdot, t_2) \), ILPP 3 is solved and the solution \( \mathbf{z}_2 = [0, 0, 1, 0, 0, 0]^T \) is found. Then the algorithm updates \( C_u, n_u \), and \( \mathbf{y} \) with \( n_u = 1, \mathbf{y} = [1]^T \) and adds \( e_1 \) to \( T_u \). The Petri net \( PN = PN_o \otimes PN_u \) is shown in Figure 4(b). Now \( \nu = t_1t_2t_2 \).
**Input:** A net system \( \langle PN_o, M_0 \rangle \)

**Output:** A Petri net \( PN \) with \( PN = PN_o \otimes PN_{u} \)

1. \( T_u = 0, n_u = 0, C_u = [], \vec{y} = [], \nu = n' \)
2. Wait until transition \( t \) is observed;
3. If \( n_u = 0 \) then
   4. If \( M_0 + C_o \cdot \vec{y} \geq Pre(t) \) then
      5. \( v = v; \) goto 2;
   6. else
      7. goto 13;
   8. end if
9. else
   10. If ILPP 1 admits a solution \( \vec{\sigma}_u \) then
      11. \( v = v; \vec{y} = \vec{y} + \vec{\sigma}_u; \) goto 2;
   12. else
      13. If ILPP 3 admits a solution \( \vec{\epsilon}_n \) and \( \vec{\sigma}_u \) then
         14. \( n_u = n_u + 1; C_u = [C_o, \vec{\epsilon}_n]; T_u = T_u \cup [\vec{\epsilon}_n]; \)
         15. \( \vec{y} = \vec{y} + \vec{\sigma}_u; \vec{y} = \begin{bmatrix} \vec{y} \\ 1 \end{bmatrix}; \)
         16. \( v = v; \)
         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 \( t_1, t_2, t_3 \), (c) the net identified after sequence \( t_1, t_2, t_3, t_4, t_6 \), and (d) the net identified after sequence \( t_1, t_2, t_3, t_4, t_6 \) by applying Algorithm 1.

**Occurrence of \( t_3 \):** Since \( n_u > 0 \), we solve ILPP 1 and obtain the solution \( \vec{\sigma}_u = \vec{0} \), implying that \( t_3 \) is enabled at the current marking, and \( \vec{y} = \vec{y} + \vec{0} \) 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 \( \vec{\epsilon}_2 = [0, 0, 0, -1, 1, 0]^T \),
we add $\varepsilon_3$ to $T_u$ and update $C_u$. Now the net $PN$ is shown in Figure 4(c) with $\overrightarrow{y} = [1, 1, 1]^T$ and $\nu = t_1 t_2 t_3 t_4 t_5$.

Occurrence of the second $t_4$: ILPP 1 has a solution $\overrightarrow{\nu}_u = \overrightarrow{1}$. Thus we just need to increase the observed sequence, i.e., $\nu = t_1 t_2 t_3 t_4 t_5 t_6$.

Occurrence of the second $t_6$: A solution to ILPP 1 is $\overrightarrow{\nu}_u = [0, 1, 1]^T$, implying that $t_6$ is enabled after firing unobservable transition $\varepsilon_3$. The vector $\overrightarrow{y}$ is updated with $\overrightarrow{y} = \overrightarrow{y} + \overrightarrow{\nu}_u = [1, 2]^T$ and let $\nu = t_1 t_2 t_3 t_4 t_5 t_6$.

Occurrence of the third $t_7$: ILPP 1 has no feasible solution and a solution $\overrightarrow{\nu}_3 = [0, 1, 0, 0, 0, 0, -1]^T$ is found by solving ILPP 3. We add $\varepsilon_3$ 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 $\nu = t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8 t_9 t_{10}$.

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 $M_o$, 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 $C_o$ is null. If transition $t$ is observed, then the relation $C_o \cdot \overrightarrow{v} \geq Pr_{e_o}(v, t)$ is true, i.e., the relation $M_0 \geq Pr_{e_o}(v, t)$. If it holds, let $\omega = \nu = t$, and $\nu \in \Omega(PN, M_0)$ trivially holds. Otherwise, the solution $\overrightarrow{\nu}_o$ is obtained by solving ILPP 3 and $PN_{o}$ is updated by adding an unobservable transition $\varepsilon_3$. Let $\omega = t$ and $PN = PN_o \otimes PN_u$. By Proposition 11, $M_o[\varepsilon_3] M_{t}[t]$ is trivially verified, namely, $\nu \in \Omega(PN, M_o)$.

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

1. $n_o$ equals 0, i.e., no unobservable transition is introduced during the period of observation $v$. If $M_0 + C_o \cdot \overrightarrow{v} \geq Pr_{e_o}(v, t)$, then $t$ is enabled at the current marking, i.e., $\nu = \nu \in \Omega(PN, M_0)$. Otherwise, a solution $\overrightarrow{\nu}_o$ is found by solving ILPP 3 and the structure of $PN$ is updated by adding a new unobservable transition $\varepsilon_3$ whose incidence matrix column is $\overrightarrow{\nu}_o$. Let $\omega = \nu \cdot t$. By Proposition 11, $M_o[\varepsilon_3] M_{t}[t]$ holds, namely, $\nu \in \Omega(PN, M_o)$.

2. $n_o$ is greater than 0. In this situation, if ILPP 1 admits a solution, we update $\overrightarrow{y}$ and the observed sequence with $\omega = \nu \cdot t$. By Theorem 7, $\omega \in \Omega(PN, M_0)$ holds. On the other hand, if ILPP 1 has no feasible solution, the solution including $\overrightarrow{\nu}_o$ and $\overrightarrow{\nu}_o$ is obtained by solving ILPP 3. We update the related variables with $\overrightarrow{y} = \overrightarrow{y} + \overrightarrow{\nu}_o$ and $C_u = [C_u \overrightarrow{\nu}_o]$ and add a new unobservable transition $\varepsilon_3$ to Petri net $PN$. Let $\omega = \nu \cdot t$, and then $\omega \in \Omega(PN, M_0)$ 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 $\overrightarrow{\nu}_o$, $\overrightarrow{\nu}_z$, and $\overrightarrow{\nu}$ are the integer variables in ILPP 3 and $S$ is a binary variable. Then the number of integer variables is $n_o + 3m$, where $n_o$ denotes the number of unobservable transitions and $3m$ denotes the sum of the dimensions of $\overrightarrow{\nu}_o$, $\overrightarrow{\nu}$, 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 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 $(PN, M_0)$, where $PN = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Assume that $P_o = \{P_{o_1}, P_{o_2}, \ldots, P_{o_l}\} \subseteq P$ 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 $M^{P_o}$, and $M|_{po} = M^{P_o}$ denotes that the restriction of $M$ to $P_o$ is $M^{P_o}$.

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 \( w = t_1 t_2 \ldots t_k \) with \( k \geq 1 \) and the corresponding reached partial markings by \( M_1^p M_2^p \ldots M_k^p \), where \( t_\beta \in T_o \) and \( M_i^p \) is the observed partial marking after transition \( t_\beta \) for \( i = 1, \ldots, k \).

**Example 15.** Consider again the observable subnet shown in Figure 4(a). If the set of observable places is \( P_o = \{ P_1, P_2, P_3 \} \), and the initial marking \( M_i \) is given and \( P_o = \{ P_1, P_2, P_3 \} \subseteq P \) with \( h \leq m \) is the set of observable places. The output observed includes the transition sequence \( w = t_1 t_2 \ldots t_k \) with \( k \geq 1 \) and the corresponding reached partial markings \( M_1^p M_2^p \ldots M_k^p \), where \( t_\beta \in T_o \) and \( M_i^p \) is the observed partial marking after transition \( t_\beta \) for \( i = 1, \ldots, k \). The problem consists in determining a Petri net \( PN \) with \( PN = PN_o \otimes PN_a \) such that \( w \in \Omega(PN, M_i) \) and each marking \( M_i \in R(PN, M_i) \) after transition \( t_\beta \) satisfies \( M_i \mid p_o = M_i^p \) for \( i = 1, \ldots, k \).

**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 \( (PN, M_o) \). Assume that \( P_o = \{ P_1, P_2, \ldots, P_h \} \) 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 \( t_\beta \) and the corresponding partial markings \( M_i^p \) are observed. If (17a), (17b), and (17c) admit a solution, then \( v t_\beta \in \Omega(PN, M_o) \) and the marking\( M_i \) after transition \( t_\beta \) fires satisfies \( M_i \mid p_o = M_i^p \).

\[
M_0 + C_o \cdot \vec{v} + C_u \cdot \vec{y} + C_u \cdot \vec{u} \geq Pre_o (\cdot, t_\beta) 
\]

(17a)

\[
A \cdot (M_0 + C_o \cdot \vec{v} + C_u \cdot \vec{y} + C_u \cdot \vec{u} + C_o (\cdot, t_\beta)) = M_i^p
\]

(17b)

\[
\vec{u} \in N^o
\]

(17c)

where \( A \) is an \( h \times m \) matrix such that \( A(i, \alpha) = 1 \) for \( i = 1 \ldots h \) and \( 0 \) otherwise. The matrix \( A \) is dedicated to projecting a marking \( M \in R(PN, M_o) \) 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, \( vt_\beta \in \Omega(PN, M_o) \) is trivially verified. However, if constraint (17b) is removed, there may be numerous solutions that satisfy \( vt_\beta \in \Omega(PN, M_o) \) but violate \( M_k \mid p_o = M_k^p \). When the set of observable places is taken into account, constraint (17b) guarantees that these solutions that satisfy \( M_k \mid p_o = M_k^p \) 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 \( \vec{1}_m \cdot \vec{u} \) and obtain the following integer linear programming problem that minimizes the considered performance index:

\[
\text{ILPP 4:} \quad \begin{cases} 
\min & \vec{1}_m^T \cdot \vec{u} \\ 
s.t. & \text{Eq. (17a), (17b), and (17c)}
\end{cases}
\]

**Proposition 18.** Consider net system \( (PN, M_o) \). 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 \( t_\beta \) and the corresponding partial marking \( M_k^p \) are observed. If there exists a vector \( \vec{z} \) satisfying (19a), (19b), (19c), (19d), and (19e), then \( vt_\beta \in \Omega(PN, M_o) \) and the marking \( M_k \) after transition \( t_\beta \) fires satisfies \( M_k \mid p_o = M_k^p \), where \( PN' \) is a Petri net obtained by adding \( e_z \) to \( PN \).

\[
M_0 + C_o \cdot \vec{v} + C_u \cdot (\vec{y} + \vec{u}) + \vec{z} \geq Pre_o (\cdot, t_\beta)
\]

(19a)

\[
A \cdot (M_0 + C_o \cdot \vec{v} + C_u \cdot (\vec{y} + \vec{u}) + \vec{z}) + C_o (\cdot, t_\beta) = M_k^p
\]

(19b)

\[
\vec{u} \in N^o
\]

(19c)

\[
\vec{z} \in \mathbb{Z}^n
\]

(19d)

\[
\vec{z} \neq 0
\]

(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, \( vt_\beta \in \Omega(PN', M_o) \) is trivially verified. On the other hand, constraint (19b) guarantees that the solutions satisfying \( M_k \mid p_o = M_k^p \) 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 \( \vec{1}_m \cdot \abs{\vec{z}} \) as the performance index and obtain the following programming problem:

\[
\text{ILPP 5:} \quad \begin{cases} 
\min & \vec{1}_m^T \cdot \abs{\vec{z}} \\ 
s.t. & \text{Eq. (19a), (19b), (19c), (19d), and (19e)}
\end{cases}
\]

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 analyzed 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
