Critical Observability of Labeled Time Petri Net Systems

The full bibliographic citation of the final version: X. Cong, M. P. Fanti, A. M. Mangini and Z. Li, "Critical Observability of Labeled Time Petri Net Systems," in IEEE Transactions on Automation Science and Engineering, vol. 20, no. 3, pp. 2063-2074, July 2023, doi: 10.1109/TASE.2022.3193493.

Critical Observability of Labeled Time Petri Net Systems

Xuya Cong, Maria Pia Fanti, *Fellow, IEEE,* Agostino Marcello Mangini, *Member, IEEE* and Zhiwu Li, *Fellow, IEEE*

***Abstract*—A time Petri net is said to be critically observable at a given time instant if the markings consistent with any observation at the time instant are included either in the set of critical markings or non-critical markings. This work studies the verification problem of critical observability of timed discrete event systems modeled by bounded labeled time Petri nets. The proposed method is a two-fold process: a preliminary verification of critical observability of the underlying logic labeled Petri net and a further verification considering the time constraint associated with each transition. The first step is based on the concurrent composition of a reachability graph of the logic net. If the logic net is critically observable, then the time net is also critically observable at any given time instant. Otherwise, the second step is to design an algorithm to compute all pairs of transition-class sequences that violate critical observability at the given time instant, and then a set of linear programming problems is exploited to check critical observability for the corresponding timed system.**

***Note to Practitioners*—Timed discrete event systems provide a theoretical model for safety-critical real applications such as air traffic management and industrial control, which are vulnerable to malicious attack and destruction at some particular time instants such that a system may be misled to dangerous states. Critical observability of a timed discrete event system is a property with which the predefined dangerous states of the system can be determined and detected from the observation by an observer at a given time instant. This research aims to offer a systematic approach to check critical observability for timed discrete event systems modeled by bounded labeled time Petri nets, avoiding the enumeration of the full state space of a real-world system with the time information.**

***Index Terms*—Discrete event system, time Petri net, critical observability.**

I. Introduction

In the last decade, cyber-physical systems have received extensive attention from both academic and industrial communities. In particular, air traffic, cyber-security, and highly automated production systems are some of the important safety-critical applications that are usually required to verify whether their evolution is associated with a critical or undesirable procedure. In order to formally check this safety-critical property in discrete event systems (DESs), a notion called critical observability is initially proposed by Santis *et al.* in [29].

Automata and Petri nets (PNs) are two mathematical tools to model, analyze and control DESs. In a recent work [24], the author investigates the computational complexity of verifying critical observability for (networks of) finite-state automata and PNs. The work in [32] touches upon critical observability at a given time step and proposes a method to find all possible inputs moving a finite-state automaton to a critical state of interest. Moreover, the work [8] handles the problem of critical observability and checks this property for live and bounded labeled Petri nets (LPNs) by using integer linear programming solutions, thus avoiding the computation of the full reachability graph.

The critical observability problem is related to the state estimation [12] that has been widely investigated in the field of logic DESs. In many real applications, the factor of time is necessary and significant for the evolution and verification of systems such as automated manufacturing systems [14, 18, 33], healthcare systems [36], commercial systems [22], and intelligent transportation systems [15]. In order to model real systems with the time information, researchers usually use timed DESs to describe their behaviours. Due to the explicit consideration of time, the state estimation related problems of timed DESs become much more complex. However, few works address this issue in timed DESs modeled by automata [10, 17, 35] and PNs [1, 3, 11, 27].

In the PN framework, there are two types of net systems with time information called time PNs [28] and timed PNs [25]. In time PNs, enabled transitions can fire within the given time intervals that may either be associated with places or transitions (P-time PNs [6] and T-time PNs [1], respectively). In timed PNs, enabled transitions fire as soon as the given time delays have elapsed. Analogously, if the time delays are associated with places (respectively, transitions), the net systems are called P-timed PNs [26] (respectively, T-timed PNs [9]).

This paper considers the problem of critical observability for bounded labeled T-time PNs (TPNs): given a set of critical markings representing some undesirable states, a system is said to be critically observable at a time instant if the set of markings consistent with any observation of the system at the time instant is either a subset of critical markings or a subset of non-critical markings. In practice, a critically observable

This work is supported by the National Key R&D Project of China under Grant No. 2018YFB1700104 (*Corresponding author: Zhiwu Li*).

X. Cong is with the College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China (email: congxuya@163.com).

M. Fanti and A. Mangini are with the Department of Electrical and Information Engineering, Polytechnic of Bari, 70125 Bari, Italy (email: mariapia.fanti@poliba.it, agostinomarcello.mangini@poliba.it).

Z. Li is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China and also with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau (email: zhwli@xidian.edu.cn).

Corresponding author: zhwli@xidian.edu.cn.

timed system requires that its undesirable operations or faults should be identified immediately since the faults may lead to some catastrophic results.

In order to deal with this problem, we first use a structure called a twin reachability graph (TRG) that is the concurrent composition of a reachability graph (RG) [37] of the logic net system. In particular, we prove that if the underlying logic PN is critically observable, the corresponding TPN is also critically observable at any given time instant. Otherwise, if the logic PN is not critically observable, it is possible that the system becomes critically observable when time constraints are considered. For the sake of checking critical observability in a labeled TPN framework, we construct some particular pairs of transition-class sequences (TCSs) of the *Modified State Class Graph* (MSCG) [1, 2, 13]. Specifically, an MSCG is a graph whose nodes are the state classes of the TPN and edges are labeled with transitions, labels, and timing constraints associated with transitions. Then, given a time instant, for each pair of TCSs that is feasible at this time instant but one ends in a class containing a critical marking and the other does not, a linear programming problem (LPP) verifies whether the two sequences in the pair can generate the same observations at the same time instants. In such a case, the system is not critically observable.

In the literature, there exist few works dealing with state estimation, fault diagnosis, and diagnosability for bounded labeled TPNs [1, 2, 13] by constructing the full MSCGs. However, these methods do not study how to find all the pairs of TCSs leading to marking pairs that violate critical observability, and hence these methods cannot be applied to verify critical observability for bounded labeled TPNs in general. In addition, by using the linear algebraic method, the work [21] solves the online marking estimation problem of a class of bounded and unbounded labeled TPNs without computing any graph of the timed systems, i.e., it determines the markings set consistent with a given observation. However, there are some strict structure constraints on the system in [21]. First, it requires that the upper bound of time interval for each transition is infinite. Second, the unobservable subnet of the system needs to be backward-conflict-free, and each conflict place cannot have unobservable output transitions. As far as we know, this work is the first one introducing the notion of critical observability and verifying this property in the framework of bounded labeled TPNs without computing the full MSCG.

The structure of the work is outlined as follows. Section II presents the basics of PNs and labeled TPNs. Section III provides some preliminary results to check critical observability for bounded labeled TPNs. In Section IV, a formal method is proposed to verify critical observability of bounded labeled TPNs, and in Section V two examples illustrate the proposed approach. Finally, Section VI concludes the paper.

II. Petri Nets and Labeled Time Petri Nets

A. PN and LPN

Let $\mathbb{N}$ be a set of non-negative integers. A Petri net structure is a four-tuple $PN = (P, T, Pre, Post)$, where P is a set of m places, T is a set of n transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$, $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the pre- and post-incidence functions that specify the arcs from places to transitions and transitions to places in the net, respectively. Specifically, $Pre(p,t) = x$ $(x > 0)$ if there is an arc from place p to transition t with weight x; $Pre(p,t) = 0$ if there is no arc from p to t. $Post(p,t) = x$ $(x > 0)$ if there is an arc from transition t to place p with weight x; $Post(p,t) = 0$ if there is no arc from t to p. For the sake of computation, Pre ($Post$) can be represented by an $m \times n$ matrix indexed by P and T. The incidence matrix C of the PN is defined as $C = Post - Pre$.

A marking is a function $M : P \to \mathbb{N}$ that assigns to each place a non-negative integer number of tokens. Similarly, a marking M can be represented by an m-dimensional vector indexed by P. The marking of place p at M is denoted as $M(p)$. For the sake of brevity, a marking can be written as the sum of tokens of all places in P as $M = \sum_{p \in P} M(p)p$. A PN system $\langle PN, M_0 \rangle$ is a net PN with an initial marking M_0. A transition t is said to be logically enabled at M if $M \geq Pre(\cdot, t)$, and the set of transitions enabled at M is denoted as $\mathcal{A}(M)$. The firing of a transition t at M leads to marking $M' = M + C(\cdot, t)$, which is denoted as $M[t\rangle M'$.

An LPN system is a four-tuple $G = (PN, M_0, E, \lambda)$, where $\langle PN, M_0 \rangle$ is a PN system, E is an alphabet (a set of labels) and $\lambda : T \to E \cup \{\varepsilon\}$ is a labeling function that assigns to each transition $t \in T$ either a symbol $\gamma_i \in E$ or the empty word ε. Moreover, the set of transitions can be partitioned into $T = T_o \cup T_u$ with $T_o \cap T_u = \emptyset$, where T_o (resp. T_u) is the set of n_o (resp. n_u) observable (resp. unobservable) transitions. The labeling function λ satisfies the following statement: if $t \in T_o$, then $\lambda(t) = \gamma_i \in E$; if $t \in T_u$, then $\lambda(t) = \varepsilon$. In an LPN, the same label $\gamma_i \in E$ can be associated with n_i $(n_i \geq 1)$ transitions.

A marking M is said to be reachable from M_0 if there exist a transition sequence $\sigma = t_1 t_2 \dots t_k$ and markings $M_1, M_2, \dots, M_{k-1}$ such that $M_0[t_1\rangle M_1[t_2\rangle M_2 \dots M_{k-1}[t_k\rangle M$ holds, denoted as $M_0[\sigma\rangle M$. We write it as $M_0[\sigma\rangle$ if the destination marking is of no interest. The set of all reachable markings from M_0 is denoted as $R(G)$. An LPN system G is said to be bounded if the set $R(G)$ is finite. An evolution of G from $M \in R(G)$ is defined as a transition-marking sequence $Mt_{\alpha_1}M_1t_{\alpha_2}M_2 \dots t_{\alpha_h}M_h$ satisfying $M[t_{\alpha_1}\rangle M_1[t_{\alpha_2}\rangle M_2 \dots [t_{\alpha_h}\rangle M_h$, where $\alpha_i \in \{1, 2, \dots, n\}$, $t_{\alpha_i} \in T$, $M_i \in R(G)$, $i = 1, 2, \dots, h$. The set of all transition sequences that can fire in G is defined as $L(G) = \{\sigma \in T^* | M_0[\sigma\rangle\}$, where T^* is the *Kleene-closure* [7] of T.

For a bounded LPN, its RG $\mathcal{R} = (\mathcal{M}, \Sigma, f, M_0)$ is a deterministic finite-state automaton, where $\mathcal{M} = R(G)$ is the set of states, $\Sigma \subseteq T \times (E \cup \{\varepsilon\})$ is the set of events, $f : \mathcal{M} \times \Sigma \to \mathcal{M}$ is the transition function, M_0 is the initial state. In particular, $f(M, (t, \lambda(t))) = M'$ if $M[t\rangle M'$ holds.

Let w denote the sequence of labels associated with a transition sequence $\sigma \in T^*$ such that $w = \lambda(\sigma)$ by using the extended form of the labeling function $\lambda : T^* \to E^*$. Given an LPN system $G = (PN, M_0, E, \lambda)$, we define the language generated by G as $\mathcal{L}(G) = \{w \in E^* | \exists \sigma \in L(G) : \lambda(\sigma)$

$= w\}$. For an observed word $w \in \mathcal{L}(G)$, let $\mathcal{S}(w) = \{\sigma \in L(G)|\lambda(\sigma) = w\}$ and $\mathcal{C}(w) = \{M \in \mathbb{N}^m|\exists\sigma \in \mathcal{S}(w) : M_0[\sigma\rangle M\}$ denote the set of firing sequences consistent with w and the set of markings consistent with w, respectively.

B. Labeled TPN

In this subsection, the basics about labeled TPN [1, 28] are recalled.

A labeled TPN system is a five-tuple $G_t = (PN, M_0, E, \lambda, Q)$, where (PN, M_0, E, λ) denotes an LPN system and $Q : T \to \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$ defines the set of static intervals associated with transitions. More precisely, Q associates two non-negative real numbers with a transition $t_i \in T$, namely $Q(t_i) = (l_i, u_i)$, where $l_i \geq 0$, $u_i \geq l_i$, $l_i \neq \infty$. Then, if t_i remains logically enabled during the time interval $[l_i, u_i]$, then t_i may fire.

A *time-transition sequence* is a sequence of couples of transitions and time instants, which specifies the sequence of transitions and the corresponding time instants. Specifically, the firing of a feasible sequence $\sigma_t = (t_{i_1}, \tau_1)(t_{i_2}, \tau_2)\dots(t_{i_k}, \tau_k) \in (T \times \mathbb{R}_{\geq 0})^*$ at M_0 means that transition t_{i_j} fires at time τ_j for $j = 1, 2, \dots, k$, i.e., $M_0[t_{i_1}(\tau_1)\rangle M_1[t_{i_2}(\tau_2)\rangle M_2 \dots M_{k-1}[t_{i_k}(\tau_k)\rangle M_k$ or $M_0[\sigma_t\rangle M_k$, where $\tau_1 \leq \tau_2 \leq \dots \leq \tau_k$. Here, τ_i for $i = 1, 2, ..., k$ is used to denote the absolute time determined by a global clock. We write it as $M_0[\sigma_t\rangle$ if the destination marking is of no interest. Let $\tau_l(\sigma_t)$ denote the time instant of the firing of the last transition in σ_t, i.e., $\tau_l(\sigma_t) = \tau_k$. The set of all time-transition sequences that can fire in a labeled TPN system is denoted as $L(G_t) = \{\sigma_t \in (T \times \mathbb{R}_{\geq 0})^*|M_0[\sigma_t\rangle\}$.

A marking M is said to be reachable from M_0 if there exists a time-transition sequence σ_t such that $M_0[\sigma_t\rangle M$. The time reachability set $R_t(G_t)$ is the set of markings reachable from M_0. A labeled TPN system G_t is said to be bounded if the set $R_t(G_t)$ is finite.

We assume that the considered labeled TPNs follow a single server and strong semantics [4, 23] and an enabling memory policy [31], which are commonly adopted in the state-estimation, supervisory control, fault diagnosis and diagnosability of labeled TPNs [1, 2, 19, 20, 27].

The enabling degree of transition t_i enabled at a marking M_j, denoted by $\alpha_i(j)$, is the biggest integer number φ such that $M_j \geq \varphi \cdot Pre(\cdot, t_i)$. The single server semantics specifies one clock to an enabled transition t_i with an arbitrary enabling degree $\alpha_i(j)$, i.e., t_i can only fire once each time at M_j regardless of the value of $\alpha_i(j)$. Given a firing interval $[l_i, u_i]$ of an enabled transition t_i, strong semantics implies that t_i must fire within its firing interval unless it is disabled. The enabling memory policy indicates that there is no memory of any previous enabling for a transition.

The time labeling function λ_t of a labeled TPN assigns to each time-transition pair $(t, \tau) \in T \times \mathbb{R}_{\geq 0}$ either a time-label pair $(e, \tau) \in E \times \mathbb{R}_{\geq 0}$ or the empty word ε. In particular, if $\lambda(t) = \gamma_i \in E$, then $\lambda_t((t, \tau)) = (\gamma_i, \tau)$; if $\lambda(t) = \varepsilon$, then $\lambda_t((t, \tau)) = \varepsilon$. Let δ_o denote the *time-label sequence* (TLS) associated with a time-transition sequence σ_t such that $\delta_o = \lambda_t(\sigma_t)$ by using the extended form of the time labeling function $\lambda_t : (T \times \mathbb{R}_{\geq 0})^* \to (E \times \mathbb{R}_{\geq 0})^*$.

Let $\tau_l(\delta_o)$ denote the time instant of the occurrence of the last label in δ_o. The set of TLSs generated by a labeled TPN system is defined as $\mathcal{L}(G_t) = \{\delta_o \in (E \times \mathbb{R}_{\geq 0})^*|\exists\sigma_t \in L(G_t) : \lambda_t(\sigma_t) = \delta_o\}$. Given a TLS $\delta_o = (\gamma_{e_1}, \tau_1)(\gamma_{e_2}, \tau_2)\dots(\gamma_{e_h}, \tau_h) \in (E \times \mathbb{R}_{\geq 0})^*$, $\log(\delta_o) = \gamma_{e_1}\gamma_{e_2}\dots\gamma_{e_h}$ denotes the "logic" sequence of labels associated with δ_o, neglecting the time instants at which they have fired.

Given a TLS $\delta_o \in (E \times \mathbb{R}_{\geq 0})^*$ and a time instant $\tau \geq \tau_l(\delta_o)$, the following two definitions are given.

Definition 1: [1] Given a TLS $\delta_o \in (E \times \mathbb{R}_{\geq 0})^*$ and a time instant $\tau \geq \tau_l(\delta_o)$, the set of time-transition sequences consistent with δ_o and τ is $\Sigma(\delta_o, \tau) = \{\sigma_t \in (T \times \mathbb{R}_{\geq 0})^*|\exists M \in R_t(G_t), M_0[\sigma_t\rangle M, \lambda_t(\sigma_t) = \delta_o, \tau_l(\sigma_t) = \bar{\tau}$, with $\bar{\tau} \leq \tau$, $\nexists t \in T_u : M[t\rangle$ and $r_t(M_0, \sigma_t) \leq \tau - \bar{\tau}\}$, where $r_t(M_0, \sigma_t)$ denotes the residual time[2] of t after firing σ_t at M_0, i.e., the amount of time within which the unobservable transition t surely fires at M.

Definition 2: [1] Given a TLS $\delta_o \in (E \times \mathbb{R}_{\geq 0})^*$ and a time instant $\tau \geq \tau_l(\delta_o)$, the set of markings consistent with δ_o and τ is $\mathcal{C}(\delta_o, \tau) = \{M \in \mathbb{N}^m|\exists\sigma_t \in \Sigma(\delta_o, \tau) : M_0[\sigma_t\rangle M\}$.

In particular, given a labeled TPN system $G_t = (PN, M_0, E, \lambda, Q)$, we say that a TLS δ_o is feasible in G_t at a time instant τ, if there exists at least one time-transition sequence that is consistent with the TLS and the time instant. In the following, we use $\mathcal{L}(G_t, \tau)$ to denote the set of all TLSs that are feasible in G_t at the time instant τ.

C. MSCG and TCS

Now, we briefly introduce some basics of the *Modified State Class Graph* (MSCG) [13] that is a directed graph whose nodes are called classes. In particular, each class represents a state of a labeled TPN system $G_t = (PN, M_0, E, \lambda, Q)$, namely a reachable marking $M \in R_t(G_t)$ and a conjunction of inequalities Θ defining the time (firing) domains associated with all transitions enabled at M. More precisely, a state of G_t is a pair (M_k, Θ_k), where M_k is a reachable marking and Θ_k is a conjunction of x_k inequalities $\bar{l}_{k_i} \leq \theta_{k_i} \leq \bar{u}_{k_i}$, $i = 1, \dots, x_k$, where x_k is the number of transitions enabled at M_k. Actually, it holds $\bar{l}_{k_i} \leq l_{k_i}$ and $\bar{u}_{k_i} \leq u_{k_i}$, where $(l_{k_i}, u_{k_i}) = Q(t_{k_i})$ for all $i = 1, \dots, x_k$. It is obvious that $\bar{l}_{k_i} = l_{k_i}$ and $\bar{u}_{k_i} = u_{k_i}$ when t_{k_i} has just been enabled at marking M_k. In general, the bounds $\bar{l}_{k_i}$ and $\bar{u}_{k_i}$ are characterized in linear algebraic terms as a function of a series of parameters related to the time elapsed of some previous fired transitions. Moreover, an edge of an MSCG is labeled as $(t, \gamma, \Delta \in [l^*, u^*])$, where $t \in T$ is the transition whose firing yields the marking in the tail node to the marking in the head node, $\gamma = \lambda(t)$ is the label of transition t, and $\Delta \in [l^*, u^*]$ is a time (firing) domain of transition t with $l^* \leq l$ and $u^* \leq u$, where l and u are the lower and upper bounds of the static interval associated with t, respectively. Then, an evolution of $G_t = (PN, M_0, E, \lambda, Q)$ from a generic marking $M \in R_t(G_t)$ is defined as a TCS: $(M, \Theta)(t_{\alpha_1}, \lambda(t_{\alpha_1}), \Delta_{(1)} \in [l^*_{\alpha_1}, u^*_{\alpha_1}])(M_1, \Theta_1)\ (t_{\alpha_2}, \lambda(t_{\alpha_2}), \Delta_{(2)} \in [l^*_{\alpha_2}, u^*_{\alpha_2}]) \dots (t_{\alpha_L},$

[2]The residual time of a transition t is a function of the particular evolution that leads to the current marking, i.e., it is a function of M_0 and σ_t.

$\lambda(t_{\alpha_L}), \Delta_{(L)} \in [l^*_{\alpha_L}, u^*_{\alpha_L}])(M_L, \Theta_L)$ that can represent a generic path in an MSCG.

III. Preliminary results

In this section, we provide a formal definition about critical observability of the bounded LPNs and exploit a structure called a twin reachability graph (TRG) using concurrent composition of an RG [37] to check critical observability of the bounded LPNs.

Definition 3: Let $G = (PN, M_0, E, \lambda)$ be a bounded LPN system and C_R be a set of critical markings. G is said to be critically observable if $[\mathcal{C}(w) \subseteq C_R] \vee [\mathcal{C}(w) \subseteq R(G) \setminus C_R]$ for all words $w \in \mathcal{L}(G)$.

By Definition 3, a bounded LPN system is not critically observable if there exist two firing sequences consistent with an observed word $w \in \mathcal{L}(G)$ such that one of the sequences leads to a critical marking and the other leads to a non-critical marking.

Let $\mathcal{R} = (\mathcal{M}, \Sigma, f, M_0)$ be an RG of a bounded LPN system. We construct the TRG of $\mathcal{R}$ as a deterministic finite-state automaton $V = (\mathcal{M}^V, \Sigma^V, f^V, M_0^V)$ as follows:

1) $\mathcal{M}^V \subseteq \mathcal{M} \times \mathcal{M}$;
2) $\Sigma^V = \Sigma_o^V \cup \Sigma_u^V$, where $\Sigma_o^V = \{(t_a, t_b) | t_a, t_b \in T_o, \lambda(t_a) = \lambda(t_b)\}$, $\Sigma_u^V = \{(t_a, \varepsilon) | t_a \in T_u\} \cup \{(\varepsilon, t_b) | t_b \in T_u\}$;
3) $M_0^V = (M_0, M_0)$;
4) for all $(M_1', M_1), (M_2', M_2) \in \mathcal{M}^V$, $(t_a, t_b) \in \Sigma_o^V$, $(t_a, \varepsilon) \in \Sigma_u^V$, and $(\varepsilon, t_b) \in \Sigma_u^V$,
   - $((M_1', M_1)(t_a, t_b)(M_2', M_2) \in f^V$ if $(M_1', (t_a, \lambda(t_a)), M_2'), (M_1, (t_b, \lambda(t_b)), M_2) \in f$,
   - $((M_1', M_1)(t_a, \varepsilon)(M_2', M_2)) \in f^V$ if $(M_1', (t_a, \lambda(t_a)), M_2') \in f$, $M_1 = M_2$,
   - $((M_1', M_1)(\varepsilon, t_b)(M_2', M_2)) \in f^V$ if $M_1' = M_2'$, $(M_1, (t_b, \lambda(t_b)), M_2) \in f$.

By extending the transition function f^V to the domain $\mathcal{M}^V \times (\Sigma^V)^*$, the language of V is defined as $\mathcal{L}(V) = \{(\sigma^1, \sigma^2) \in (\Sigma^V)^* | \exists M^V \in \mathcal{M}^V : f^V(M_0^V, (\sigma^1, \sigma^2)) = M^V\}$. Intuitively, based on the construction of the TRG V, V tracks all pairs of transition sequences with the same observation; namely, for any $(\sigma^1, \sigma^2) \in \mathcal{L}(V)$, we have $\lambda(\sigma^1) = \lambda(\sigma^2)$, and for any $\sigma^1, \sigma^2 \in L(G)$ with $\lambda(\sigma^1) = \lambda(\sigma^2)$, we have $(\sigma^1, \sigma^2) \in \mathcal{L}(V)$.

Definition 4: Given a bounded LPN G, let $\mathcal{R} = (\mathcal{M}, \Sigma, f, M_0)$ be its RG and $V = (\mathcal{M}^V, \Sigma^V, f^V, M_0^V)$ be the TRG of $\mathcal{R}$. We define a critically-bad state as a pair of markings $M^V = (M', M) \in \mathcal{M}^V$ with $[M' \in C_R \wedge M \notin C_R]$. The set of critically-bad states in V is denoted as $\mathcal{M}_C^V$.

Definition 5: [38] A generalized mutual exclusion constraint (GMEC) is a pair $(\mathbf{w}, k)$, where $\mathbf{w} \in \mathbb{Z}^m$ and $k \in \mathbb{Z}$, that defines a set of markings:

$$\mathcal{L}_{(\mathbf{w},k)} = \{M \in \mathbb{N}^m | \mathbf{w}^T \cdot M \leq k\}.$$

An OR-AND GMEC is a set $W = \{(\mathbf{W}_1, \mathbf{k}_1), \ldots, (\mathbf{W}_r, \mathbf{k}_r)\}$, where $\mathbf{W}_i \in \mathbb{Z}^{m \times s_i}$ and $\mathbf{k}_i \in \mathbb{Z}^{s_i}$ for $i \in \{1, \ldots, r\}$, that defines a set of markings:

$$\mathcal{L}_W = \{M \in \mathbb{N}^m | (\exists i \in \{1, \ldots r\}) \, \mathbf{W}_i^T \cdot M \leq \mathbf{k}_i\}.$$

In this paper, a critical markings set C_R can be described by an arbitrary set of markings, or a set of GMECs $\mathcal{L}_W$ in Definition 5 if the set C_R is huge.

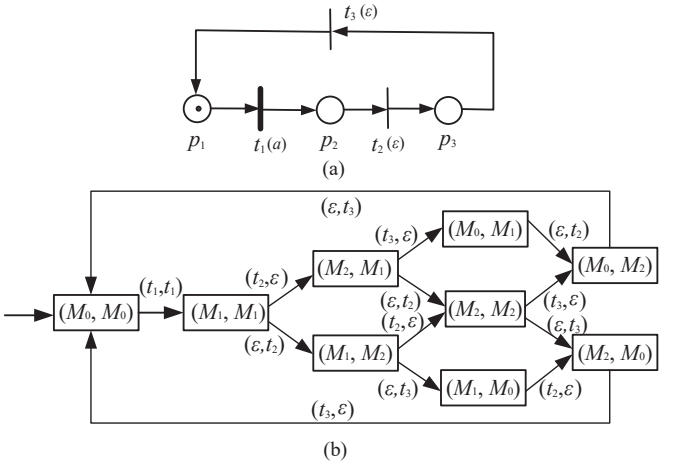

Fig. 1: (a) An LPN system $G = (PN, M_0, E, \lambda)$ and (b) its TRG V.

Example 1: Let us consider the LPN system G in Fig. 1(a) with $M_0 = p_1$, which can be used to represent a cyber-security system affected by an attack. In fact, this example can be seen as an abstracted model of a smart grid, which consists of the intrusion detection system and the firewall. The net has three reachable markings $M_0 = p_1$, $M_1 = p_2$, and $M_2 = p_3$. The set of its observable transitions is $T_o = \{t_1\}$ with $\lambda(t_1) = a$. The meaning of places and transitions in the LPN in Fig. 1(a) is shown in Table I. Fig. 1(b) depicts its TRG V. Given a set of critical markings $C_R = \{M \in \mathbb{N}^3 | -M(p_1) \leq -1 \vee -M(p_2) \leq -1\}$, there exist two critically-bad states (M_0, M_2) and (M_1, M_2) in V as shown in Fig. 1(b). Actually, the existence of the two critically-bad states means that the observer cannot always determine whether the system is normal or abnormal. □

TABLE I: The meaning of places and transitions in the LPN in Fig. 1(a)

p_1	the system is under attack
p_2	the attack is detected but the system is still abnormal
p_3	the system is normal
t_1	a firewall detects the attack
t_2	a firewall starts a block action that is unobservable
t_3	a malicious software attacks the system

Remark 1: Based on the construction of a TRG V, the existence (non-existence) of any critically-bad state (M_i, M_j) in V implies the existence (non-existence) of state (M_j, M_i) in V. Since both (M_i, M_j) and (M_j, M_i) correspond to the same set $\{M_i, M_j\} \subseteq \mathcal{C}(w)$, only (M_i, M_j) is selected as a critically-bad state that violates critical observability.

In the following, we formalize the problem of critical observability in the framework of labeled TPNs. To prove the main results about the critical observability, two assumptions are made:

A1) The considered labeled TPN is bounded.
A2) The sum of the lower bounds in the cycles of transitions (observable and unobservable) of the labeled TPN is not equal to zero.

Assumption A1 guarantees that the RG of the logic net system is finite. In this case, the critical observability of the logic net system is decidable [24]. Note that Assumption A1 is commonly used in the literature for state estimation [1], supervisory control [19, 20], diagnosis and diagnosability [2, 13, 27] of labeled TPNs. Assumption A2 is also called the zeno-free assumption [2, 13, 27] that is necessary for Algorithm 2, which will be discussed in Section IV.B.

In the framework of labeled TPN systems, a critical set is modeled as a set of markings C_{R_t}. Then, we define the critical observability of labeled TPN systems.

Definition 6: Let $G_t = (PN, M_0, E, \lambda, Q)$ be a labeled TPN system and C_{R_t} be a set of critical markings. Given a time instant $\tau \in \mathbb{R}_{\geq 0}$, G_t is said to be critically observable at the time instant τ if $[\mathcal{C}(\delta_o, \tau) \subseteq C_{R_t}] \vee [\mathcal{C}(\delta_o, \tau) \subseteq R_t(G_t) \setminus C_{R_t}]$ for all TLSs $\delta_o \in \mathcal{L}(G_t, \tau)$.

Proposition 1: Let $G_t = (PN, M_0, E, \lambda, Q)$ be a labeled TPN system and C_{R_t} be a set of critical markings. Let $G = (PN, M_0, E, \lambda)$ be the corresponding logic LPN system and C_R with $C_R = C_{R_t}$ being a set of critical markings. If there is no critically-bad state in the TRG V of G, then G_t is critically observable at any time instant $\tau \in \mathbb{R}_{\geq 0}$.

Proof: The result follows from the fact that the additional time information in a labeled TPN system can only disable sequences that are enabled in the corresponding logic LPN system. Thus, given a labeled TPN system $G_t = (PN, M_0, E, \lambda, Q)$ with any time instant $\tau \in \mathbb{R}_{\geq 0}$ and its corresponding logic LPN system $G = (PN, M_0, E, \lambda)$, for each $\delta_o \in \mathcal{L}(G_t, \tau)$ and $w \in \mathcal{L}(G)$ with $\log(\delta_o) = w$, it is obvious that $\Sigma(\delta_o, \tau) \subseteq \mathcal{S}(w)$ and $\mathcal{C}(\delta_o, \tau) \subseteq \mathcal{C}(w)$ hold. Since there is no critically-bad state in V, based on the construction of the TRG, there do not exist two transition sequences $\sigma^1, \sigma^2 \in L(G)$ such that $M_0[\sigma^1\rangle M_1 \in C_R$ and $M_0[\sigma^2\rangle M_2 \notin C_R$ with $\lambda(\sigma^1) = \lambda(\sigma^2)$ hold. Thus, by Definition 3, G is critically observable and we have $[\mathcal{C}(w) \subseteq C_R] \vee [\mathcal{C}(w) \subseteq R(G) \setminus C_R]$. By $C_{R_t} = C_R$ and $R_t(G_t) \subseteq R(G)$, we can conclude that $[\mathcal{C}(\delta_o, \tau) \subseteq C_{R_t}] \vee [\mathcal{C}(\delta_o, \tau) \subseteq R_t(G_t) \setminus C_{R_t}]$ holds. Hence, the proposition holds. □

Remark 2: Given a bounded LPN G and its RG $\mathcal{R}$, the number of states and transitions in the TRG of $\mathcal{R}$ is the same as that of the RG of the twin-plant net of G in [24], i.e., the two structures are isomorphic. Compared with the RG of the twin-plant net in [24], we could directly obtain the pairs of the transition-marking sequences from the paths of the TRG.

Remark 3: Based on the relations among critical observability [24], current-state opacity [39], and state detectability [40], some methods to verify current-state opacity and state detectability can be adapted to verify system critical observability after a finite-state automaton is obtained. In fact, most of the works in the literature deal with these kinds of problems in the framework of logic DESs. However, if we consider the additional time information in a system, none of the methods in [24, 39, 40] for logic DESs are applicable to verify critical observability for bounded labeled TPNs in general. In this sense, it is necessary to design new methods to verify critical observability for bounded labeled TPNs, which has never been considered in the literature.

According to Proposition 1, if a logic LPN system is not critically observable, its corresponding TPN version may be either critically or not critically observable.

IV. Critical Observability in Labeled TPNs

Given a time instant τ and a set of critical markings C_{R_t}, we have to verify if there exist two reachable markings consistent with an observation δ_o and τ such that one is in the critical set and the other is not in the critical set.

To this end, the following steps are performed: 1) Determine the set of critically-bad states by the TRG V of the corresponding logic LPN system G; 2) Construct the set of all pairs of TCSs ending in a pair of classes with respect to (wrt) a critically-bad state at τ; 3) Check whether each pair of observable transitions in the pair of TCSs can occur at the same time instant.

A. Construction of a TCS

This section presents an algorithm for the construction of a TCS of a labeled TPN by using its corresponding transition-marking sequence of the underlying logic LPN.

Algorithm 1: Construction of a TCS.

Input: A labeled TPN system $G_t = (PN, M_0, E, \lambda, Q)$ and a transition-marking sequence $M_{\beta_0} t_{\alpha_1} M_{\beta_1} \ldots t_{\alpha_h} M_{\beta_h}$

Output: A TCS corresponding to the transition-marking sequence

The root node C_0 is labeled with the initial marking M_{β_0} and a set of inequalities Θ_0 defined as follows: for all $t_i \in \mathcal{A}(M_{\beta_0})$, let $l_i^0 \leq \theta_i \leq u_i^0$, where $l_i^0 = l_i$ and $u_i^0 = u_i$.
for *each* t_{α_j} $(j = 1, 2, \ldots, h)$ **do**
  select the previously generated node C_{j-1};
  if $\max\{0, l_{\alpha_j}^{j-1}\} \leq \min_{q, t_q \in \mathcal{A}(M_{\beta_{j-1}})}\{u_q^{j-1}\}$, *where* $l_{\alpha_j}^{j-1}$ (u_q^{j-1}) *is the lower (upper) bound associated with* t_{α_j} (t_q) *at class* C_{j-1} **then**
    if $t_r \in \mathcal{A}(M_{\beta_{j-1}})$ *and* $M_{\beta_{j-1}} - Pre(\cdot, t_{\alpha_j}) \geq Pre(\cdot, t_r)$ **then**
      let $l_r^j = l_r^{j-1} - \Delta_{(j)}$ and $u_r^j = u_r^{j-1} - \Delta_{(j)}$;
    else
      let $l_r^j = l_r$ and $u_r^j = u_r$;
    add a new node C_j labeled with marking M_{β_j} and a set of inequalities Θ_j defined as follows: for all $t_r \in \mathcal{A}(M_{\beta_j})$, let $\max\{0, l_r^j\} \leq \theta_r \leq u_r^j$;
    add an edge from C_{j-1} to C_j labeled "$t_{\alpha_j}, \lambda(t_{\alpha_j}), \Delta_{(j)} \in [\max\{0, l_{\alpha_j}^{j-1}\}, \min_{q, t_q \in \mathcal{A}(M_{\beta_{j-1}})}\{u_q^{j-1}\}]$";
  else
    terminate.

Now, we discuss the main steps of Algorithm 1. Step 1 starts a TCS from the initial class that contains the initial marking M_0 (denoted as M_{β_0} in the transition-marking sequence, i.e., $M_{\beta_0} = M_0$) and a set of static intervals of transitions enabled at M_{β_0}. In Steps 2 and 3, each time we consider a newly fired transition t_{α_j} $(j = 1, 2, \ldots, h)$ in the transition-marking sequence and the previous generated node (class) C_{j-1} associated with $M_{\beta_{j-1}}$ in the TCS, respectively. Step 4 checks if the logically enabled transition t_{α_j} at $M_{\beta_{j-1}}$ can also fire with the time constraints, i.e., the enabled transition whose lower bound at C_{j-1} is no greater than the minimum of the upper bounds at the same class of the transitions in $\mathcal{A}(M_{\beta_{j-1}})$. Moreover, in Step 4, $\max\{0, l_{\alpha_j}^{j-1}\}$ is used to ensure the non-negativeness of the lower bound. If the condition in Step 4 is verified, then Step 5 checks whether the corresponding transition has already been enabled at the previous class. More precisely, each inequality in Θ_j depends on the fact that the corresponding transition has already been enabled at the previous class, or it has just been enabled. If the first case happens, as shown in Step 6, it is necessary to reduce the upper and lower bounds at the previous class by a variable $\Delta_{(j)}$ that is equal to the time elapsed when going from the previous class to the new one. Otherwise, in Step 8, the firing domain obviously coincides with the static interval of the transition. Step 9 adds the newly obtained class to the TCS. In addtion, Step 10 adds the edge from C_{j-1} to C_j, which contains the transition, its label, and the time information that must elapse from C_{j-1} to C_j. Finally, if the condition in Step 4 does not hold, the algorithm terminates since the logically enabled transition t_{α_j} cannot fire and its subsequent transitions in this sequence cannot fire accordingly.

B. Construction of Feasible TCSs

Now, let

$$\begin{aligned}
&\pi' = C'_0 \xrightarrow{t_{\alpha'_1},\ \lambda(t_{\alpha'_1}),\ \Delta'_{(1)} \in [\max\{0, l^0_{\alpha'_1}\},\ \min_{q:t_q \in \mathcal{A}(M'_{\beta_0})}\{u^0_q\}]} \\
&\quad \ldots \\
&\quad C'_{h'-1} \qquad t_{\alpha'_{h'}},\ \lambda(t_{\alpha'_{h'}}), \\
&\qquad\qquad \xrightarrow{\Delta'_{(h')} \in [\max\{0, l^{h'-1}_{\alpha'_{h'}}\},\ \min_{q:t_q \in \mathcal{A}(M'_{\beta_{h'-1}})}\{u^{h'-1}_q\}]} \\
&\quad C'_{h'}
\end{aligned} \tag{1}$$

and

$$\begin{aligned}
&\pi = C_0 \xrightarrow{t_{\alpha_1},\ \lambda(t_{\alpha_1}),\ \Delta_{(1)} \in [\max\{0, l^0_{\alpha_1}\},\ \min_{q:t_q \in \mathcal{A}(M_{\beta_0})}\{u^0_q\}]} \\
&\quad \ldots \\
&\quad C_{h-1} \qquad t_{\alpha_h},\ \lambda(t_{\alpha_h}), \\
&\qquad\qquad \xrightarrow{\Delta_{(h)} \in [\max\{0, l^{h-1}_{\alpha_h}\},\ \min_{q:t_q \in \mathcal{A}(M_{\beta_{h-1}})}\{u^{h-1}_q\}]} \\
&\quad C_h
\end{aligned} \tag{2}$$

be two TCSs constructed by Algorithm 1 for the transition-marking sequences $M'_{\beta_0} t_{\alpha'_1} M'_{\beta_1} \ldots t_{\alpha'_{h'}} M'_{\beta_{h'}}$ and $M_{\beta_0} t_{\alpha_1} M_{\beta_1} \ldots t_{\alpha_h} M_{\beta_h}$, respectively.

Now, the following definition characterizes the feasibility of a TCS and a pair of TCSs at a given time instant.

Definition 7: Given a TCS π in (2) and a time instant $\tau \in \mathbb{R}_{\geq 0}$, π is said to be feasible at τ if the TCS can reach and remain in the last visited class at τ. A pair of TCSs (π', π) is feasible at τ if both TCSs π' and π are feasible at τ.

The following definitions are necessary to characterize the paths of the TRG V.

Definition 8: Given a critically-bad state $M^V_{j_k}$, an elementary bad path wrt $M^V_{j_k}$ in the TRG V is a path $\zeta_j = M^V_{j_1}(\eta'_{j_1}, \eta_{j_1}) M^V_{j_2}(\eta'_{j_2}, \eta_{j_2}) \ldots M^V_{j_{k-1}}(\eta'_{j_{k-1}}, \eta_{j_{k-1}}) M^V_{j_k}$ that starts from the initial state M^V_0 (denoted as $M^V_{j_1}$ in ζ_j, i.e., $M^V_{j_1} = M^V_0$) and ends in the critically-bad state $M^V_{j_k}$ but without passing twice through the same state. The set of elementary bad paths in V wrt $M^V_{j_k}$ is denoted by $\Gamma_b(M^V_{j_k})$.

Given an elementary bad path ζ_j wrt $M^V_{j_k}$, the corresponding event sequence is $s = (\eta'_{j_1}, \eta_{j_1}) \ldots (\eta'_{j_{k-1}}, \eta_{j_{k-1}})$, and ζ_j can be also written as $\zeta_j = M^V_{j_1} \xrightarrow{s} M^V_{j_k}$. Moreover, for a path ζ_j, let $\zeta_j(L)$ and $\zeta_j(R)$ denote its left and right components, respectively. By removing each symbol ε and its succeeding marking in $\zeta_j(L)$ and $\zeta_j(R)$, we can obtain the left and right transition-marking sequences of ζ_j, respectively.

Definition 9: Let ζ_1 and ζ_2 be two elementary bad paths wrt $M^V_{j_k}$. ζ_2 is said to be structurally identical to ζ_1 if the left and right transition-marking sequences of ζ_2 are the same as the left and right transition-marking sequences of ζ_1, respectively.

Definition 10: An elementary cycle in the TRG V is a path $\mu_j = M^V_{j_1}(\eta'_{j_1}, \eta_{j_1}) M^V_{j_2}(\eta'_{j_2}, \eta_{j_2}) \ldots M^V_{j_{k-1}}(\eta'_{j_{k-1}}, \eta_{j_{k-1}}) M^V_{j_1}$ that starts from and ends in the same state but without passing twice through any other state. The set of elementary cycles in V is denoted by Γ_c.

Given an elementary cycle μ_j, any state $M^V_{j_i}$ in μ_j can be seen as the first state of μ_j. Let s_i be an event sequence in μ_j such that $f^V(M^V_{j_i}, s_i) = M^V_{j_i}$, μ_j can be also written as $\mu_j = M^V_{j_i} \xrightarrow{s_i} M^V_{j_i}$ for $i = 1, \ldots, k-1$. For a cycle μ_j, let $\mu_{j_i}(L)$ and $\mu_{j_i}(R)$ denote its left and right components starting from any state $M^V_{j_i}$ of μ_j, respectively. By removing each symbol ε and its succeeding marking in $\mu_{j_i}(L)$ and $\mu_{j_i}(R)$, we can obtain the left and right transition-marking sequences of μ_j starting from $M^V_{j_i}$, respectively.

Definition 11: Let μ_1 and μ_2 be two elementary cycles. μ_2 is said to be structurally identical to μ_1 if the left and right transition-marking sequences starting from any state of μ_2 are the same as the left and right transition-marking sequences starting from the same state of μ_1, respectively.

Consider two paths ζ_j and μ_j, $\zeta_j = M^V_{j_1} \xrightarrow{s} M^V_{j_k}$ starts from the initial state M^V_0 (denoted as $M^V_{j_1}$ in ζ_j) and ends in a critically-bad state $M^V_{j_k}$, and μ_j is an elementary cycle. The insertion of μ_j to ζ_j is denoted as $\zeta_j \oplus \mu_j$ that is defined as follows: 1) $\zeta_j \oplus \mu_j = \{\zeta | \zeta = M^V_{j_1} \xrightarrow{s'} M^V_{j_i} \xrightarrow{s_i} M^V_{j_i} \xrightarrow{s''} M^V_{j_k}$ with $s = s's''$, $s', s'' \in (\Sigma^V)^*$ and $M^V_{j_i}$ is the first common state of ζ_j and $\mu_j\}$, if μ_j and ζ_j have the common state; 2) $\zeta_j \oplus \mu_j = \{\zeta | \zeta = M^V_{j_1} \xrightarrow{s} M^V_{j_k}\}$, if μ_j and ζ_j have no common state.

Example 2: Consider the LPN in Fig. 1 and a critically-bad state (M_0, M_2). There are three elementary bad paths ζ_1, ζ_2, and ζ_3 wrt (M_0, M_2), where the left transition-marking

sequence of ζ_1, ζ_2, and ζ_3 is $M_0t_1M_1t_2M_2t_3M_0$, and the right transition-marking sequence of ζ_1, ζ_2, and ζ_3 is $M_0t_1M_1t_2M_2$. Based on Definition 9, ζ_2 and ζ_3 are structurally identical to ζ_1. In addition, for the LPN in Fig. 1, there are six elementary cycles μ_1 to μ_6, where the left transition-marking sequence of μ_1 to μ_6 starting from (M_0, M_0) is $M_0t_1M_1t_2M_2t_3M_0$, and the right transition-marking sequence of μ_1 to μ_6 starting from (M_0, M_0) is $M_0t_1M_1t_2M_2t_3M_0$. Based on Definition 11, μ_2 to μ_6 are structurally identical to μ_1. The insertion of $\mu_1 = (M_0, M_0)(t_1, t_1)(M_1, M_1)(t_2, \varepsilon)(M_2, M_1)(t_3, \varepsilon)(M_0, M_1)(\varepsilon, t_2)(M_0, M_2)(\varepsilon, t_3)(M_0, M_0)$ to $\zeta_1 = (M_0, M_0)(t_1, t_1)(M_1, M_1)(t_2, \varepsilon)(M_2, M_1)(t_3, \varepsilon)(M_0, M_1)(\varepsilon, t_2)(M_0, M_2)$ is $\zeta_1 \oplus \mu_1 = \{(M_0, M_0)(t_1, t_1)(M_1, M_1)(t_2, \varepsilon)(M_2, M_1)(t_3, \varepsilon)(M_0, M_1)(\varepsilon, t_2)(M_0, M_2)(\varepsilon, t_3)(M_0, M_0)(t_1, t_1)(M_1, M_1)(t_2, \varepsilon)(M_2, M_1)(t_3, \varepsilon)(M_0, M_1)(\varepsilon, t_2)(M_0, M_2)\}$. □

Considering a TCS π and a time instant $\bar{\tau}$, let $out(C_q)$ denote the set of transitions associated with the edges that exit from C_q. In order to verify if a TCS π is feasible at $\bar{\tau}$, the following set of constraints has to be verified:

$$\sum_{j=1}^{h} \Delta_{(j)} \leq \bar{\tau}, \tag{3.1}$$

$$\bar{\tau} - \sum_{j=1}^{h} \Delta_{(j)} < \min_{r:t_r \in out(C_h)}\{u_r^h\}, \tag{3.2}$$

$$\Delta_{(j)} \geq \max\{0, l_{\alpha_j}^{j-1}\}, \quad j = 1, \ldots, h, \tag{3.3}$$

$$\Delta_{(j)} \leq \min_{q:q \in \mathcal{A}(M_{\beta_{j-1}})}\{u_q^{j-1}\}, \quad j = 1, \ldots, h. \tag{3.4}$$

Constraint (3.1) means that the total time elapsed to reach class C_h cannot be greater than a time instant $\bar{\tau}$. Constraint (3.2) imposes that C_h can be the last visited class in the TCS. Indeed, once the node C_h is reached, if the difference between $\bar{\tau}$ and the time spent to reach the class C_h is greater than or equal to the minimum of the upper bounds of the output transitions of the class C_h, then a transition in $out(C_h)$ has fired. Constraints (3.3) and (3.4) impose the limitations on the time intervals for all the edges in the TCS. Note that constraints (3.2)–(3.4) can be readily linearized by the technique in [1].

In order to check critical observability of a labeled TPN at a given time instant τ, for each critically-bad state M^V, it is necessary to consider all the pairs of TCSs ending in a pair of classes wrt M^V, which are feasible at τ. Moreover, to find such pairs of TCSs, we need to consider all the elementary bad paths ζ_j wrt M^V that are not structurally identical to each other including the possible insertion of cycles μ_j to obtain $\zeta_j \oplus \mu_j$ of V and the associated pairs of TCSs. Consequently, constraints (3.1)–(3.4) should be applied to an infinite number of TCSs by replacing $\bar{\tau}$ with τ. With the aim of obtaining an efficient methodology that limits the number of TCSs, the following two linear programming problems (LPPs) are formulated:

LPP 1: min $\bar{\tau}$ subject to constraints (3.1)–(3.4);

LPP 2: max $\bar{\tau}$ subject to constraints (3.1)–(3.4).

Given a TCS π' (π), we denote by $\bar{\tau}'_{max}$ and $\bar{\tau}'_{min}$ ($\bar{\tau}_{max}$ and $\bar{\tau}_{min}$) the optimal values of the objective functions of LPPs 1 and 2, respectively, for the TCS π' (π). Now, the following result is proved.

Proposition 2: Given a time instant $\tau \in \mathbb{R}_{\geq 0}$ and a critically-bad state $M^V = (M', M)$, a pair of TCSs (π', π) that ends in a pair of classes wrt M^V is feasible at τ, *iff* $\max\{\bar{\tau}'_{min}, \bar{\tau}_{min}\} \leq \tau \leq \min\{\bar{\tau}'_{max}, \bar{\tau}_{max}\}$ holds.

Proof: (*If*) Since LPPs 1 and 2 have the optimal solutions with $\max\{\bar{\tau}'_{min}, \bar{\tau}_{min}\} \leq \tau \leq \min\{\bar{\tau}'_{max}, \bar{\tau}_{max}\}$, the value of τ satisfies constraints (3.1)–(3.4) of LPPs 1 and 2 for the two TCSs π' and π. Thus, the two TCSs can reach and remain in the last visited classes $C'_{h'}$ and C_h at τ, respectively. Based on Definition 7, the pair of TCSs is feasible at τ.

(*Only if*) By contradiction, suppose that either $\tau < \max\{\bar{\tau}'_{min}, \bar{\tau}_{min}\}$ or $\tau > \min\{\bar{\tau}'_{max}, \bar{\tau}_{max}\}$ holds. In particular, if $\tau < \max\{\bar{\tau}'_{min}, \bar{\tau}_{min}\}$, then the pair of TCSs cannot remain in the last classes pair at τ. If $\tau > \min\{\bar{\tau}'_{max}, \bar{\tau}_{max}\}$, then the pair of TCSs cannot reach the last classes pair at τ. Based on Definition 7, both the cases contradict the hypothesis. □

Algorithm 2: Construction of the set of all pairs of TCSs ending in a pair of classes wrt a critically-bad state, which are feasible at τ.

Input: A critically-bad state M^V, a time instant $\tau \in \mathbb{R}_{\geq 0}$, and a TRG V

Output: The set $\Omega(M^V, \tau)$ of all the TCSs pairs ending in a pair of classes wrt M^V, which are feasible at τ

```
Compute the set Γ_b(M^V) wrt M^V of V.
Compute the set Γ_c of V.
for each ζ_j ∈ Γ_b(M^V) do
    compute the set of paths that are structurally identical to ζ_j, which is denoted by Γ_b^{ζ_j}(M^V);
    Γ_b(M^V) := Γ_b(M^V) \ Γ_b^{ζ_j}(M^V);
for each μ_j ∈ Γ_c do
    compute the set of the cycles that are structurally identical to μ_j, which is denoted by Γ_c^{μ_j};
    Γ_c := Γ_c \ Γ_c^{μ_j};
Ω(M^V, τ) := ∅, W := Γ_b(M^V);
while W ≠ ∅ do
    select a path ζ_j ∈ W;
    construct a pair of TCSs (π', π) for the left and right transition-marking sequences of ζ_j by Algorithm 1;
    if max{τ̄'_min, τ̄_min} ≤ τ ≤ min{τ̄'_max, τ̄_max} then
        Ω(M^V, τ) := Ω(M^V, τ) ∪ {(π', π)};
        for each μ_j ∈ Γ_c do
            W := W ∪ (ζ_j ⊕ μ_j);
    else if τ > min{τ̄'_max, τ̄_max} then
        for each μ_j ∈ Γ_c do
            W := W ∪ (ζ_j ⊕ μ_j);
    W := W \ {ζ_j}.
```

Based on the aforementioned results, we propose Algorithm 2 that is used to generate all the pairs of TCSs ending in a pair of classes wrt a critically-bad state M^V, which are feasible at a given time instant τ.

In particular, Step 1 computes the set of elementary bad paths $\Gamma_b(M^V)$ in V by using depth-first search. Step 2 computes the set of elementary cycles Γ_c in V by using the method in [16]. For each elementary bad path ζ_j in $\Gamma_b(M^V)$ and each elementary cycle μ_j in Γ_c, we remove all the elementary bad paths that are structurally identical to ζ_j and all the elementary cycles that are structurally identical to μ_j (Steps 3 to 8), since the pairs of TCSs related to these paths including the possible insertion of these cycles are the same.

The set of all pairs of required TCSs is initialized at the empty set, which is denoted as $\Omega(M^V,\tau)$; the set of unexplored paths ending in M^V is initialized at $\Gamma_b(M^V)$, which is denoted as W (Step 9). For each path ζ_j in W that has not been explored, we construct a pair of TCSs (π',π) for the left and right transition-marking sequences of ζ_j by Algorithm 1 (Steps 10 to 12). If (π',π) is feasible at τ, it is added to $\Omega(M^V,\tau)$, and each time we insert one elementary cycle $\mu_j \in \Gamma_c$ to ζ_j and add the set $(\zeta_j \oplus \mu_j)$ to the set W (Steps 13 to 16). Since the condition $\tau \geq \max\{\bar{\tau}'_{min}, \bar{\tau}_{min}\}$ in Step 13 is always verified if the sum of the lower bounds in the cycles of transitions is equal to zero, by Assumption A2 the set $(\zeta_j \oplus \mu_j)$ cannot be added infinite times. If the pair of TCSs (π',π) cannot reach the last classes pair wrt M^V at τ, each time we insert one elementary cycle $\mu_j \in \Gamma_c$ to ζ_j and add the set $(\zeta_j \oplus \mu_j)$ to W (Steps 17 to 20). By Assumption A2, the set $(\zeta_j \oplus \mu_j)$ cannot be added infinite times if the condition in Step 17 is verified. Step 20 removes the explored path ζ_j from W. Steps 10 to 20 execute iteratively until there is no unexplored path in W.

C. Verification of Critical Observability in Labeled TPNs

The following proposition proves the relationship between the critical observability of a labeled TPN system and the pair of TCSs.

Proposition 3: Consider a labeled TPN system $G_t = (PN, M_0, E, \lambda, Q)$, a set of critical markings C_{R_t} and a time instant $\tau \in \mathbb{R}_{\geq 0}$. The system G_t is critically observable at τ *iff* no pair of TCSs (π',π) in $\Omega(M^V,\tau)$ for any critically-bad state M^V satisfies the following set of constraints:

$$\begin{cases} \sum_{j=1}^{h'} \Delta'_{(j)} \leq \tau, \\ \tau - \sum_{j=1}^{h'} \Delta'_{(j)} < \min_{r:t_r \in out(C'_{h'})}\{u_r^{h'}\}, \\ \sum_{j=1}^{h} \Delta_{(j)} \leq \tau, \\ \tau - \sum_{j=1}^{h} \Delta_{(j)} < \min_{r:t_r \in out(C_h)}\{u_r^{h}\}, \\ \Delta'_{(j)} \geq \max\{0, l_{\alpha_j}^{j-1}\}, & j = 1,\ldots,h', \\ \Delta'_{(j)} \leq \min_{q:q \in \mathcal{A}(M'_{\beta_{j-1}})}\{u_q^{j-1}\}, & j = 1,\ldots,h', \\ \Delta_{(j)} \geq \max\{0, l_{\alpha_j}^{j-1}\}, & j = 1,\ldots,h, \\ \Delta_{(j)} \leq \min_{q:q \in \mathcal{A}(M_{\beta_{j-1}})}\{u_q^{j-1}\}, & j = 1,\ldots,h, \\ \sum_{j=1}^{p'} \Delta'_{(j)} = \sum_{j=1}^{p} \Delta_{(j)}, & \forall t_{\alpha'_{p'}}, t_{\alpha_p} \text{ with the same label and same observable prefix.} \end{cases} \quad (4)$$

Proof: *(If)* We prove the contrapositive. Based on the proof of [2, Proposition 7], if the set of constraints (4) admits a feasible solution for at least one pair of TCSs (π',π) in $\Omega(M^V,\tau)$ for a critically-bad state M^V at τ, then π and π' can generate the same observations in the same time instants. According to Definition 6, G_t is not critically observable at τ.

(Only if) We prove the contrapositive. From Definition 6, if G_t is not critically observable at τ, then there exists a pair of TCSs (π',π) in $\Omega(M^V,\tau)$ for a critically-bad state M^V such that π' and π can generate the same observations in the same time instants. On the basis of the proof of [2, Proposition 7], if such a pair of TCSs exists, then the set of constraints (4) admits a feasible solution. □

Note that if none of the edges of a pair of TCSs (π',π) is labeled with any observable transition, then the last constraint that considers the transitions with the same label and same observable prefix in constraints (4) is omitted. According to the aforementioned results, we present Algorithm 3 to verify critical observability of the labeled TPN systems. In particular, Algorithm 3 consists of two parts. The first part includes Steps 1 to 5 that are the preliminary verification of the underlying logic LPN. The second part includes Steps 6 to 12 that address the time constraints associated with transitions.

Algorithm 3: Verification of critical observability for a labeled TPN.

Input: A labeled TPN system $G_t = (PN, M_0, E, \lambda, Q)$, a set of critical markings C_{R_t} and a time instant $\tau \in \mathbb{R}_{\geq 0}$

Output: $Crit \in \{0,1\}$: $Crit = 1$ ($Crit = 0$) means that the system is (is not) critically observable at τ

1 Remove the time information of G_t to obtain the logic version of G_t, denoted as $G = (PN, M_0, E, \lambda)$.
2 Build its RG $\mathcal{R} = (\mathcal{M}, \Sigma, f, M_0)$ and the corresponding TRG $V = (\mathcal{M}^V, \Sigma^V, f^V, M_0^V)$.
3 Find the set of critically-bad states $\mathcal{M}_C^V$ in V.
4 **if** $\mathcal{M}_C^V = \emptyset$ **then**
5 return $Crit = 1$;
6 **else**
7 **for** *each critically-bad state* $M^V \in \mathcal{M}_C^V$ **do**
8 construct the set $\Omega(M^V,\tau)$ of all the pairs of TCSs (π',π) ending in a pair of classes wrt M^V, which are feasible at τ by Algorithm 2;
9 **for** *each pair of TCSs* $(\pi',\pi) \in \Omega(M^V,\tau)$ **do**
10 **if** *the set of constraints* (4) *has a solution* **then**
11 return $Crit = 0$;
12 return $Crit = 1$.

Proposition 4: Given a labeled TPN system $G_t = (PN, M_0, E, \lambda, Q)$, a set of critical markings C_{R_t} and a time instant $\tau \in \mathbb{R}_{\geq 0}$, Algorithm 3 allows checking whether the system is/is not critically observable at τ.

Proof: At Step 1, we obtain the logic LPN system G by removing all the time information of G_t. At Step 2, we build the RG of G and the corresponding TRG V. Step 3 checks critical observability for the logic version of the labeled TPN system.

At Step 4, based on Proposition 1, if the logic system is critically observable, then G_t is also critically observable at

any given time instant, i.e., the algorithm returns $Crit = 1$. If the logic system is not critically observable, it is necessary to check whether there exists a pair of transition-marking sequences that violates the critical observability can occur with the time constraints. To this end, for each critically-bad state M^V, Step 8 constructs the set of all the pairs of TCSs that are required to check critical observability at τ. Finally, for each pair of TCSs in $\Omega(M^V, \tau)$, according to Proposition 3, if constraints (4) admit a feasible solution for one pair of the TCSs, then G_t is not critically observable at the time instant τ, and the algorithm returns $Crit = 0$ (Steps 9 to 11). On the contrary, if there does not exist such a pair of TCSs, G_t is critically observable at τ, and the algorithm returns $Crit = 1$ (Step 12). □

Remark 4: We point out that the critical observability of a labeled TPN system is strongly related to the given time instant. Generally, if a system is critically observable at a given time instant τ, we cannot deduce critical observability for the system at a time instant $\tau' > \tau$. Indeed, at the time instant τ', some new pairs of TCSs may occur and they can violate critical observability. Hence, it is necessary to apply Algorithm 3 again to check critical observability for the labeled TPN system at τ'.

Remark 5: If there exist some cycles in the concurrent composition of an RG, for a critically-bad state in a cycle, there exist an infinite number of sequences that can lead to this state. Thus, if we want to check whether a system is always critically observable at any time instant, we need to consider an infinite number of TCSs, which is infeasible in practice. Moreover, in a realistic timed system, the occurrence of some dangerous states or situations is always associated with some particular time instants. In this sense, the notion of critical observability of a labeled TPN is associated with a given time instant instead of any time instant in this paper.

D. Computational Complexity

In this subsection, we analyze the complexity of Algorithm 3 (including the construction of a TRG and the application of Algorithms 1 and 2) that is the main result to check critical observability of the labeled TPN systems.

Given a labeled TPN G_t, let G be its logic version and $\mathcal{R} = (\mathcal{M}, \Sigma, f, M_0)$ be the RG of G. As for the computation of the TRG V of $\mathcal{R}$, it has at most $|\mathcal{M}|^2$ states and at most $|\mathcal{M}|^2 \cdot (2n_u + \sum_{\gamma_i \in E} n_i^2)$ transitions, where n_u is the number of unobservable transitions and n_i is the number of transitions labeled with $\gamma_i \in E$ of G. Hence, the complexity of constructing the TRG is $O(|\mathcal{M}|^2 \cdot (n_u + \sum_{\gamma_i \in E} n_i^2))$. However, for a bounded LPN system, the size of the RG increases exponentially both with the number of tokens of the initial marking and with the number of places [31]. Thus, the computational complexity of verifying critical observability for bounded LPNs in Step 2 of Algorithm 3 is exponential wrt the dimension of the net systems.

As for Algorithm 1, for each transition-marking sequence with l transitions, the complexity of Algorithm 1 is $O((l+1) \cdot (n+1) + l)$, where n is the number of transitions in G.

As for Algorithm 2, the complexity of Steps 1 and 2 in Algorithm 2 is $O(|\mathcal{M}|^2 \cdot (n_u + \sum_{\gamma_i \in E} n_i^2) \cdot (d+c+1))$, where d and c are the number of elementary bad paths wrt a critically-bad state and the number of elementary cycles in the TRG V, respectively. The complexity of Steps 3 to 8 in Algorithm 2 is $O(d + c)$. The complexity of Steps 10 to 20 in Algorithm 2 firstly depends on the number of explored paths in V and the complexity of Algorithm 1. Second, it also depends on the complexity of solving LPPs 1 and 2 for each pair of TCSs. Let l_{max} be the number of events in the longest explored path in V. The number of explored paths is at most $\sum_{j=1}^{l_{max}} (2n_u + \sum_{\gamma_i \in E} n_i^2)^j$. Hence, Algorithm 2 needs to apply Algorithm 1, LPP 1, and LPP 2 for $2 \cdot \sum_{j=1}^{l_{max}} (2n_u + \sum_{\gamma_i \in E} n_i^2)^j$ times in the worst case. Moreover, the computational complexity of solving an LPP is polynomial and depends on its number of constraints and variables. The number of constraints and the number of variables in LPP 1 or LPP 2 are $2+n+(2+n) \cdot l_{max}$ and $1 + l_{max}$ in the worst case, respectively.

Based on the complexity of the computation of a TRG and the complexity of Algorithms 1 and 2, we show the complexity of Algorithm 3. In particular, Step 1 in Algorithm 3 requires no time by simply ignoring the time information of each transition. The complexity of Step 2 in Algorithm 3 is equal to the complexity of constructing the TRG. The complexity of Step 8 in Algorithm 3 is equal to the complexity of Algorithm 2, and this step needs to be applied for at most $|\mathcal{M}_C^V|$ times that is less than $|\mathcal{M}|^2$.

For a critically-bad state M^V and a time instant τ, let l'_{max} be the number of events in the longest path in V that corresponds to the longest pair of TCSs in $\Omega(M^V, \tau)$. Step 10 in Algorithm 3 needs to be executed at most $\sum_{j=1}^{l'_{max}} (2n_u + \sum_{\gamma_i \in E} n_i^2)^j$ times and each time it solves an LPP formulated by an arbitrary linear objective function subject to the linearization of (4). The number of constraints and the number of variables in the linearization of (4) are $2 + 2n + (5 + 2n) \cdot l'_{max}$ and $2l'_{max}$ in the worst case, respectively. In summary, we emphasize that the complexity of the proposed method mainly depends on the construction of the TRG, Algorithm 2, and Step 10 in Algorithm 3, and the most important parameters are the size of a net system, the initial marking, and the cardinality of $\Omega(M^V, \tau)$.

V. Examples

Let us consider the labeled TPN system that has the same net structure, the initial marking, and the labeling function as the LPN system in Fig. 1, where the time constraints for the transitions are $Q(t_1) = (0, 1)$, $Q(t_2) = (2, 3)$, and $Q(t_3) = (1, 3)$. Let $C_R = \{M \in \mathbb{N}^3 | -M(p_1) \leq -1 \vee -M(p_2) \leq -1\}$ and $\tau = 3$. By Algorithm 3, the LPN system is not critically observable. There are two critically-bad states (M_1, M_2) and (M_0, M_2). For the critically-bad state (M_1, M_2), there exists one pair of TCSs (π'_0, π_0) in the set $\Omega((M_1, M_2), 3)$, which corresponds to $(M_0, M_0)(t_1, t_1)(M_1, M_1)(\varepsilon, t_2)(M_1, M_2)$, and its graphic representation is shown in Fig. 2. Then, constraints (4) for this pair of TCSs (π'_0, π_0) are the following:

$$\begin{cases} \Delta'_{(1)} \le 3 \\ 3 - \Delta'_{(1)} < 3 \\ \Delta_{(1)} + \Delta_{(2)} \le 3 \\ 3 - (\Delta_{(1)} + \Delta_{(2)}) < 3 \\ 0 \le \Delta'_{(1)} \le 1 \\ 0 \le \Delta_{(1)} \le 1 \\ 2 \le \Delta_{(2)} \le 3 \\ \Delta'_{(1)} = \Delta_{(1)}. \end{cases} \quad (5)$$

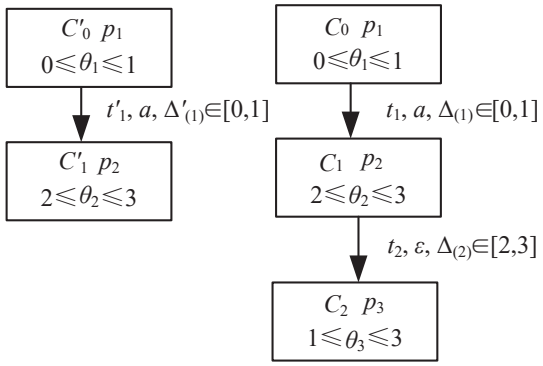

Fig. 2: The pair of TCSs (π'_0, π_0).

We verify that the set of constraints is feasible and a solution is $\Delta'_{(1)} = 0.1$, $\Delta_{(1)} = 0.1$, and $\Delta_{(2)} = 2$. Then, on the basis of Algorithm 3, we can conclude that the labeled TPN system is not critically observable at $\tau = 3$.

In order to show that the property of critical observability is strongly related to the time constraint of each transition, we change $Q(t_1)$, $Q(t_2)$, and $Q(t_3)$ to $(0.5, 3)$, $(2, 2)$, and $(0.6, 1)$, respectively. Let us consider the same set of critical markings and the same time instant. For the critically-bad state (M_1, M_2), there exists one pair of TCSs (π'_1, π_1) in $\Omega((M_1, M_2), 3)$ corresponding to $(M_0, M_0)(t_1, t_1)(M_1, M_1)(\varepsilon, t_2)(M_1, M_2)$, and its graphic representation is shown in Fig. 3. Then, we solve constraints (4) for the pair of TCSs (π'_1, π_1) as follows:

$$\begin{cases} \Delta'_{(1)} \le 3 \\ 3 - \Delta'_{(1)} < 2 \\ \Delta_{(1)} + \Delta_{(2)} \le 3 \\ 3 - (\Delta_{(1)} + \Delta_{(2)}) < 1 \\ 0.5 \le \Delta'_{(1)} \le 3 \\ 0.5 \le \Delta_{(1)} \le 3 \\ \Delta_{(2)} = 2 \\ \Delta'_{(1)} = \Delta_{(1)}. \end{cases} \quad (6)$$

However, the set of constraints (6) does not admit any feasible solution. In addition, for the critically-bad state (M_0, M_2), we have $\Omega((M_0, M_2), 3) = \emptyset$. Consequently, by Algorithm 3, we conclude that the labeled TPN system is critically observable at $\tau = 3$.

Now, we consider another example in Fig. 4 to further show the advantage of the proposed method and the application in cyber-security area. The labeled TPN in Fig. 4 is obtained by slightly modifying the one in [41]. In particular, the plant consists of four places p_1–p_4 and four transitions t_1–t_4 with $\lambda(t_1) = a$, $\lambda(t_2) = b$, $\lambda(t_3) = \varepsilon$, and $\lambda(t_4) = c$. Transitions t_1^- and t_2^+ represent two kinds of sensor attack, i.e., sensor erasure

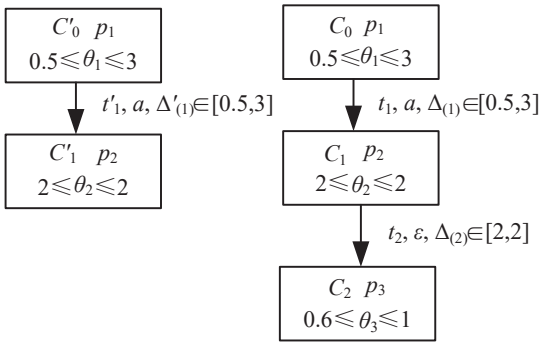

Fig. 3: The pair of TCSs (π'_1, π_1).

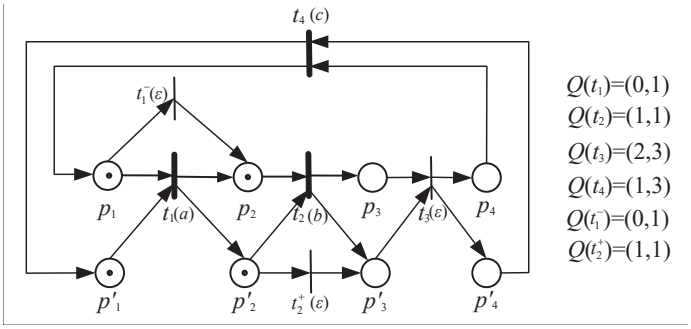

Fig. 4: A labeled TPN under attack.

and sensor insertion, respectively, which are unobservable. Places p'_1–p'_4 are the places of an operator monitor. Moreover, for the labeled TPN G_t in Fig. 4, the projection of marking $M \in R_t(G_t)$ on $P = \{p_1, p_2, p_3, p_4\}$ indicates the real state of the system under attack, while the projection of marking $M \in R_t(G_t)$ on $P' = \{p'_1, p'_2, p'_3, p'_4\}$ indicates the state that is observed by the operator. In this case, a reachable marking is said to be critical if the projection of the marking on P is different from the projection of the marking on P', i.e., a real state of the system is different from what is observed from the operator.

By using Algorithm 3, the underlying net system is not critically observable. There are 72 states in its TRG, and 15 of them are critically-bad states. Given a time instant $\tau = 2$, for any critically-bad state M^V, we can obtain $\Omega(M^V, \tau) = \emptyset$ by Algorithm 2. Thus, Algorithm 3 returns $Crit = 1$, which means that the time net system of Fig. 4 is critically observable at $\tau = 2$. Actually, at the time instant $\tau = 2$, for any observation $\delta_o \in \mathcal{L}(G_t, \tau)$, the observer can determine whether the set of markings consistent with δ_o are critical.

Furthermore, in order to check critical observability of the labeled TPN at time instant τ, by exploiting the technique in [1, 2, 13], it is necessary to first build and search the full state space, and then construct the corresponding pairs of TCSs to check whether the marking pairs that violate critical observability can survive with the time constraints. However, the authors of [1, 2, 13] do not study how to find all such pairs of TCSs that have the same observation. That is to say, the methods in [1, 2, 13] cannot be used to verify critical observability for a bounded labeled TPN in general.

VI. Conclusion

This paper introduces the notion of critical observability in the framework of labeled TPN models, and presents a formal method to verify this property for the bounded labeled TPNs. The proposed approach is based on two main procedures: i) the first procedure is a preliminary verification analysis of the underlying logic net; ii) the second procedure constructs some TCSs and solves some LPPs. We prove that the presented technique allows verifying critical observability of these timed DESs, and we provide a discussion about the computational complexity of the methodology. Two examples are presented to demonstrate the method.

Our future work is to investigate the enforcement of critical observability in the framework of labeled TPNs, i.e., a labeled TPN system that is not critically observable at a given time instant has forced to be critically observable by re-designing the time intervals for some transitions. In addition, we also plan to investigate the methods to verify critical observability for unbounded labeled TPNs in the future.

References

[1] F. Basile, M. P. Cabasino, and C. Seatzu, "State estimation and fault diagnosis of time labeled Petri net systems with unobservable transitions," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 997–1009, Apr. 2015.

[2] F. Basile, M. P. Cabasino, and C. Seatzu, "Diagnosability analysis of labeled time Petri net systems," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1384–1396, Mar. 2017.

[3] F. Basile, P. Chiacchio, and J. Coppola, "Identification of time Petri net models," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 47, no. 9, pp. 2586–2600, Sep. 2017.

[4] B. Bérard, F. Cassez, S. Haddad, D. Lime, and O. H. Roux, "Comparison of different semantics for time Petri nets," in *Proc. 3rd Int. Conf. Autom. Technol. Verification Anal.*, Taipei, Taiwan, pp. 293–307, 2005.

[5] B. Berthomieu and M. Diaz, "Modeling and verification of time dependent systems using time Petri nets," *IEEE Trans. Software Eng.*, vol. 17, no. 5, pp. 259–273, Mar. 1991.

[6] P. Bonhomme, "Marking estimation of P-time Petri nets with unobservable transitions," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 45, no. 3, pp. 508–518, Mar. 2015.

[7] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, New York, NY, USA: Springer, 2008.

[8] X. Y. Cong, M. P. Fanti, and A. M. Mangini, and Z. W. Li, "Critical observability of discrete-event systems in a Petri net framework," *IEEE Trans. Syst., Man, Cybern.: Syst.*, DOI: 10.1109/TSMC.2021.3056693, 2021.

[9] P. Declerck and P. Bonhomme, "State estimation of timed labeled Petri nets with unobservable transitions," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 103–110, Jan. 2014.

[10] J. Fearnley and M. Jurdzínski, "Reachability in two-clock timed automata is PSPACE-complete," *Inform. and Computation*, vol. 243, pp. 26–36, Aug. 2015.

[11] M. Ghazel, A. Toguyeni, and P. Yim, "State observer for DES under partial observation with time Petri nets," *Discrete Event Dyn. Syst.*, vol. 19, no. 2, pp. 137–165, Jun. 2009.

[12] C. N. Hadjicostis. *Estimation and Inference in Discrete Event Systems*, Cham, Switzerland: Springer, 2020.

[13] Z. He, Z. W. Li, A. Giua, F. Basile, and C. Seatzu, "Some remarks on "State estimation and fault diagnosis of time labeled Petri net systems with unobservable transitions"," *IEEE Trans. Autom. Control*, vol. 64, no. 12, pp. 5253–5259, Dec. 2019.

[14] Z. He, Z. Y. Ma, Z. W. Li, and A. Giua, "Parametric transformation of timed weighted marked graphs: applications in optimal resource allocation," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 1, pp. 179–188, Jan. 2021.

[15] Y. S. Huang, Y. S. Weng, and M. C. Zhou, "Design of traffic safety control systems for emergency vehicle preemption using timed Petri nets," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 2113–2120, Aug. 2015.

[16] D. B. Johnson, "Finding all the elementary circuits of a directed graph," *SIAM J. Computing*, vol. 4 no. 1, pp.77–84, 1975.

[17] A. W. Lai, S. Lahaye, and J. Komenda, "Observer construction for polynomially ambiguous max-plus automata," *IEEE Trans. Autom. Control*, DOI: 10.1109/TAC.2021.3069899, 2021.

[18] D. Lefebvre and C. Daoui, "Control design for bounded partially controlled TPNs using timed extended reachability graphs and MDP," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 50, no. 6, pp. 2273–2283, Jun. 2020.

[19] L. Li, F. Basile, and Z. W. Li, "An approach to improve permissiveness of supervisors for GMECs in time Petri net systems," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 237–251, Jan. 2020.

[20] L. Li, F. Basile, and Z. W. Li, "Closed-loop deadlock-free supervision for GMECs in time Petri net systems," *IEEE Trans. Autom. Control*, vol. 66, no. 11, pp. 5326–5341, Nov. 2021.

[21] Z. Y. Ma, Z. W. Li, and A. Giua, "Marking estimation in a class of time labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 493–506, Feb. 2020.

[22] R. Entezari-Maleki, S. E. Etesami, N. Ghorbani, A. A. Niaki, L. Sousa, and A. Movaghar, "Modeling and evaluation of service composition in commercial multiclouds using timed colored Petri nets," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 50, no. 3, pp. 947–961, Mar. 2020.

[23] X. Y. Mao, J. Cardoso, and R. Valette, "A new graph of classes for the preservation of quantitative temporal constraints," in *Proc. 3rd Int. Conf. Autom. Technol. Verification Anal.*, Taipei, Taiwan, pp. 278–292, 2005.

[24] T. Masopust, "Critical observability for automata and Petri nets," *IEEE Trans. Autom. Control*, vol. 65, no. 1, pp. 341–346, Jan. 2020.

[25] P. M. Merlin, A study of the recoverability of computing systems, Ph.D. dissertation, Univ. of Califonia, Irvine, 1974.

[26] C. K. Pang and C. V. Le, "Optimization of total energy consumption in flexible manufacturing systems using weighted P-timed Petri nets and dynamic programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1083–1096, Oct. 2014.

[27] Y. Pencolé and A. Subias, "Diagnosability of event patterns in safe labeled time Petri nets: A model-checking approach," *IEEE Trans. Autom. Sci. Eng.*, DOI: 10.1109/TASE.2020.3045565, 2020.

[28] C. Ramchandani, "Analaysis of asynchronous concurrent systems by timed Petri nets," Massachusetts Inst. of Technology, Cambridge, MA, USA, Tech. Rep., 1974.

[29] E. De Santis, M. D. Di Benedetto, S. Di Gennaro, A. D'Innocenzo, and G. Pola, "Critical observability of a class of hybrid systems and application to air traffic management," *Lecture Notes in Control and Inform. Sci.*, Springer-Verlag, vol. 337, pp. 141–170, 2006.

[30] E. De Santis and M. D. Di Benedetto, "Observability and diagnosability of finite state systems: A unifying framework," *Automatica*, vol. 81, pp. 115–122, Jul. 2017.

[31] C. Seatzu, M. Silva, and J. H. van Schuppen, Eds., *Control of Discrete-Event Systems. Automata and Petri Net Perspectives*, in ser. *Lecture Notes in Control and Inform. Sci.*, vol. 433. New York, NY, USA: Springer, 2013.

[32] Y. Y. Yan, H. Deng, and Z. Q. Chen, "A new look at the critical observability of finite state machines from an algebraic viewpoint," *Asian J. Control*, DOI: 10.1002/ASJC.2705, 2021.

[33] F. J. Yang, N. Q. Wu, Y. Qiao, M. C. Zhou, R. Su, and T. Qu, "Modeling and optimal cyclic scheduling of time-constrained single-robot-arm cluster tools via Petri nets and linear programming," *IEEE Trans. Syst., Man, Cybern.: Syst.*,

vol. 50, no. 3, pp. 871–883, Mar. 2020.
[34] T. S. Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially observed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 47, no. 9, pp. 1491–1495, Sep. 2002.
[35] S. Hashtrudi Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: Incorporating timing information," *IEEE Trans. Autom. Control*, vol. 50, no. 7, pp. 1010–1015, Jul. 2005.
[36] J. N. Zhou, J. C. Wang, and J. Wang, "A simulation engine for stochastic timed Petri nets and application to emergency healthcare systems," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 4, pp. 969–980, Jul. 2019.
[37] K. Z. Zhang and A. Giua, "On detectability of labeled Petri nets and finite automata," *Discrete Event Dyn. Syst.*, vol. 30, pp. 465–497, 2020.
[38] Z. Y. Ma, Z. W. Li, and A. Giua, "Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 1774–1785, Jul. 2015.
[39] Y.-C. Wu and S. Lafortune, "Comparative analysis of related notions of opacity in centralized and coordinated architecturesm" *Discrete Event Dyn. Syst.*, vol. 23 no. 3, pp. 307–339, 2013.
[40] S. L. Shu, F. Lin, and H. Ying, "Detectability of discrete event systems," *IEEE Trans. Autom. Control*, vol. 52, no. 12, pp. 2356–2359, Dec. 2007.
[41] Q. Zhang, C. Seatzu, Z. W. Li, and A. Giua, "Stealthy sensor attacks for plants modeled by labeled Petri nets," in *Proc. 15th IFAC Workshop on Discrete Event Syst.*, Rio de Janeiro, Brazil, pp. 14–20, 2020.