On the Decidability and Complexity of Diagnosability for Labeled Petri Nets

Xiang Yin, *Member, IEEE*, and Stéphane Lafortune, *Fellow, IEEE*

***Abstract*—In this paper, we investigate the decidability and complexity of the fault diagnosis problem in unbounded labeled Petri nets. First, we show that checking diagnosability for unbounded Petri nets is decidable. We present a new necessary and sufficient condition for diagnosability, which can be reduced to a model checking problem for unbounded Petri nets. Then, we show that checking diagnosability for unbounded Petri nets is EXPSPACE-complete. This complexity result is further extended to various subclasses of Petri nets. To the best of our knowledge, this is the first paper that establishes decidability and complexity results for diagnosability of unbounded Petri nets.**

***Index Terms*—Computational complexity, discrete event systems, fault diagnosis, model checking, Petri nets.**

I. Introduction

Fault diagnosis is an important task in large-scale complex systems. In this paper, we investigate the problem of fault diagnosis in a discrete event systems (DES) formalism. The framework of language-based fault diagnosis of DES was initially studied in [1], where the system's behavior is modeled by a finite-state automaton. The notion of *diagnosability* was proposed in order to determine *a priori* if any fault occurrence in the system can be diagnosed online during the operation of the system. Since then, many different approaches for diagnosis using automata models have been investigated; see, e.g., the recent survey [2] and the references therein.

Petri nets are widely used to model many classes of concurrent systems, e.g., manufacturing systems, software programs, and communication networks. Compared with finite-state automata, using Petri nets may have several advantages. First, Petri nets provide a compact representation of a system without enumerating the entire state space. Second, Petri net languages are more expressive than regular languages. Consequently, Petri nets can represent some systems that cannot be represented by finite-state automata. Due to these benefits, the problem of fault diagnosis using Petri nets has received considerable attention in the literature; see, e.g., [3]–[20]. In particular, following the language-based diagnosis framework of [1], diagnosability of unbounded labeled Petri nets was investigated in [17]. Specifically, reference [17] provides a necessary and sufficient condition for diagnosability based on the coverability graph (CG) of a special Petri net called the *verifier net*. However, although a necessary and sufficient condition for diagnosability is provided, the approach proposed in [17] for checking this condition is only sufficient. In other words, the decidability of the diagnosability verification problem for unbounded Petri nets is still open.

In this paper, we revisit the diagnosability verification problem for unbounded Petri nets. Compared with previous works, the contributions of this paper are twofold.

1) First, we show that the diagnosability verification problem for unbounded labeled Petri nets is decidable. To this end, we provide a new necessary and sufficient condition for diagnosability that effectively reduces the diagnosability verification problem to a model checking problem for unbounded Petri nets called the "satisfiability problem of Yen's formula" [21]. In contrast to [17], our new necessary and sufficient condition is presented without using the CG of the Petri net, which allows us to show that the verification problem can be solved with exponential space. Moreover, we relax the previous assumption that the subnet induced by unobservable transitions is acyclic. In other words, the existence of unobservable cycles in the system is allowed.

2) The second contribution of this paper is that, in addition to the decidability result, we establish the precise complexity of the fault diagnosis problem for unbounded Petri nets. It is known that the verification of diagnosability for finite-state automata has polynomial-time complexity [22], [23]. Also, it has been shown that checking diagnosability for timed automata and for pushdown automata are PSPACE-complete [24] and undecidable [25], [26] problems, respectively. However, to the best of our knowledge, the complexity of checking diagnosability for Petri nets is still open. In this paper, we show that checking diagnosability for unbounded Petri nets is EXPSPACE-complete. Moreover, we further investigate some restrictive classes of Petri nets, e.g., free-choice Petri nets and 1-safe Petri nets, and establish the complexity results for these special cases. We show that, even for some very restrictive class of Petri nets, the diagnosability verification problem is still computationally intractable.

II. Preliminaries

A. Petri Nets

A place/transition *net* is defined as a four-tuple $\mathcal{N} = (P, T, A, w)$, where $P = \{p_1, p_2, \ldots, p_{|P|}\}$ is the set of places, $T = \{t_1, t_2, \ldots, t_{|T|}\}$ is the set of transitions, $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs (or flow relation), and $w : A \to \mathbb{N}$ is the weight function that assigns to each arc a non-negative integer. For any place $p \in P$, its preset ${}^\bullet p$ is defined by ${}^\bullet p = \{t \in T : (t, p) \in A\}$ and its postset $p^\bullet$ is defined by $p^\bullet = \{t \in T : (p, t) \in A\}$. The preset ${}^\bullet t$ and the postset $t^\bullet$ for a transition $t \in T$ are defined analogously.

A marking M of a net $\mathcal{N}$ is a vector $M = [M(p_1) \;\; M(p_2) \;\; \ldots \;\; M(p_{|P|})]^\top \in \mathbb{N}^{|P|}$ that assigns to each

Manuscript received December 1, 2016; revised April 13, 2017; accepted April 21, 2017. Date of publication April 28, 2017; date of current version October 25, 2017. This work was supported in part by NSF Grant CNS-1421122 and Grant CNS-1446298. Recommended by Associate Editor C. Hadjicostis. *(Corresponding Author: Xiang Yin).*

X. Yin is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xiangyin.sjtu@gmail.com).

S. Lafortune is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA (e-mail: stephane@umich.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2017.2699278

place $p \in P$ a number of *tokens*. A *Petri net* is a two-tuple $\langle \mathcal{N}, M_0 \rangle$, where $\mathcal{N}$ is a net and M_0 is the initial marking. Given a transition $t \in T$ and a marking M, we say that t is *enabled* at M if $\forall p \in t : M(p) \geq w(p,t)$. If t is enabled, then it may *fire* and yield a new marking $M' = M - w(\cdot, t) + w(t, \cdot)$. We denote by $M \xrightarrow{t}$ that transition $t \in T$ is enabled at M and by $M \xrightarrow{t} M'$ that firing t yields M'. We denote by $R(\mathcal{N}, M_0)$ the set of reachable markings from M_0, i.e., $R(\mathcal{N}, M_0) = \{M : \exists \sigma \in T^* \text{ s.t. } M_0 \xrightarrow{\sigma} M\}$.

Let T^* be the set of all finite sequences of transitions. We say a sequence of transitions $\sigma = t_1 t_2 \ldots t_k \in T^*$ is enabled at M if $\forall i \in \{1, \ldots, k\} : M_i \xrightarrow{t_i}$, where $M_1 = M$ and $M_{i+1} = M_i - w(\cdot, t_i) + w(t_i, \cdot)$. Similarly, we denote by $M \xrightarrow{\sigma}$ that $\sigma \in T^*$ is enabled at M and by $M \xrightarrow{\sigma} M'$ that firing σ yields M'. Given a Petri net $\langle \mathcal{N}, M_0 \rangle$, we denote by $L(\mathcal{N}, M_0)$ the set of *finite* sequences *generated* by $\langle \mathcal{N}, M_0 \rangle$, i.e., $L(\mathcal{N}, M_0) = \{\sigma \in T^* : M_0 \xrightarrow{\sigma}\}$. We denote by λ the empty transition, i.e., for any $\sigma \in T^*$, we have $\sigma\lambda = \lambda\sigma = \sigma$. Let $\sigma \in T^*$ be a sequence of transitions and $t \in T$ be a transition. We denote by $\#_\sigma(t)$ the number of occurrence of transition t in sequence σ. For any sequences σ_1, σ_2, we say that σ_1 is a prefix of σ_2, denoted by $\sigma_1 \leq \sigma_2$, if $\sigma_1 \sigma' = \sigma_2$ for some $\sigma' \in T^*$; we also denote by $\sigma_1 < \sigma_2$ if $\sigma_1 \leq \sigma_2$ and $\sigma_1 \neq \sigma_2$. For any sequence $\sigma \in T^*$, we define $L(\mathcal{N}, M_0)/\sigma := \{\sigma' \in T^* : \sigma\sigma' \in L(\mathcal{N}, M_0)\}$.

Let Σ be a finite set of events and $\Sigma^\epsilon = \Sigma \cup \{\epsilon\}$, where ϵ is the empty string. A *labeled Petri net* is a triple $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$, where $\langle \mathcal{N}, M_0 \rangle$ is a Petri net and $\mathcal{L} : T \to \Sigma^\epsilon$ is a labeling function. We say a transition $t \in T$ is observable if $\mathcal{L}(t) \in \Sigma$ and unobservable if $\mathcal{L}(t) = \epsilon$. We denote by T_o and T_{uo} the set of observable transitions and the set of unobservable transitions, respectively. The labeling function $\mathcal{L}$ is also extended to T^* recursively by $\mathcal{L}(\lambda) = \epsilon$ and $\mathcal{L}(\sigma t) = \mathcal{L}(\sigma)\mathcal{L}(t)$. The language generated by $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is $\mathcal{L}(L(\mathcal{N}, M_0)) := \{\mathcal{L}(\sigma) : \sigma \in L(\mathcal{N}, M_0)\}$.

Given a net $\mathcal{N} = (P, T, A, w)$ and a subset of transitions $T' \subseteq T$, the T'-induced subnet of $\mathcal{N}$ is defined as the new net $\mathcal{N}' = (P, T', A', w')$, where A' and w' are the restriction of A and w to $(P \times T') \cup (T' \times P)$, respectively.

B. Review of Computational Complexity

We briefly review some concepts and results from the theory of computation. We refer the reader to [27] for more details.

We say that a problem is in class PTIME if it can be solved in polynomial time by a deterministic turing machine. A problem is in class NP if it can be solved in polynomial time by a nondeterministic turing machine. Similarly, EXPTIME and NPEXPTIME are the classes of problems that can be solved in exponential time by deterministic turing machines and nondeterministic turing machines, respectively. In addition to time complexity, in many cases, we are also interested in how much memory is required in order to solve a problem. PSPACE and EXPSPACE are the classes of problems that can be solved by deterministic turing machines using polynomial space and exponential space, respectively. It is known that

$$\text{NP} \subseteq \text{PSPACE} \subseteq \text{EXPTIME} \subseteq \text{NPEXPTIME} \subseteq \text{EXPSPACE}. \quad (1)$$

We say that a problem is EXPSPACE-complete if 1) it is in EXPSPACE and 2) any problem in EXPSPACE can be reduced to this problem in polynomial time. We say that a problem is EXPSPACE-hard if there exists a EXPSPACE-complete problem that can be reduced to it in polynomial time. The notions of PSPACE-complete and PSPACE-hard are defined analogously. According to (1), we know that EXPSPACE-complete problems are much more difficult than NP-complete or PSPACE-complete problems, which are already considered as intractable problems.

III. Fault Diagnosis of Labeled Petri Nets

In this section, we review the fault diagnosis problem for labeled Petri nets, as formulated in [17]. In this problem, the goal is to diagnose any fault occurrence unambiguously within a finite delay. To this end, we partition the set of unobservable transitions into two disjoint sets $T_{\text{uo}} = T_f \dot{\cup} T_{\text{reg}}$, where T_f denotes the set of fault transitions. We denote by $\mathcal{N}_N$ the $(T_o \cup T_{\text{reg}})$-induced subnet of $\mathcal{N}$, i.e., $\mathcal{N}_N$ models the nonfaulty behavior of $\mathcal{N}$. We define $\Psi(T_f) = \{\sigma t \in L(\mathcal{N}, M_0) : t \in T_f\}$ to be the set of sequences that end with a fault transition. For any sequence $\sigma = t_1 t_2 \ldots t_n \in T^*$, with a slight abuse of notation, we write that $T_f \in \sigma$ if a fault transition occurs in σ, i.e., $\exists i \in \{1, \ldots, n\} : t_i \in T_f$. We make the following standard assumption in the literature.

A1 $\langle \mathcal{N}, M_0 \rangle$ does not enter a deadlock after a fault transition, i.e., $(\forall \sigma \in \Psi(T_f))(\forall \sigma' \in T^* : M_0 \xrightarrow{\sigma\sigma'} M)(\exists t \in T)[M \xrightarrow{t}]$.

Now, we recall the definition of diagnosability of unbounded Petri nets from [17].

Definition III.1: (Diagnosability). Let $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ be a labeled Petri net. We say that $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is diagnosable w.r.t. T_f if

$$(\forall s \in \Psi(T_f))(\exists n \in \mathbb{N})(\forall v \in L(\mathcal{N}, M_0)/s)[|v| \geq n \Rightarrow D] \quad (2)$$

where the diagnosability condition D is

$$(\forall w \in L(\mathcal{N}, M_0))[\mathcal{L}(w) = \mathcal{L}(sv) \Rightarrow T_f \in w]. \quad (3)$$

Remark III.1: In the above definition, diagnosability is referred to as *uniformly bounded diagnosability* if the universal quantifier term "$\forall s \in \Psi(T_f)$" and the existential quantifier term "$\exists n \in \mathbb{N}$" are swapped. It is known that diagnosability and uniformly bounded diagnosability are equivalent when the system is modeled as a finite-state automaton, namely the system's behavior is a regular language [28]. However, when we consider Petri nets languages, diagnosability is strictly weaker than uniformly bounded diagnosability; an example is provided in [17]. In other words, the diagnosis delay depends on the specific fault string and there does not exist an upper bound for delay in general. For unbounded Petri nets, an effective algorithm for checking uniformly bounded diagnosability was provided in [17] while the decidability of diagnosability is still open. ■

IV. Decidability of Diagnosability

In this section, we first provide a new necessary and sufficient condition for diagnosability in terms of a special formula. Then we show that checking diagnosability for Petri nets is decidable by using a result from [21].

A. Necessary and Sufficient Condition

First, we define the parallel composition of two labeled Petri nets, which is similar to the unlabeled case; see, e.g., [29].

Definition IV.1 (Parallel Composition): Let $N_1 = \langle \mathcal{N}_1, M_{0,1}, \mathcal{L}_1 \rangle$ and $N_2 = \langle \mathcal{N}_2, M_{0,2}, \mathcal{L}_2 \rangle$ be two labeled Petri nets, where $\mathcal{N}_i = (P_i, T_i, A_i, w_i)$ and $\mathcal{L}_i : T_i \to \Sigma_i^\epsilon$ for $i = 1, 2$. Their parallel composition, denoted by $N_1 \| N_2$, is defined as the new labeled Petri net $N_{12} = \langle \mathcal{N}_{12}, M_{0,12}, \mathcal{L}_{12} \rangle, \mathcal{N}_{12} = \langle P_{12}, T_{12}, A_{12}, w_{12} \rangle$, where

1) $P_{12} = P_1 \cup P_2$;
2) $T_{12} \subseteq (T_1 \cup \{\lambda\}) \times (T_2 \cup \{\lambda\})$;
3) A_{12} and w_{12} are defined by
   a) For any $t_1 \in T_1$ and $t_2 \in T_2$ such that $\mathcal{L}_1(t_1) = \mathcal{L}_2(t_2) \in \Sigma_1 \cap \Sigma_2$, we have that $(t_1, t_2) \in T_{12}$ with ${}^\bullet(t_1, t_2) = {}^\bullet t_1 \cup {}^\bullet$

t_2 and $(t_1, t_2)^\bullet = t_1^\bullet \cup t_2^\bullet$. Also

$$w_{12}((t_1, t_2), p) = \begin{cases} w_1(t_1, p) & \text{if } p \in P_1 \\ w_2(t_2, p) & \text{if } p \in P_2 \end{cases} \tag{4}$$

$$w_{12}(p, (t_1, t_2)) = \begin{cases} w_1(p, t_1) & \text{if } p \in P_1 \\ w_2(p, t_2) & \text{if } p \in P_2 \end{cases} \tag{5}$$

b) For any $t_1 \in T_1$ such that $\mathcal{L}_1(t_1) \in (\Sigma_1 \setminus \Sigma_2) \cup \{\epsilon\}$, we have $(t_1, \lambda) \in T_{12}$ with ${}^\bullet(t_1, \lambda) = {}^\bullet t_1$ and $(t_1, \lambda)^\bullet = t_1^\bullet$. For any $p \in P_1$, $w_{12}((t_1, \lambda), p) = w_1(t_1, p)$ and $w_{12}(p, (t_1, \lambda)) = w_1(p, t_1)$.

c) For any $t_2 \in T_2$ such that $\mathcal{L}_2(t_2) \in (\Sigma_2 \setminus \Sigma_1) \cup \{\epsilon\}$, we have $(\lambda, t_2) \in T_{12}$ with ${}^\bullet(\lambda, t_2) = {}^\bullet t_2$ and $(\lambda, t_2)^\bullet = t_2^\bullet$. For any $p \in P_2$, $w_{12}((\lambda, t_2), p) = w_2(t_2, p)$ and $w_{12}(p, (\lambda, t_2)) = w_2(p, t_2)$.

4) $M_{0,12} = \left[M_{0,1}^\top \; M_{0,2}^\top \right]^\top$;

5) $\mathcal{L}_{12} : T_{12} \to \Sigma_1^\epsilon \times \Sigma_2^\epsilon$ is defined by for any $(t_1, t_2) \in T_{12}$, $\mathcal{L}_{12}((t_1, t_2)) = (\mathcal{L}_1(t_1), \mathcal{L}_2(t_2))$.

Remark IV.1: The above-defined parallel composition essentially synchronizes two labeled Petri nets in the following manner. If a transition in one net with event label in $\Sigma_1 \cap \Sigma_2$ is fired, then a transition in the other net with the same label must be fired simultaneously. For each net $i = 1, 2$, if a transition has an event label in $\Sigma_i \setminus \Sigma_j, j \in \{1, 2\} \setminus \{i\}$, or if it is an unobservable transition (i.e., its label is ϵ), then this transition can be freely fired in this net without involving the other net. Note that a sequence in $\mathcal{N}_{12}$ is a tuple of sequences in $\mathcal{N}_1$ and $\mathcal{N}_2$. For any sequence $\sigma \in L(\mathcal{N}_{12}, M_{0,12})$, we denote by $\sigma_1 \in L(\mathcal{N}_1, M_{0,1})$ (respectively, $\sigma_2 \in L(\mathcal{N}_2, M_{0,2})$) the first (respectively, the second) component of σ by absorbing all λ. Then by the definition of parallel composition, we know that, for any $\sigma \in L(\mathcal{N}_{12}, M_{0,12})$, $P_{\Sigma_1 \cap \Sigma_2}(\mathcal{L}_1(\sigma_1)) = P_{\Sigma_1 \cap \Sigma_2}(\mathcal{L}_2(\sigma_2))$, where $P_{\Sigma_1 \cap \Sigma_2}$ is the natural projection $P_{\Sigma_1 \cap \Sigma_2} : (\Sigma_1 \cup \Sigma_2)^* \to (\Sigma_1 \cap \Sigma_2)^*$. Also, for any two sequences $\sigma_1 \in L(\mathcal{N}_1, M_{0,1})$ and $\sigma_2 \in L(\mathcal{N}_2, M_{0,2})$, if $P_{\Sigma_1 \cap \Sigma_2}(\mathcal{L}_1(\sigma_1)) = P_{\Sigma_1 \cap \Sigma_2}(\mathcal{L}_2(\sigma_2))$, then we know that there exists a sequence $\sigma \in L(\mathcal{N}_{12}, M_{0,12})$ such that the first component of σ is σ_1 and the second component of σ is σ_2 (by absorbing all λ). ■

Hereafter, we will consider the parallel composition of Petri net $\langle \mathcal{N}_N, M_0, \mathcal{L} \rangle$, which models the behavior without faults, with the entire Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$, which contains the nonfaulty and the faulty behavior. Since the places of $\mathcal{N}_N$ and $\mathcal{N}$ have the same name, for the sake of clarity, we rename the nonfaulty net $\mathcal{N}_N = \langle P, T_o \cup T_{\text{reg}}, A, w \rangle$, M_0 and $\mathcal{L}$ using distinct symbols by $\mathcal{N}_N = \langle P_N, T_N, A_N, w_N \rangle$, $M_{0,N}$, and $\mathcal{L}_N$, respectively. We still use $\mathcal{N} = \langle P, T, A, w \rangle$ to denote the entire net. We denote by $\langle \mathcal{N}_\|, M_{0,\|}, \mathcal{L}_\| \rangle$ the parallel composition of $\langle \mathcal{N}_N, M_{0,N}, \mathcal{L}_N \rangle$ and $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$, where $\mathcal{N}_\| = \langle P_\|, T_\|, A_\|, w_\| \rangle$, $P_\| = P_N \cup P$, and $T_\| \subseteq (T_N \times T) \cup (T_N \times \{\lambda\}) \cup (\{\lambda\} \times T)$. One can easily verify that the parallel-composed net $\mathcal{N}_\|$ is the same as the verifier net defined in [17].

The following theorem establishes a necessary and sufficient condition for diagnosability. Its proof is provided in the Appendix.

Theorem IV.1: Labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is not diagnosable w.r.t. T_f, if and only if, there exist an ordered subset of places $S = \{p_{k_1}, p_{k_2}, \ldots, p_{k_{|S|}}\} \subseteq P_\|$ in $\mathcal{N}_\|$, an integer $m \in \{0, 1, \ldots, |S|\}$ and a sequence

$$M_{0,\|} \xrightarrow{\sigma'_0} M_1 \xrightarrow{\sigma_1} M'_1 \xrightarrow{\sigma'_1} \cdots \xrightarrow{\sigma'_{m-1}} M_m \xrightarrow{\sigma_m} M'_m \xrightarrow{\sigma'_m} \cdots \tag{6}$$
$$\xrightarrow{\sigma'_{|S|-1}} M_{|S|} \xrightarrow{\sigma_{|S|}} M'_{|S|} \xrightarrow{\sigma'_{|S|}} M_{|S|+1} \xrightarrow{\sigma_{|S|+1}} M'_{|S|+1}$$

in $\mathcal{N}_\|$ such that the following formulas hold simultaneously

$$\bigwedge_{p \in P_\| \setminus S} M'_{|S|+1}(p) \geq M_{|S|+1}(p) \tag{7}$$

$$\bigwedge_{i : 1 \leq i \leq |S|} \left(\begin{matrix} [M'_i(p_{k_i}) > M_i(p_{k_i})] \wedge \\ \left[\bigwedge_{p \in P_\| \setminus \{p_{k_1}, \ldots, p_{k_i}\}} M'_i(p) \geq M_i(p) \right] \end{matrix} \right) \tag{8}$$

$$\bigwedge_{i : 1 \leq i \leq m} \; \bigwedge_{t \in (T_N \cup \{\lambda\}) \times T} \#_{\sigma_i}(t) = 0 \tag{9}$$

$$\bigvee_{t \in \{\lambda\} \times T_f} \#_{\sigma'_m}(t) \geq 1 \tag{10}$$

$$\bigvee_{t \in (T_N \cup \{\lambda\}) \times T} \#_{\sigma_{|S|+1}}(t) \geq 1. \tag{11}$$

Remark IV.2: The intuition of Theorem IV.1 is explained as follows. By Definition III.1, the system is not diagnosable if there exists a fault sequence $v_1 \in \Psi(T_f)$ such that we can find an arbitrarily long continuation of v_1, say v_2, such that $v_1 v_2$ looks the same as some nonfault sequence $u \in L(\mathcal{N}_N, M_{0,N})$. That is, for any $n \in \mathbb{N}$, there exists a sequence in $\langle \mathcal{N}_\|, M_{0,\|} \rangle$ such that its first component is u, its second component is $v_1 v_2$ and $|v_2| \geq n$. Observe that, to fire a sequence σ for an arbitrary number of times, which gives an arbitrarily long sequence, we need to make sure that there are enough tokens in places whose tokens are consumed by firing σ. The sequence in (6) essentially captures this observation. Specifically, S are places in which we need to "store" tokens such that $\sigma_{|S|+1}$ can be fired for an arbitrary given number of times from $M_{|S|+1}$. Formula (7) says that we do not need to "store" tokens for any place in $P_\| \setminus S$ since firing $\sigma_{|S|+1}$ will not consume tokens in these places. Formula (8) essentially encodes that tokens in S can be "stored" by suitably firing each $\sigma_i, i = 1, \ldots, |S|$ for a certain number of times. Integer m denotes the instant where the fault transition occurs. Therefore, Formula (10) simply says that σ'_m contains a fault transition of interest. Formula (11) guarantees that the second component of $\sigma_{|S|+1}$ is not λ, i.e., firing $\sigma_{|S|+1}$ for an arbitrary number of times can yield an arbitrarily long continuation of the fault sequence. Finally, if we need to fire σ_i to "store" tokens in place p_{k_i} before the fault transition, i.e., $i \leq m$, then the second component of σ_i must be λ; otherwise it will change the fault sequence v_1 of interest. This requirement is captured by Formula (9).

Note that the necessary and sufficient condition in Theorem IV.1 includes the case of $S = \emptyset$ and $m = 0$. In this case, since there does not exist an integer i such that $1 \leq i \leq 0$, Formulas (8) and (9) always hold. Therefore, we just need to check the existence of a sequence $M_{0,\|} \xrightarrow{\sigma'_0} M_1 \xrightarrow{\sigma_1} M'_1$ such that Formulas (7), (10), and (11) hold. This situation actually corresponds to the case where $\sigma_{|S|+1}$ can be fired for an arbitrary given number of times directly without "storing" tokens in any place.

Let us illustrate Theorem IV.1 by the following example.

Example IV.1: Let us consider the labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ shown in Fig. 1 (a), where $T_o = \{t_1, t_2, t_3\}$, $T_{\text{reg}} = \{\epsilon_1\}$, and $T_f = \{f_1\}$. Also, let $\mathcal{L}(t_1) = a$ and $\mathcal{L}(t_2) = \mathcal{L}(t_3) = b$. Its $(T_o \cup T_{\text{reg}})$-induced net $\langle \mathcal{N}_N, M_{0,N}, \mathcal{L}_N \rangle$ is shown in Fig. 1(b). For the sake of clarity, we use p_i^N and t_i^N to denote a place and a transition in $\mathcal{N}_N$, respectively. We do not depict place p_5^N and transition t_3^N in $\mathcal{N}_N$ since they are not involved in $\mathcal{N}_N$ after removing t_f. Then the parallel-composed labeled Petri net $\langle \mathcal{N}_\|, M_{0,\|}, \mathcal{L}_\| \rangle = \langle \mathcal{N}_N, M_{0,N}, \mathcal{L}_N \rangle \| \langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is shown in Fig. 1(c). Note that the parallel composition does not depend on the initial marking and we omit p_5^N and t_3^N here just for this specific verification problem. This system is not

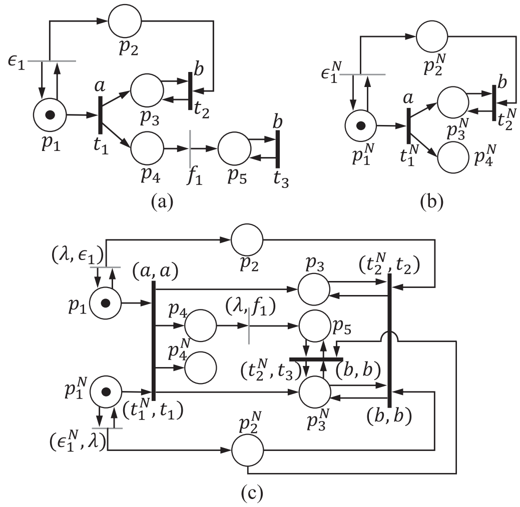

Fig. 1. $T_o = \{t_1, t_2, t_3\}, T_{\mathsf{reg}} = \{\epsilon_1\}$, and $T_f = \{f_1\}$. (a) $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$. (b) $\langle \mathcal{N}_N, M_{0,N}, \mathcal{L}_N \rangle$. (c) $\langle \mathcal{N}_\|, M_{0,\|}, \mathcal{L}_\| \rangle = \langle \mathcal{N}_N, M_{0,N}, \mathcal{L}_N \rangle \| \langle \mathcal{N}, M_0, \mathcal{L} \rangle$.

diagnosable, since for fault sequence $t_1 t_f \in \Psi(T_f)$, for any integer $n \in \mathbb{N}$, we can find sequences $(\epsilon_1)^n t_1 (t_2)^n \in L(\mathcal{N}_N, M_{0,N})$ and $(t_3)^n \in L(\mathcal{N}, M_0)/t_1 t_f$ such that $\mathcal{L}((\epsilon_1)^n t_1 (t_2)^n) = \mathcal{L}(t_1 t_f (t_3)^n) = ab^n$.

Now, let us show the system is not diagnosable using Theorem IV.1. We choose the following sequence that can be fired from the initial marking:

$$M_{0,\|} \xrightarrow{\overbrace{(\epsilon_1^N, \lambda)}^{:=\sigma'_0}} M_1 \xrightarrow{\overbrace{(\epsilon_1^N, \lambda)}^{:=\sigma_1}} M'_1 \xrightarrow{\overbrace{(t_1^N, t_1)(\lambda, f_1)}^{:=\sigma'_1}} M_2 \xrightarrow{\overbrace{(t_2^N, t_3)}^{:=\sigma_2}} M'_2$$

where the places in each marking are ordered by $\{p_1^N, p_2^N, p_3^N, p_4^N, p_1, p_2, p_3, p_4, p_5\}$ and

$$\begin{array}{rl}
M_1 = & [\begin{array}{cccc:ccccc} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}]^\top \\
M'_1 = & [\begin{array}{cccc:ccccc} 1 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{array}]^\top \\
M_2 = & [\begin{array}{cccc:ccccc} 0 & 2 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}]^\top \\
M'_2 = & [\begin{array}{cccc:ccccc} 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{array}]^\top .
\end{array}$$

Let us choose $S = \{p_2^N\}$ and $m = 1$. First, Formula (7) holds for the above sequence, since $\forall p \in P_\| \setminus \{p_2^N\} : M_2(p)' \geq M_2(p)$. Also, we have $M'_1 \geq M_1$ and $M'_1(p_2^N) > M_1(p_2^N)$, i.e., Formula (8) holds. Moreover, Formula (9) holds since $\sigma_1 = (\epsilon_1^N, \lambda)$, which does not contain a transition in $(T_N \cup \{\lambda\}) \times T$. Formula (10) also holds since $\sigma'_m = \sigma'_1 = (t_1^N, t_1)(\lambda, f_1)$, which contains a transition in $\{\lambda\} \times T_f$. Finally, Formula (11) holds since $\sigma_{|S|+1} = \sigma_2 = (t_2^N, t_3)$, which contains a transition in $(T_N \cup \{\lambda\}) \times T$. Since all formulas in Theorem IV.1 hold, we know that the system is not diagnosable. ■

Remark IV.3: In the above example, we see that sequence (t_2^N, t_3) consumes a token from place p_2^N. However, it can be fired for an arbitrary number of times if we "store" enough tokens in p_2^N by (silently) firing sequence (ϵ_1^N, λ) for an arbitrary number of times. Moreover, firing (ϵ_1^N, λ) will not affect the fault sequence of interest since it only contributes λ to the second component. This example suggests the following phenomenon in Petri nets. By Definition III.1, the system is not diagnosable if there exists a fault sequence $v_1 \in \Psi(T_f)$ such that, for any $n \in \mathbb{N}$, there exist $v_2 \in L(\mathcal{N}, M_0)/v_1$ and $u_1 u_2 \in L(\mathcal{N}_N, M_{0,\|})$ such that $\mathcal{L}(v_1) = \mathcal{L}(u_1)$ and $\mathcal{L}(v_2) = \mathcal{L}(u_2)$. However, u_1 may not be fixed for a given v_1 and it may depend on which v_2 we choose. Due to the presence of an arbitrarily long sequence that only consists of transitions in T_{uo}, u_1 can be arbitrarily long without changing its fixed observation. This issue does not exist in automata since we can always remove unobservable cycles in a sequence. However, the unobservable sequence in u_1 cannot be removed arbitrarily, since we may need to "store" tokens by firing this sequence. This phenomenon makes the verification problem much more challenging when the T_{uo}-induced net is not acyclic.

Remark IV.4: Note that the parallel composition of the nonfaulty net $\mathcal{N}_N$ and the entire net $\mathcal{N}$ is the same as the verifier net defined in [17]. However, compared with [17], our new necessary and sufficient condition has the following important features. First, our necessary and sufficient condition does not rely on the assumption that the T_{uo}-induced net is acyclic, which is required in [17]. In other words, the existence of unobservable cycles, which is more difficult to handle in Petri nets, is allowed. Second, our necessary and sufficient condition is stated in terms of a special formula. We will show later that this allows the necessary and sufficient condition to be effectively checked, while the linear programming approach proposed in [17] can only verify the sufficiency part of the necessary and sufficient condition therein. Finally, in contrast to [17], the statement of our necessary and sufficient condition does not rely on the CG of the Petri net. It is known that the complexity of the CG is not even in primitive recursive space [30], [31], which implies that constructing the CG requires even more than exponential space. However, our condition avoids using the CG. As we will discuss later, this further allows us to establish the two results that 1) checking diagnosability is decidable and that 2) it has an exponential-space upper bound for its complexity. ■

B. Checking the Necessary and Sufficient Condition

Now, let us discuss how to check the existence or the nonexistence of the sequence in Theorem IV.1. Specifically, we show that checking the necessary and sufficient condition is essentially a special case of a model checking problem studied by Yen [21].

In [21], Yen studied the model checking problem of a class of formulas for unbounded Petri nets. The problem is formulated as follows.

Definition IV.2 (Yen's Problem): Given a general unbounded Petri net $\langle \mathcal{N}, M_0 \rangle$, decide whether or not there exists a sequence

$$M_0 \xrightarrow{\sigma_1} M_1 \xrightarrow{\sigma_2} \ldots M_{k-1} \xrightarrow{\sigma_k} M_k \tag{12}$$

such that a formula $F(M_1, \ldots, M_k, \sigma_1, \ldots, \sigma_k)$ holds. More specifically, $F(M_1, \ldots, M_k, \sigma_1, \ldots, \sigma_k)$ is a formula obtained from the following syntax:[1]

S-1 For any markings M_i, M_j, constant c and places $p, p' \in P$, $M_i(p) \geq c, M_i(p) > c, M_i(p) = M_j(p'), M_i(p) > M_j(p')$, and $M_i(p) < M_j(p')$ are formulas.

S-2 For any sequences σ_i, σ_j, constant c and transitions $t, t' \in T$, $\#_{\sigma_i}(t) \leq c, \#_{\sigma_i}(t) \geq c$ and $\#_{\sigma_i}(t) \leq \#_{\sigma_j}(t')$ are formulas.

S-3 For any formulas F_1 and F_2, $F_1 \wedge F_2$, and $F_1 \vee F_2$ are formulas.

It was shown in [21] that 1) the above problem is decidable; and 2) solving this problem requires exponential space in the size of $\mathcal{N}$. In fact, Yen's formula is very powerful, since many well-known problems can be reformulated in terms of Definition IV.2; one such example is the coverability problem. Note that the general Petri net reachability problem cannot be solved by Yen's result, since $M_i(p) = c$ is not a valid formula.

Let us return to the diagnosability verification problem. Clearly, the necessary and sufficient condition presented in Theorem IV.1 is a valid formula in Definition IV.2. Moreover, the size of the composed net $\mathcal{N}_\|$ is polynomial in the size of the original net $\mathcal{N}$. Also, given an ordered subset S and an integer m, the size of the formula is also polynomial in the size of $\mathcal{N}_\|$. Since Yen's problem can be solved in exponential space, checking the formulas in Theorem for given S and m can be done in exponential space. To check diagnosability, it

[1] The original syntax in [21] is a bit more general.

suffices to enumerate all possible S and m, i.e., we need to repeat the above EXPSPACE procedure for $\sum_{k=0}^{|P_\|\|}(k+1)!\binom{|P_\|\|}{k}$ times, which still requires exponential space. Overall, we have the following result.

Theorem IV.2: Checking diagnosability for labeled Petri nets is decidable. Moreover, it is in EXPSPACE.

Remark IV.5: How to solve Yen's problem is beyond the scope of this paper, since our goal is to show that diagnosability of unbounded Petri nets is decidable. However, it may be useful to discuss the general idea of Yen's solution approach. In fact, Yen's approach is a generalization of the results of Rackoff in [31], which show that the coverability problem for unbounded Petri nets is EXPSPACE-complete. Specifically, Yen showed that, if a formula in the form in Definition IV.2 is satisfiable, then there must exist a sequence whose length is bounded by $O(2^{2^{D\times N\times\log N}})$ such that the formula is satisfied, where D is a constant and N denotes the size of the net and the formula. In other words, in order to check whether F is satisfiable or not, it suffices to search a bounded reachable set, rather than the entire unbounded set of reachable markings. Moreover, such a search can be implemented in a nondeterministic manner, which only requires $O(2^{N\times\log N})$ space, i.e., this problem is in EXPSPACE. A similar approach is also used in [32] in order to study the complexity of the linear temporal logic (LTL) model checking problem for *vector addition systems with states*, a model known to be equivalent to Petri nets. We refer the reader to the very comprehensive survey [33] for more details on this issue. The only results we need for our purposes in this paper are: 1) checking the condition in Definition IV.2 is decidable; and 2) it can be done by using exponential space. ■

V. EXPSPACE-Completeness of Diagnosability

In the preceding section, we have shown that checking diagnosability for labeled Petri nets can be mapped to an instance of the satisfiability problem of Yen's formula, which can be solved by using exponential space. One may ask whether or not this complexity can be further improved. In this section, we will answer this question. Specifically, we show that checking diagnosability for labeled Petri nets is EXPSPACE-complete. In other words, this extremely high complexity seems to be unavoidable.

A. General Case

In the analysis of unbounded Petri nets, one of the biggest challenges is the well-known exponential space lower bound proved by Lipton [34], which results in the EXPSPACE-hardness of many fundamental problems in Petri nets. Here, we recall a well-known EXPSPACE-complete problem for unbounded Petri nets [31], [34].

Coverability Problem

1) INSTANCE: A Petri net $\langle\mathcal{N},M_0\rangle$ and a marking M.
2) QUESTION: Whether or not there exists a reachable marking $M'\in R(\mathcal{N},M_0)$ such that $M\le M'$.

We use the coverability problem to show that checking diagnosability for unbounded Petri nets is EXPSPACE-complete.

Theorem V.1: Checking diagnosability for labeled Petri nets is EXPSPACE-complete.

Proof: We have already shown that this problem is in EXPSPACE. Hereafter, we show that it is EXPSPACE-hard by reducing the coverability problem to the diagnosability verification problem.

Let $\langle\mathcal{N}=(P,T,A,w),M_0\rangle$ and M be the instance of the coverability problem. Then we construct a labeled Petri net $\langle\hat{\mathcal{N}}=(\hat{P},\hat{T},\hat{A},\hat{w}),\hat{M}_0,\hat{\mathcal{L}}\rangle$ from $\langle\mathcal{N},M_0\rangle$ and M as follows.

1) $\hat{P}=P\cup\{p_f\}$, where $p_f\notin P$ is a new fault place;
2) $\hat{T}=T\cup\{t_f,t_{\text{uo}}\}$, where $t_f,t_{\text{uo}}\notin T$ are two new transitions;
3) $\hat{A}$ and $\hat{w}$ are obtained from A and w by adding the following arcs and weights
   a) An arc from each place $p\in P$, where $M(p)\neq 0$, to the fault transition t_f with $\hat{w}(p,t_f)=M(p)$;
   b) An arc from the fault transition t_f to the fault place p_f with $\hat{w}(t_f,p_f)=1$;
   c) An unobservable self-loop transition t_{uo} at the fault place $p_f\in\hat{P}$, i.e., two arcs (t_{uo},p_f) and (p_f,t_{uo}) with $\hat{w}(t_{\text{uo}},p_f)=\hat{w}(p_f,t_{\text{uo}})=1$.
4) $\hat{M}_0=[M_0^\top\ 0]^\top$; [2]
5) The labeling function $\hat{\mathcal{L}}:\hat{T}\to\Sigma^\epsilon$ is defined by $\hat{\mathcal{L}}(t)=a$ if $t\in T$ and $\hat{\mathcal{L}}(t)=\epsilon$ if $t\in\{t_f,t_{\text{uo}}\}$, where $\Sigma=\{a\}$ is the set of events. In other words, $\hat{T}_o=T$ and $\hat{T}_{\text{uo}}=\{t_f,t_{\text{uo}}\}$.

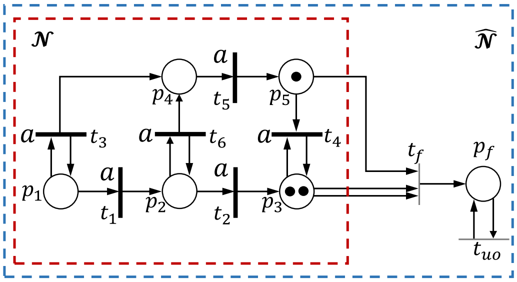

Fig. 2. Conceptual illustration of how to construct $\langle\hat{\mathcal{N}},M_0,\hat{\mathcal{L}}\rangle$ from $\langle\mathcal{N},M_0\rangle$, where we are interested in whether or not marking $M=[0\ 0\ 2\ 0\ 1]^\top$ is covered in $\mathcal{N}$. Since M is the initial marking of $\langle\mathcal{N},M_0\rangle$, we know that it is covered immediately. Therefore, $\langle\hat{\mathcal{N}},M_0,\hat{\mathcal{L}}\rangle$ is not diagnsoable.

By the above construction, M is reachable in $\mathcal{N}$ if and only if $[M^\top\ 0]^\top$ is reachable in $\hat{\mathcal{N}}$. Moreover, the self-loop at p_f guarantees that $\hat{\mathcal{N}}$ will not reach a deadlock marking after the fault transition, which implies that diagnosability for $\langle\hat{\mathcal{N}},\hat{M}_0,\hat{\mathcal{L}}\rangle$ is well defined. Fig. 2 provides a conceptual illustration showing how $\hat{\mathcal{N}}$ is constructed from $\mathcal{N}$. Clearly, constructing $\hat{\mathcal{N}}$ is linear in the size of $\mathcal{N}$. By our construction, the fault transition can occur if M can be covered. Moreover, once the fault occurs, we cannot diagnose it. Therefore, $\langle\mathcal{N},M_0\rangle$ covers M if and only if $\langle\hat{\mathcal{N}},\hat{M}_0,\hat{\mathcal{L}}\rangle$ is not diagnosable w.r.t. $T_f:=\{t_f\}$, which is proved as follows.

($\Rightarrow$) Suppose that $\langle\mathcal{N},M_0\rangle$ covers M, which implies that $\langle\hat{\mathcal{N}},\hat{M}_0\rangle$ covers $[M^\top\ 0]^\top$. Let $\sigma\in L(\hat{\mathcal{N}},\hat{M}_0)$ be a sequence such that $T_f\notin\sigma$, $\hat{M}_0\xrightarrow{\sigma}M_1$ and $[M^\top\ 0]^\top\le M_1$. By construction, we have that $\forall p\in{}^\bullet t_f: M_1(p)\ge M(p)=\hat{w}(p,t_f)$, which implies that t_f is enabled at M. Moreover, unobservable transition t_{uo} can be fired at any reachable marking for an arbitrary number of times. Therefore, we have that

$$(\exists\sigma t_f\in\Psi(T_f))(\forall n\in\mathbb{N})(\exists t_{\text{uo}}^n\in L(\hat{\mathcal{N}},\hat{M}_0)/(\sigma t_f))\tag{13}$$
$$[|t_{\text{uo}}^n|\ge n\wedge(\exists\sigma\in L(\hat{\mathcal{N}},\hat{M}_0))[\hat{\mathcal{L}}(\sigma)=\hat{\mathcal{L}}(\sigma t_f t_{\text{uo}}^n)\wedge T_f\notin\sigma].$$

Therefore, we know that $\langle\hat{\mathcal{N}},\hat{M}_0,\hat{\mathcal{L}}\rangle$ is not diagnosable.

($\Leftarrow$) By contraposition. Suppose that $\langle\mathcal{N},M_0\rangle$ does not cover M, i.e., $\langle\mathcal{N}',M_0\rangle$ does not cover M. Then we know that $\forall M'\in R(\hat{\mathcal{N}},\hat{M}_0):[M^\top\ 0]^\top\not\le M'$. By the construction of $\hat{\mathcal{N}}$, we know that transition t_f cannot be fired in $\langle\hat{\mathcal{N}},\hat{M}_0\rangle$, which implies that $\forall\sigma\in L(\hat{\mathcal{N}},\hat{M}_0):t_f\notin\sigma$. Therefore, $\langle\hat{\mathcal{N}},\hat{M}_0,\hat{\mathcal{L}}\rangle$ is clearly diagnosable. ■

B. Special Cases

In the development of the preceding EXPSPACE-completeness result, we reduced the coverability problem to the diagnosability verification problem. This reduction is applicable to any class of Petri nets.

[2] We assume that the place order in $\hat{\mathcal{N}}$ is the same as the place order in $\mathcal{N}$ except for the last place p_f.

Based on this observation, we establish complexity results for certain special classes of Petri nets. First, we recall some standard definitions.

Definition V.1: A net $\langle \mathcal{N}, M_0 \rangle$ is said to be

1) free-choice, if $\forall a \in A : w(a) = 1$ and $\forall p \in P : |p^\bullet| \leq 1$ or ${}^\bullet(p^\bullet) = \{p\}$;
2) 1-safe, if $\forall M \in R(\mathcal{N}, M_0), \forall p \in P : M(p) \leq 1$;
3) conflict-free, if $\forall p \in P : |p^\bullet| > 1 \Rightarrow p^\bullet \subseteq {}^\bullet p$;
4) a state-machine net, if $\forall a \in A : w(a) = 1$ and $\forall t \in T : |{}^\bullet t| = |t^\bullet| = 1$.
5) a marked graph, if $\forall a \in A : w(a) = 1$ and $\forall p \in P : |{}^\bullet p| = |p^\bullet| = 1$;

It was shown in [35] that the coverability problem for free-choice Petri nets, a relatively restrictive class of Petri nets, is still EXPSPACE-complete, although checking liveness for free-choice Petri nets is NP-complete. Therefore, we know that checking diagnosability for labeled free-choice Petri nets is EXPSPACE-complete. Another important class of Petri nets is that of 1-safe Petri nets. It is known that the coverability problem is still PSPACE-complete for 1-safe Petri nets [35]. Hence, we know that checking diagnosability for labeled 1-safe Petri nets is PSPACE-hard.

One may ask whether or not there exist restricted classes of Petri nets for which diagnosability can be checked in polynomial time. The answer is positive. For example, it was shown in [33] that, if the Petri net is both 1-safe and conflict-free, then Yen's problem can be efficiently solved in polynomial time. Therefore, checking diagnosability for labeled 1-safe conflict-free Petri nets is in PTIME. Another class of Petri nets for which this result holds is that of state-machine nets, which are equivalent to finite-state automata. In this case, the known results developed for regular languages [22], [23] apply and we can state that checking diagnosability for state-machine nets is in PTIME. One interesting future direction on this topic is to show whether or not diagnosability of marked graphs can be checked in polynomial time.

VI. Conclusion

In this paper, we showed that checking diagnosability of unbounded Petri nets is decidable. Moreover, we showed that this problem is EXPSPACE-complete. This result reveals that although Petri nets provide a compact way for modeling systems, in order to analyze diagnosability of a Petri net, an extremely high computational complexity still seems to be unavoidable. This computational intractability result also suggests the following future research directions. First, one may be interested in finding more subclasses of Petri nets for which checking diagnosability is tractable. Finding sufficient conditions for diagnosability of general Petri nets by using structural analysis may also be an interesting topic for future investigations.

Acknowledgment

The authors would like to thank the anonymous reviewers for their useful comments on improving this paper. In particular, they sincerely thank an anonymous reviewer who pointed out a mistake in an earlier version of Theorem IV.1.

Appendix

A. Proof of Theorem IV.1

First, we prove the sufficiency of Theorem IV.1.

Proof: (*The Sufficiency of Theorem IV.1*) Suppose that there exists a sequence in (6) satisfying Formulas (7)–(11) simultaneously.

Based on the sequence in (6), we construct the following new sequence in $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$

$$M_{0,\parallel} \xrightarrow{\sigma'_0} \hat{M}_1 \xrightarrow{(\sigma_1)^{n_1}} \hat{M}'_1 \xrightarrow{\sigma'_1} \cdots \xrightarrow{\sigma'_{m-1}} \hat{M}_m \xrightarrow{(\sigma_m)^{n_m}} \hat{M}'_m \xrightarrow{\sigma'_m}$$
$$\cdots \xrightarrow{\sigma'_{|S|-1}} \hat{M}_{|S|} \xrightarrow{\left(\sigma_{|S|}\right)^{n_{|S|}}} \hat{M}'_{|S|} \xrightarrow{\sigma'_{|S|}} \hat{M}_{|S|+1} \xrightarrow{\left(\sigma_{|S|+1}\right)^{n}} \hat{M}'_{|S|+1}. \quad (14)$$

We claim that, for any $n \in \mathbb{N}$, the above sequence is well-defined in $\mathcal{N}_\parallel$ for some positive integers $n_1, \ldots, n_{|S|} \in \mathbb{N}$. To see this, we proceed inductively as follows. By Formula (8), we know that $M_1 \leq M'_1$ and $M_1(p_{k_1}) < M'_1(p_{k_1})$. Therefore, sequence $M_{0,\parallel} \xrightarrow{\sigma'_0 (\sigma_1)^{n_1} \sigma'_1}$ is well-defined for any n_1. Moreover, we can make $\hat{M}_2(p_{k_1})$ arbitrarily large by choosing n_1 to be sufficiently large. For sequence σ_2, still by Formula (8), we know that $\forall p \in P_\parallel \setminus \{p_{k_1}\} : M_2(p) \leq M'_2(p)$ and $M_2(p_{k_2}) < M'_2(p_{k_2})$. Note that, it is possible that $M'_2(p_{k_1}) < M_2(p_{k_1})$, i.e., firing σ_2 may consume tokens in place p_{k_1}. However, we can "store" enough tokens in p_{k_1} by taking a sufficiently large n_1 before $(\sigma_2)^{n_2}$ is fired. Therefore, we can choose n_1 and n_2 such that 1) $M_{0,\parallel} \xrightarrow{\sigma'_0 (\sigma_1)^{n_1} \sigma'_1 (\sigma_2)^{n_2} \sigma'_2}$ is well-defined; 2) $\forall p \in P_\parallel \setminus \{p_{k_1}, p_{k_2}\} : M_3(p) \leq \hat{M}_3(p)$; and 3) both $\hat{M}_3(p_{k_1})$ and $\hat{M}_3(p_{k_2})$ can be arbitrarily large. By inductively applying the above argument, we know that we can choose $n_1, \ldots, n_{|S|}$, such that $\forall p \in S : \hat{M}_{|S|+1}(p) \geq n \times (M_{|S|+1}(p) - M'_{|S|+1}(p))$ and $\forall p \in P_\parallel \setminus S : \hat{M}_{|S|+1}(p) \geq M_{|S|+1}(p)$. Recall that, by Formula (7), we have $\forall p \in P_\parallel \setminus S : M_{|S|+1}(p) \leq M'_{|S|+1}(p)$. Therefore, $(\sigma_{|S|+1})^n$ can be fired from $\hat{M}_{|S|+1}$, i.e., for any $n \in \mathbb{N}$, we can choose $n_1, \ldots, n_{|S|}$ such that the sequence in (14) is well-defined.

Recall that each sequence in (6) or (14) is a tuple. For each σ_i, we still denote by $\sigma_{i,1}$ its first component and by $\sigma_{i,2}$ its second component, where $\sigma_{i,1} \in T_N^*$ and $\sigma_{i,2} \in T^*$; the same notation for σ'_i. By Formula (10), we know that $T_f \in \sigma'_{m,2}$. We write $\sigma'_{m,2} = \sigma'_{m,F} \sigma'_{m,C}$ such that the last transition of $\sigma_{m,F}$ is in T_f. For the sake of simplicity, we define

$$\alpha := \sigma'_{0,2} (\sigma_{1,2})^{n_1} \sigma'_{1,2} \ldots \sigma'_{m-1,2} (\sigma_{m,2})^{n_m} \ \sigma'_{m,F}$$
$$\beta := \sigma'_{m,C} \ (\sigma_{m+1,2})^{n_{m+1}} \ldots \sigma'_{|S|-1,2} \left(\sigma_{|S|,2}\right)^{n_{|S|}} \sigma'_{|S|,2} \left(\sigma_{|S|+1,2}\right)^{n}$$
$$\gamma := \sigma'_{0,1} \ (\sigma_{1,1})^{n_1} \ \sigma'_{1,1} \ldots \sigma'_{|S|-1,1} \left(\sigma_{|S|,21}\right)^{n_{|S|}} \sigma'_{|S|,21} \left(\sigma_{|S|+1,1}\right)^{n}.$$

By the definition of parallel composition, we know that $\mathcal{L}(\alpha\beta) = \mathcal{L}(\gamma)$ for any $n_1, \ldots, n_{|S|}$. Moreover, by Formula (9), we know that $\sigma_1, \ldots, \sigma_m \in (T_N \times \{\lambda\})^*$, i.e., $\sigma_{1,2} = \sigma_{2,2} = \cdots = \sigma_{m,2} = \lambda$. Therefore, $\alpha = \sigma'_{0,2} \sigma'_{1,2} \ldots \sigma'_{m-1,2} \sigma'_{m,2}$ for any $n_1, \ldots, n_m$. That is, the choice of n_i does not change the fault transition $\alpha \in \Psi(T_f)$; however, β and γ may be changed by choosing different n_i.

Recall that the sequence in (14) is defined for any $n \in \mathbb{N}$. Therefore, we know that

$$(\exists \alpha \in \Psi(T_f))(\forall n \in \mathbb{N})(\exists \beta \in L(\mathcal{N}, M_0)/\alpha)$$
$$\text{s.t. } |\beta| \geq n \text{ and } (\exists \gamma \in L(\mathcal{N}_N, M_0))[\mathcal{L}(\alpha\beta) = \mathcal{L}(\gamma)].$$

Note that $|\beta| \geq n$ comes from Formula (11), i.e., $\sigma_{|S|+1,2} \neq \lambda$. Therefore, $|\beta| \geq n \times |\sigma_{|S|+1,2}| \geq n$. Overall, we know that $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is not diagnosable. ■

To prove the necessity, we need to use a modified version of the coverability tree/graph for the parallel composed net $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$. Following the standard notation, we denote by ω "infinity" such that for any $n \in \mathbb{N}, \omega > n, \omega \pm n = \omega$, and $\omega \geq \omega$. Let $\sigma \in T^*$ be a sequence in the original net $\langle \mathcal{N}, M_0 \rangle$. Then the modified coverability tree for $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$ w.r.t. σ, denoted by $CT_M(\mathcal{N}_\parallel, M_{0,\parallel}, \sigma)$, is constructed according to Algorithm 1. For each node q, $\Xi(q)$ is used to track the second component of the sequence that leads to q in the tree. Intuitively, the modified coverability tree follows the same construction rules of the standard coverability tree [36] except the following constraint: "*ω cannot be added to any place in $P \subset P_\parallel = P_N \cup P$ if the second component is a prefix of σ.*" In other words, if σ has not been fully executed in the second component of the sequence, then we will keep adding the integer in any place in P rather than introducing ω even if we have a covering. By Dickson's Lemma [37], we

Algorithm 1: Construction of $CT_M(\mathcal{N}_\|, M_{0,\|}, \sigma)$.

```
Label the root node $q_0$ by $M_{0,\|}$, tag it "new" and set $\Xi(q_0)=\lambda$;
while exists a "new" node do
    Select a "new" node $q$ and let $M$ be its label;
    for $(t_1,t_2) \in T_\| : M \xrightarrow{(t_1,t_2)} M'$ do
        if $\Xi(q) < \sigma$ then
            if $q$ is reached from a node $q''$ with label $M''$ s.t. $M'' < M'$ and the path from $q''$ to $q$ only consists of transitions in $T_N \times \{\lambda\}$ then
                $\forall p \in P_\| : M''(p) < M'(p)$ replace $M'(p)$ by $\omega$;
        else
            if $q$ is reached from a node $q''$ with label $M''$ s.t. $M'' < M'$ then
                $\forall p \in P_\| : M''(p) < M'(p)$ replace $M'(p)$ by $\omega$;
        Add a new node, say $q'$, with label $M'$, add an arc with label $(t_1,t_2)$ from $q$ to $q'$ and set $\Xi(q') = \Xi(q)t_2$;
        if there exists a node with label $M'$ in the tree then
            tag node $q'$ "duplicate"
        else
            tag node $q'$ "new"
    Untag node $q$.
```

know that $CT_M(\mathcal{N}_\|, M_{0,\|}, \sigma)$ is also finite for any finite σ. Essentially, the modified coverability tree "unfolds" the standard coverability tree along the sequence whose second component is σ. By taking $\sigma = \lambda$, $CT_M(\mathcal{N}_\|, M_{0,\|}, \lambda)$ is the standard coverability tree. We define the modified CG for $\langle \mathcal{N}_\|, M_{0,\|} \rangle$ w.r.t. σ as the graph obtained by merging each duplicated node with the untagged node that has the same label and denote it by $CG_M(\mathcal{N}_\|, M_{0,\|}, \sigma)$.

Now we are ready to complete the proof of Theorem IV.1.

Proof: (*The Necessity of Theorem IV.1*) Suppose that $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is not diagnosable. Then we know that, there exists $v_1 t_f \in L(\mathcal{N}, M_0)$, where $t_f \in T_f$, such that, for any $n \in \mathbb{N}$, there exist $v_1 t_f v_2 \in L(\mathcal{N}, M_0)$ and $u \in L(\mathcal{N}_N, M_0)$ such that: 1) $|v_2| \geq n$; and 2) $\mathcal{L}(u) = \mathcal{L}(v_1 t_f v_2)$.

Let $CG_M(\mathcal{N}_\|, M_{0,\|}, v_1 t_f)$ be the modified CG for $\langle \mathcal{N}_\|, M_{0,\|} \rangle$ w.r.t. $v_1 t_f$. Then we choose $n = |CG_M(\mathcal{N}_\|, M_{0,\|}, v_1 t_f)| + 1$, which is greater than the number of nodes in the modified CG. For the above sequence $v_1 t_f$ and n, let v_2 and u be the sequences such that $|v_2| \geq n$ and $\mathcal{L}(u) = \mathcal{L}(v_1 t_f v_2)$. By the definition of parallel composition, we know that there exists a sequence $\alpha\beta \in L(N_\|, M_{0,\|})$ such that $\alpha_1\beta_1 = u, \alpha_2 = v_1 t_f$ and $\beta_2 = v_2$, where α_1 and α_2 denote the first and the second components of α, respectively, and the same for β. We write $\alpha\beta = t_1 t_2 \ldots t_{|\alpha|} t_{|\alpha|+1} t_{|\alpha|+2} \ldots t_{|\alpha|+|\beta|}$, where $t_i \in T_\| \subseteq (T_N \cup \{\lambda\}) \times (T \cup \{\lambda\})$. Let

$$M_{0,\|}, M_1, M_2, \ldots, M_{|\alpha|}, M_{|\alpha|+1}, M_{|\alpha|+2}, \ldots, M_{|\alpha|+|\beta|} \tag{15}$$

be the marking labels of the nodes reached long $t_1 \ldots t_{|\alpha|} t_{|\alpha|+1} \ldots t_{|\alpha|+|\beta|}$ in $\mathrm{CG}_M(\mathcal{N}_\|, M_{0,\|}, v_1 t_f)$. Note that M_i may contain ω. Let

$$M_{|\alpha|+\theta_1}, M_{|\alpha|+\theta_2}, \ldots, M_{|\alpha|+\theta_{|v_2|}}, 1 \leq \theta_1 < \cdots < \theta_{|v_2|} \leq |\beta| \tag{16}$$

be the nodes in $M_{|\alpha|+1}, \ldots, M_{|\alpha|+|\beta|}$ that are reached immediately after transitions in $(T_N \cup \{\lambda\}) \times T$, i.e., transitions whose second (fault) component is not λ. We know there are $|v_2|$ number of these nodes from $M_{|\alpha|+1}$ to $M_{|\alpha|+|\beta|}$ since $\beta_2 = v_2$. Moreover, since v_2 is chosen such that $|v_2| \geq n = |\mathrm{CG}_M(\mathcal{N}_\|, M_{0,\|}, v_1 t_f)| + 1$, we know that there exist two nodes $M_{|\alpha|+\theta_i}$ and $M_{|\alpha|+\theta_j}$, where $1 \leq \theta_i < \theta_j \leq |\beta|$, such that $M_{|\alpha|+\theta_i} = M_{|\alpha|+\theta_j}$.

Let $S := \{p \in P_\| : M_{|\alpha|+\theta_i}(p) = \omega\}$ be the set of places which are ω in $M_{|\alpha|+\theta_i}$ or $M_{|\alpha|+\theta_j}$. First, we assume that $S \neq \emptyset$. For each $p \in S$, we denote by $\mu[p]$ the instant at which ω in place p is obtained, i.e., $M_{\mu[p]-1}(p) \neq \omega$ and $M_{\mu[p]}(p) = \omega$. We also denote by $\tilde{\mu}[p]$, where $\tilde{\mu}[p] < \mu[p]$, the instant such that $M_{\tilde{\mu}[p]} < M_{\mu[p]}$. We order places in S by $S = \{p_{k_1}, \ldots, p_{k_{|S|}}\}$ such that

$$1 \leq \mu[p_{k_1}] \leq \mu[p_{k_2}] \leq \cdots \leq \mu[p_{k_{|S|}}] \leq |\alpha| + \theta_i. \tag{17}$$

For each $t_i, 1 \leq i \leq |\alpha| + |\beta|$, we denote by $t_{i,1}$ (respectively, $t_{i,2}$) the first (respectively, the second) component of t_i. We choose $m = 0$ if $v_1 t_f \leq t_{1,2} \ldots t_{\mu[p_{k_1}],2}$, i.e., if t_f occurs no late than instant $\mu[p_{k_1}]$. Otherwise, we choose $m \in \{1, \ldots, |S|\}$ be the index s.t.

$$t_{1,2} \ldots t_{\mu[p_{k_m}],2} < v_1 t_f \leq t_{1,2} \ldots t_{\mu[p_{k_m}],2} \ldots t_{\mu[p_{k_{m+1}}],2}. \tag{18}$$

That is, $\mu[p_{k_m}]$ is the latest instant in $\mu[p_{k_1}], \ldots, \mu[p_{k_{|S|}}]$ before the occurrence of the fault transition t_f.

Now, let us consider the sequence in (19) shown at the bottom of this page, where $n_1, \ldots, n_{|S|}$ are some non-negative integers. We claim that this sequence is well-defined for some $n_1, \ldots, n_{|S|}$. To see this, we proceed as follows. To fire sequence $t_{\tilde{\mu}[p_{k_2}]+1} \ldots t_{\mu[p_{k_2}]}$ from $\tilde{M}_2$ for n_2 times, it suffices to have enough tokens in place p_{k_1}; this can be obtained by choosing n_1 to be sufficiently large, since $M_{\tilde{\mu}[1]} < M_{\mu[1]}$ implies that firing $t_{\tilde{\mu}[p_{k_1}]+1} \ldots t_{\mu[p_{k_1}]}$ strictly increases the number of tokens in place p_{k_1}. Analogously, to fire sequence $t_{\mu[p_{k_{|S|}}]+1} \ldots t_{|\alpha|+\theta_j}$ from $\tilde{M}'_{|S|}$ we just need to choose $n_{|S|}$ to sufficiently large, which can again be obtained by choosing $n_{|S|-1}$ to be sufficiently large and so forth.

Now, suppose that $n_1, \ldots, n_{|S|}$ are chosen such that the sequence in (19) is well-defined. Next, we show that it is indeed a sequence satisfying the formulas in Theorem IV.1, where each $(t_{\tilde{\mu}[p_{k_i}]+1} \ldots t_{\mu[p_{k_i}]})^{n_i}$ in (19) corresponds to σ_i in (6), each $t_{\mu[p_{k_i}]+1} \ldots t_{\mu[p_{k_{i+1}}]}$ in (19) corresponds to σ'_i in (6), and $t_{|\alpha|+\theta_i+1} \ldots t_{|\alpha|+\theta_j}$ in (19) corresponds to $\sigma_{|S|+1}$ in (6). We proceed by checking each formula.

Formula (7): Since $M_{|\alpha|+\theta_i} = M_{|\alpha|+\theta_j}$ and S is chosen such that $\forall P_\| \setminus S : M_{|\alpha|+\theta_i}(p) = M_{|\alpha|+\theta_j}(p) \neq \omega$, we know that sequence $t_{|\alpha|+\theta_i+1} \ldots t_{|\alpha|+\theta_j}$ does not cause token change for any place in $P_\| \setminus S$, i.e., $\forall p \in P_\| \setminus S : \tilde{M}_{|S|+1}(p) = \tilde{M}'_{|S|+1}(p)$. Therefore, this formula holds.

Formula (8): Recall that, for each $p_{k_i} \in S, \mu[p_{k_i}]$ is the instant when ω in place p_{k_i} is obtained. This implies the followings:

$$\begin{aligned}
&M_{0,\|} \xrightarrow{t_1 \ldots t_{\mu[p_{k_1}]}} \tilde{M}_1 \xrightarrow{\left(t_{\tilde{\mu}[p_{k_1}]+1} \ldots t_{\mu[p_{k_1}]}\right)^{n_1}} \tilde{M}'_1 \xrightarrow{t_{\mu[p_{k_1}]+1} \ldots t_{\mu[p_{k_2}]}} \cdots \\
&\xrightarrow{t_{\mu[p_{k_{m-1}}]} \ldots t_{\mu[p_{k_m}]}} \tilde{M}_m \xrightarrow{\left(t_{\tilde{\mu}[p_{k_m}]+1} \ldots t_{\mu[p_{k_m}]}\right)^{n_m}} \tilde{M}'_m \xrightarrow{t_{\mu[p_{k_m}]+1} \ldots t_{\mu[p_{k_{m+1}}]}} \cdots \\
&\xrightarrow{t_{\mu[p_{k_{|S|-1}}]} \ldots t_{\mu[p_{k_{|S|}}]}} \tilde{M}_{|S|} \xrightarrow{\left(t_{\tilde{\mu}[p_{k_{|S|}}]+1} \ldots t_{\mu[p_{k_{|S|}}]}\right)^{n_{|S|}}} \tilde{M}'_{|S|} \xrightarrow{t_{\mu[p_{k_{|S|}}]+1} \ldots t_{|\alpha|+\theta_i}} \tilde{M}_{|S|+1} \xrightarrow{t_{|\alpha|+\theta_i+1} \ldots t_{|\alpha|+\theta_j}} \tilde{M}'_{|S|+1}
\end{aligned} \tag{19}$$

1) $M_{\tilde{\mu}[p_{k_i}]}(p_{k_i}) < M_{\mu[p_{k_i}]}(p_{k_i})$; and
2) $\forall p \in P_\| \setminus \{p_{k_1}, \ldots, p_{k_i}\} : M_{\tilde{\mu}[p_{k_i}]}(p) \leq M_{\mu[p_{k_i}]}(p)$.

Note that we do not need to consider the token change for places $p_{k_1}, \ldots, p_{k_{i-1}}$, since ω has been obtained in these places. Therefore, sequence $(t_{\tilde{\mu}[p_{k_i}]+1} \ldots t_{\mu[p_{k_i}]})^{n_i}$ strictly increases the number of tokens in p_{k_i} and does not decrease the number of tokens in any place in $P_\| \setminus \{p_{k_1}, \ldots, p_{k_i}\}$. Therefore, this formula holds for the sequence in (19) by setting $\sigma_i = (t_{\tilde{\mu}[p_{k_i}]+1} \ldots t_{\mu[p_{k_i}]})^{n_i}$.

Formula (9): If $m = 0$, there is no need to consider this formula. When $m > 0$, recall that m is chosen such that the second component of sequence $t_1 t_2 \ldots t_{\mu[p_{k_m}]}$ is a prefix of $v_1 t_f$. Therefore, according to the construction of the modified coverability tree, we have $t_{\tilde{\mu}[p_{k_i}]+1} \ldots t_{\mu[p_{k_i}]} \in (T_N \times \{\lambda\})^*$ for any $i \leq m$. That is, for any $i \in \{1, \ldots, m\}$, $(t_{\tilde{\mu}[p_{k_i}]+1} \ldots t_{\mu[p_{k_i}]})^{n_i}$ does not contain a transition in $(T_N \cup \{\lambda\}) \times T$. Therefore, this formula holds.

Formula (10): If $m = 0$, then we know that $(\lambda, t_f) \in \{\lambda\} \times T_f$ must occur in $\sigma'_0 = t_1 \ldots t_{\mu[p_{k_1}]}$. If $m > 0$, by (18), we know that transition $(\lambda, t_f) \in \{\lambda\} \times T_f$ must occur in $\sigma'_m = t_{\mu[p_{k_m}]} \ldots t_{\mu[p_{k_{m+1}}]}$. Therefore, this formula also holds.

Formula (11): By our choice, $M_{|\alpha|+\theta_j}$ is reached immediately after a transitions in $(T_N \cup \{\lambda\}) \times T$, i.e., $t_{|\alpha|+\theta_j} \in (T_N \cup \{\lambda\}) \times T$. Therefore, this formula holds by choosing $\sigma_{|S|+1} = t_{|\alpha|+\theta_i+1} \ldots t_{|\alpha|+\theta_j}$.

Overall, all formulas in Theorem IV.1 hold for the sequence in (19).

Note that the above proof is based on the assumption that $S = \{p \in P_\| : M_{|\alpha|+\theta_i}(p) = \omega\} \neq \emptyset$. When $S = \emptyset$, we just need to replace the sequence in (19) by $M_{0,\|} \xrightarrow{t_1 \ldots t_{|\alpha|+\theta_i}} \tilde{M}_1 \xrightarrow{t_{|\alpha|+\theta_i+1} \ldots t_{|\alpha|+\theta_j}} \tilde{M}'_1$. We can show by the same arguments that all formulas in Theorem IV.1 still hold for this sequence. This completes the proof of necessity. ■

REFERENCES

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

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

[3] V. Srinivasan and M. Jafari, "Fault detection/monitoring using time Petri nets," *IEEE Trans. Syst. Man Cybern.*, vol. 23, no. 4, pp. 1155–1162, Jul./Aug. 1993.

[4] T. Ushio, I. Onishi, and K. Okuda, "Fault detection based on Petri net models with faulty behaviors," in *Proc. IEEE Int. Conf. Syst. Man Cybern.*, 1998, vol. 1, pp. 113–118.

[5] A. Benveniste, E. Fabre, S. Haar, and C. Jard, "Diagnosis of asynchronous discrete-event systems: a net unfolding approach," *IEEE Trans. Autom. Control*, vol. 48, no. 5, pp. 714–727, May 2003.

[6] Y. Wu and C. Hadjicostis, "Algebraic approaches for fault identification in discrete-event systems," *IEEE Trans. Autom. Control*, vol. 50, no. 12, pp. 2048–2055, Dec. 2005.

[7] D. Lefebvre and C. Delherm, "Diagnosis of DES with Petri net models," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, pp. 114–118, Jan. 2007.

[8] A. Ramírez-Treviño, E. Ruiz-Beltrán, I. Rivera-Rangel , and E. López-Mellado, "Online fault diagnosis of discrete event systems. A Petri net-based approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, pp. 31–39, Jan. 2007.

[9] S. Genc and S. Lafortune, "Distributed diagnosis of place-bordered Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 2, pp. 206–219, Apr. 2007.

[10] Y. Ru and C. Hadjicostis, "Fault diagnosis in discrete event systems modeled by partially observed Petri nets," *Discrete Event Dyn. Syst., Theory Appl.*, vol. 19, no. 4, pp. 551–575, 2009.

[11] F. Basile, P. Chiacchio, and G. De Tommasi , "An efficient approach for online diagnosis of discrete event systems," *IEEE Trans. Autom. Control*, vol. 54, no. 4, pp. 748–759, Apr. 2009.

[12] G. Jiroveanu and R. Boel, "The diagnosability of Petri net models using minimal explanations," *IEEE Trans. Autom. Control*, vol. 55, no. 7, pp. 1663–1668, Jul. 2010.

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

[14] M. Cabasino, A. Giua, and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," *Automatica*, vol. 46, no. 9, pp. 1531–1539, 2010.

[15] A. Madalinski, F. Nouioua, and P. Dague, "Diagnosability verification with Petri net unfoldings," *Int. J. Knowl.-Based Intell. Eng. Syst.*, vol. 14, no. 2, pp. 49–55, 2010.

[16] F. Basile, P. Chiacchio, and G. De Tommasi , "On K-diagnosability of Petri nets via integer linear programming," *Automatica*, vol. 48, no. 9, pp. 2047–2058, 2012.

[17] M. Cabasino, A. Giua, S. Lafortune, and C. Seatzu, "A new approach for diagnosability analysis of Petri nets using verifier nets," *IEEE Trans. Autom. Control*, vol. 57, no. 12, pp. 3104–3117, Dec. 2012.

[18] D. Lefebvre, "On-line fault diagnosis with partially observed Petri nets," *IEEE Trans. Autom. Control*, vol. 59, no. 7, pp. 1919–1924, Jul. 2014.

[19] M. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of discrete-event systems using labeled Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 144–153, Jan. 2014.

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

[21] H.-C. Yen, "A unified approach for deciding the existence of certain Petri net paths," *Inf. Comput.*, vol. 96, no. 1, pp. 119–137, 1992.

[22] S. Jiang, Z. Huang, V. Chandra, and R. Kumar, "A polynomial algorithm for testing diagnosability of discrete-event systems," *IEEE Trans. Autom. Control*, vol. 46, no. 8, pp. 1318–1321, Aug. 2001.

[23] 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, Aug. 2002.

[24] S. Tripakis, "Fault diagnosis for timed automata," in *Formal Techniques in Real-Time and Fault-Tolerant Systems*. New York, NY, USA: Springer-Verlag, 2002, pp. 205–221.

[25] C. Morvan and S. Pinchinat, "Diagnosability of pushdown systems," in *Hardware and Software: Verification Testing*. New York, NY, USA: Springer, 2011, pp. 21–33.

[26] K. Kobayashi and K. Hiraishi, "Verification of opacity and diagnosability for pushdown systems," *J. Appl. Math.*, vol. 2013, 2013, Art. no. 654059.

[27] C. Papadimitriou, *Computational Complexity*. Hoboken, NJ, USA: Wiley, 2003.

[28] T.-S. Yoo and H. E. Garcia, "Event counting of partially-observed discrete-event systems with uniformly and nonuniformly bounded diagnosis delays," *Discrete Event Dyn. Syst., Theory Appl.*, vol. 19, no. 2, pp. 167–187, 2009.

[29] G. Winskel, "Petri nets, algebras, morphisms, and compositionality," *Inf. Comput.*, vol. 72, no. 3, pp. 197–238, 1987.

[30] R. Karp and R. Miller, "Parallel program schemata," *J. Comput. Syst. Sci.*, vol. 3, no. 2, pp. 147–195, 1969.

[31] C. Rackoff, "The covering and boundedness problems for vector addition systems," *Theor. Comput. Sci.*, vol. 6, no. 2, pp. 223–231, 1978.

[32] S. German and A. Sistla, "Reasoning about systems with many processes," *J. ACM*, vol. 39, no. 3, pp. 675–735, 1992.

[33] J. Esparza, "Decidability and complexity of Petri net problems: An introduction," in *Lectures on Petri Nets I: Basic Models*. New York, Ny, USA: Springer, 1998, pp. 374–428.

[34] R. Lipton, "The reachability problem requires exponential space," *Res. Report 62, Dept. Comput. Sci., Yale Univ.*, New Haven, CT, USA, 1976.

[35] N. Jones, L. Landweber, and Y. Lien, "Complexity of some problems in Petri nets," *Theor. Comput. Sci.*, vol. 4, no. 3, pp. 277–299, 1977.

[36] J. Peterson, *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1981.

[37] L. Dickson, "Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors," *Amer. J. Math.*, vol. 35, no. 4, pp. 413–422, 1913.