This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2017.2756096, IEEE Transactions on Automatic Control

Verification of Prognosability for Labeled Petri Nets

Xiang Yin, *Member, IEEE*

***Abstract*—This technical note is concerned with the fault prognosis problem for partially-observed discrete-event systems modeled by unbounded labeled Petri nets. The goal of this problem is to predict the occurrence of each fault before its occurrence. The condition of prognosability provides the necessary and sufficient condition under which any fault can be predicted with no missed detection and no false alarm. In this technical note, we investigate the verification of prognosability for unbounded labeled Petri nets. First, we show that checking prognosability is decidable for Petri net languages. Our approach is based on a reduction from this verification problem to an existing Petri nets model checking problem. Then we show that the complexity of this problem is EXPSPACE-complete. Our results extend previous works on the verification of language-based prognosability from regular languages to Petri net languages.**

***Index Terms*—Discrete Event Systems, Petri Nets, Fault Prognosis, Computational Complexity**

I. Introduction

Fault prognosis is an important task in many safety-critical cyber-physical systems. In this problem, we want to predict the occurrences of faults and to generate corresponding fault alarms in order to protect the system. In this technical note, we are concerned with the problem of fault prognosis of discrete-event systems (DES) [9].

In the context of DES, model-based fault prognosis was initially studied in [15], [16], where a language-based condition called prognosability (or predictability) was proposed. Specifically, prognosability is proposed to determine *a priori* whether or not a fault prognoser can be designed such that: (i) no false alarm, i.e., a fault is guaranteed to occur within a finite number of steps whenever a fault alarm is generated; and (ii) no missed detection, i.e., any fault will be alarmed before its occurrence. Since then, fault prognosis of DES has drawn consideration attention in the DES literature; see, e.g., [6], [12], [18], [19], [24], [28], [29], [31], [32], [34], [36], [37]. For example, the notion of prognosability has been extended to decentralized systems, where the notion of co-prognosability was proposed [18], [19], [36], [37]. The fault prognosis problem has also been studied in the distributed setting [29], [31], [32]. In [34], the authors investigated the enforcement of prognosability by sensor activation. The robust fault prognosis problem was studied in [28]. Finally, prognosability analysis has also been studied in timed systems [11] and stochastic systems [6], [12], [24].

Most of the existing works on fault prognosis of DES are based on finite-state automata models. In many concurrent systems, however, Petri nets provide a more compact and natural way for modeling DES without explicitly enumerating the entire state-space. Moreover, it is well-known that Petri net languages are strictly more expressive than regular languages, languages generated by finite-state automata. Therefore, Petri nets can model some infinite-state systems that cannot be represented by finite-state automata, e.g, manufacturing systems with infinite buffers. Due to these advantages, in the context of Petri nets, many works have been done on the fault diagnosis problem, a problem related to the fault prognosis problem; see, e.g., [3]–[5], [7], [8], [14], [17], [21], [23], [26], [27]. Recently, there have been works on fault prognosis based on Petri nets [1], [20], where procedures for online prognosis were provided.

In this technical note, we investigate the verification of prognosability in unbounded labeled Petri nets. Specifically, we follow the language-based definition of prognosability in [15], [19] to determine a priori whether or not a fault can be predicted with no missed detection and with no false alarm. The main contributions of this technical note are as follows. First, we show that prognosability is decidable for labeled Petri nets by effectively reducing the prognosability verification to a model checking problem for Petri nets. In the context of unbounded Petri nets, several (un)decidability results have been established for related notions. For example, it has been shown that the verification of diagnosability is decidable [35], while the verification of opacity is shown to be undecidable [30]. To the best of our knowledge, the decidability status of prognosability is still open and our result provides positive answer to this question. Second, we establish the precise computational complexity for the prognosability verification problem. Specifically, we show that checking prognosability for unbounded Petri nets is EXPSPACE-complete, i.e., exponential memory is required for this verification problem.

X. Yin is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China. E-mail: xiangyin@umich.edu.

II. Preliminaries

A. Petri Nets

A place/transition *net* is defined as a 4-tuple $\mathcal{N} = (P, T, A, w)$, where $P = \{p_1, p_2, \ldots, p_n\}$ is the set of n places, $T = \{t_1, t_2, \ldots, t_m\}$ is the set of m transitions, $A \subseteq (P \times T) \cup (T \times P)$ is the set of arcs, and $w : A \to \mathbb{N}$ is the weight function that assigns to each arc a non-negative integer. For any place $p \in P$, we denote by ${}^\bullet p$ its preset, i.e., ${}^\bullet p = \{t \in T : (t,p) \in A\}$; we denote by $p^\bullet$ its postset, i.e., $p^\bullet = \{t \in T : (p,t) \in A\}$. For a transition $t \in T$, its preset ${}^\bullet t$ and its postset $t^\bullet$ are defined analogously, which are sets of places. Given a net $\mathcal{N}$, a marking M is a vector $M = [M(p_1)\ M(p_2)\ \ldots\ M(p_n)]^\top \in \mathbb{N}^n$, where $M(p)$ is the number of *tokens* in place $p \in P$. A *Petri net* is a 2-tuple $\langle \mathcal{N}, M_0 \rangle$, where $\mathcal{N}$ is a net and $M_0 \in \mathbb{N}^n$ is the initial marking. We say that transition $t \in T$ is *enabled* at marking M if $\forall p \in {}^\bullet t : M(p) \geq w(p,t)$. If t is enabled, then it may *fire* and yield a new marking determined by $M' = M - w(\cdot, t) + w(t, \cdot)$. We use $M \xrightarrow{t}_{\mathcal{N}}$ to denote that transition $t \in T$ is enabled at M in net $\mathcal{N}$ and $M \xrightarrow{t}_{\mathcal{N}} M'$ means that firing t yields M' in net $\mathcal{N}$. Hereafter, we will also omit the subscript $\mathcal{N}$ when it is clear from the context.

Let T^* be the set of all finite sequences of transitions including the empty transition λ, which means that no transition is fired, and, for any $\sigma \in T^*$, we have $\sigma\lambda = \lambda\sigma = \sigma$. We say that a sequence of transitions (or, for simplicity, a sequence) $\sigma = t_1 t_2 \ldots t_k \in T^*$ is enabled at M if $\forall i \in \{1, \ldots, k\} : M_i \xrightarrow{t_i}$, where $M_1 = M$ and $M_i \xrightarrow{t_i} M_{i+1}, \forall i \geq 1$. 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 $\langle \mathcal{N}, M_0 \rangle$, $L(\mathcal{N}, M_0)$ denotes the set of finite sequences that can be fired from M_0, i.e., $L(\mathcal{N}, M_0) = \{\sigma \in T^* : M_0 \xrightarrow{\sigma}\}$. For any sequence $\sigma \in T^*$, we denote by $\overline{\sigma}$ the set of prefixes of σ, i.e., $\overline{\sigma} = \{\sigma_1 \in T^* : \exists \sigma_2 \in T^* \text{ s.t. } \sigma_1\sigma_2 = \sigma\}$. Finally, we denote by $|\sigma|$ the length of sequence σ.

Let Σ be a finite set of alphabets (or events). A string is a finite sequence of events and we denote by Σ^* the set of all strings including the empty string ϵ. 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 $\mathcal{L} : T \to \Sigma \cup \{\epsilon\}$ is a labeling function. That is, for any $t \in T$, $\mathcal{L}(t)$ specifies the event that can be observed when t fires. For any transition $t \in T$, if $\mathcal{L}(t) \in \Sigma$, then we say that transition t is *observable*; otherwise, t is *unobservable*. Therefore, T is partitioned as $T = T_o \dot{\cup} T_{uo}$, where T_o and T_{uo} are the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2017.2756096, IEEE Transactions on Automatic Control

set of observable transitions and the set of unobservable transitions, respectively. Function $\mathcal{L}$ is also extended from T to T^* recursively by: (i) $\mathcal{L}(\lambda) = \epsilon$; and (ii) $\forall \sigma \in T^*, t \in T : \mathcal{L}(\sigma t) = \mathcal{L}(\sigma)\mathcal{L}(t)$. Then the language generated by labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is a set of strings $\mathcal{L}(L(\mathcal{N}, M_0)) := \{\mathcal{L}(\sigma) : \sigma \in L(\mathcal{N}, M_0)\}$.

B. Yen's Problem

In this paper, we will leverage an existing path logic model checking problem for unbounded Petri nets in the literature originally studied by Yen [33]. For any sequence $\sigma \in T^*$ and transition $t \in T$, We denote by $\#_\sigma(t)$ the number of times t occurs in σ. Then Yen's problem is formulated as follows.

Definition II.1. *(Yen's Problem).* Given a Petri net $\langle \mathcal{N}, M_0 \rangle$, decide whether or not there exists a sequence

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

such that a predicate $F(M_1, \dots, M_k, \sigma_1, \dots, \sigma_k)$ holds, where $F(M_1, \dots, M_k, \sigma_1, \dots, \sigma_k)$ is a predicate obtained from the following syntax:

(i) The followings are predicates: $M_i(p) \geq c$, $M_i(p) \leq M_j(p')$, $\#_{\sigma_i}(t) \leq c$, $\#_{\sigma_i}(t) \geq c$ and $\#_{\sigma_i}(t) \leq \#_{\sigma_j}(t')$, where c is an arbitrary constant.

(ii) For any predicates F_1 and F_2, $F_1 \wedge F_2$ and $F_1 \vee F_2$ are also predicates.

In general, Yen's problem is decidable and it is as hard as the reachability problem[1]. Furthermore, it has been shown in [2] that, when the predicate satisfies the constraint that $F(M_1, \dots, M_k, \sigma_1, \dots, \sigma_k) \Rightarrow M_1 \leq M_k$, this problem can be solved in EXPSPACE. Hereafter, we will only use the fact that this problem is decidable and the restricted case can be solved with exponential space in the size of $\mathcal{N}$ and the size of the predicate. Details on how to solve this problem can be found in [2], [33].

III. Prognosability of Labeled Petri Nets

In the fault prognosis problem, we assume that the set of transitions is partitioned into two disjoint sets $T = T_F \dot{\cup} T_N$, where T_F denotes the set of fault transitions and T_N denotes the set of non-fault transitions. For any sequence $\sigma = t_1 t_2 \dots t_k \in T^*$, with a slight abuse of notation,we write that $T_F \in \sigma$ if a fault transition occurs in σ, i.e., $\exists i \in \{1, \dots, k\} : t_i \in T_F$.

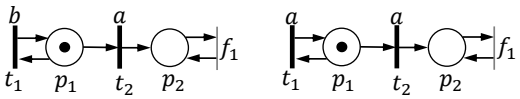
As we mentioned earlier, the main purpose of the fault prognosis problem is to *predict* any fault correctly *before* its occurrence, where "correctly" means that

(i) any fault should be alarmed before it occurs, i.e., no missed detection; and

(ii) once a fault alarm is generated, a fault is guaranteed to occur within a finite number of steps, i.e., no false alarm.

In [15], [19], the notion of prognosability (or predictability) was proposed as the necessary and sufficient condition under which there exists a prognosis mechanism such that the above two requirements can be achieved. Although [15], [19] only study the verification of prognosability for regular languages, the definition of prognosability itself is applicable to any class of languages. Here, we present the definition of prognosability for Petri net languages.

[1]In the original paper [33], it is claimed that the general case is in EXPSPACE, which is not correct as pointed out by [2].



(a) A prognosable system. (b) A non-prognosable system.

Fig. 1: Examples of prognosability, where for each system, t_f is the unique fault transition. Bold lines are used to denote observable transitions and the event associated with each observable transition denotes its observation label.

Definition III.1. (Prognosability). 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 prognosable w.r.t. T_F if

$$(\forall \alpha \in L(\mathcal{N}, M_0) : T_F \in \alpha)(\exists \beta \in \overline{\alpha} : T_F \notin \beta)$$
$$(\forall \theta \in L(\mathcal{N}, M_0) : \mathcal{L}(\theta) = \mathcal{L}(\beta) \wedge T_F \notin \theta)$$
$$(\exists K \in \mathbb{N})(\forall \theta\gamma \in L(\mathcal{N}, M_0))[|\gamma| \geq K \Rightarrow T_F \in \gamma]$$

Intuitively, prognosability can be used to determine *a priori* if any fault occurrence in the system can be correctly predicted. More specifically, it requires that, for any fault sequence, it must has a non-fault prefix for which we know for sure that a fault is guaranteed to occur within a finite number of steps, i.e., a fault alarm can be correctly issued. In other words, if the system is not prognosable, then it implies that there must exist a fault sequence for which we cannot claim that the fault will occur unambiguously along its non-fault prefixes. Therefore, any fault prognosis mechanism cannot correctly predict this fault before it occurs.

Remark III.1. Note that here we do not assume that $T_F \subseteq T_{uo}$, which is the (non-trivial) case for the fault diagnosis problem. This is because that, in the fault prognosis problem, we are mainly interested in the behavior of the system before the occurrences of faults. Therefore, even if a fault transition is observable and can be distinguished from other non-fault transitions, it is still possible that a fault alarm cannot be issued unambiguously before it occurs.

We illustrate the notion of prognosability in Petri nets by the following examples.

Example III.1. Let us consider labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ shown in Figure 1(a), where $T_o = \{t_1, t_2\}$ and $T_F = \{f_1\}$. Also, let $\Sigma = \{a, b\}, \mathcal{L}(t_1) = b$ and $\mathcal{L}(t_2) = a$. This system is prognosable, since transition t_2 has to occur before the occurrence of fault transition f_1, and once t_2 occurs, the token in place p_1 will be consumed, i.e., the only transition can occur next is the fault transition f_1. Therefore, once we observe event $\mathcal{L}(t_2) = a$, which can only be generated by transition t_2 in this example, we can claim unambiguously that the fault will occur within one step.

Example III.2. Let us consider labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ shown in Figure 1(b), where we have $T_o = \{t_1, t_2\}, T_F = \{f_1\}$ and $\Sigma = \{a\}$. We consider a labeling function defined by $\mathcal{L}(t_1) = \mathcal{L}(t_2) = a$. This system is not prognosable. To see this, let us consider fault sequence $t_2 f_1 \in L(\mathcal{N}, M_0)$. Then for $t_2 \in \overline{t_2 f_1} : T_F \notin t_2$, we can find $t_1 \in L(\mathcal{N}, M_0)$ such that $\mathcal{L}(t_1) = \mathcal{L}(t_2) = a$ and for any $K \in \mathbb{N}$, a non-fault sequence $t_1 (t_1)^K$ is defined in $\langle \mathcal{N}, M_0 \rangle$. Intuitively, the non-prognosability here can also be explained as follows. To avoid missed detection, we have to issue a fault alarm upon the occurrence of t_2, i.e., by observing event a. However, sequences t_1 and t_2 are indistinguishable and an arbitrarily long non-fault behavior can still occur after t_1. Therefore, this fault alarm cannot guarantee a fault to occur within a finite number of steps, i.e., it may be a false alarm.

Next, we will provide a characterization of prognosability for labeled Petri nets. First, motivated by relevant notions in [19] for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2017.2756096, IEEE Transactions on Automatic Control

finite-state automata, we introduce the notions of *boundary marking* and *non-indicator marking*.

Definition III.2. A marking $M \in \mathbb{N}^n$ is said to be

- a *Boundary Marking* if $(\exists t_f \in T_F)[M \xrightarrow{t_f}]$; and
- a *Non-Indicator Marking* if $(\forall K \in \mathbb{N})(\exists \sigma \in T_N^*)[M \xrightarrow{\sigma} \wedge |\sigma| \geq K]$.

Intuitively, a boundary marking is a marking from which a fault transition can occur immediately and a non-indicator marking is a marking from which an arbitrarily long non-fault sequence can occur. Note that, since vectors of integers form a well quasi-ordering [13], for any M_1, there does not exists an infinite sequence of vectors $M_1, M_2, M_3, \ldots$ such that $M_i \lneq M_j$ for any $i < j$. Therefore, M is a non-indicator marking if and only if $(\exists \sigma, \sigma' \in T_N^*)[M \xrightarrow{\sigma} M' \xrightarrow{\sigma'} M'' \wedge M' \leq M'']$.

The following result provides a characterization of prognosability in terms of boundary markings and non-indicator markings.

Lemma III.1. Labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is not prognosable w.r.t. T_F, if and only if, there exist two non-fault sequences $\sigma_1, \sigma_2 \in T_N^*$ such that

(i) M_1 is a non-indicator marking, where $M_0 \xrightarrow{\sigma_1} M_1$; and
(ii) M_2 is a boundary marking, where $M_0 \xrightarrow{\sigma_2} M_2$; and
(iii) $\mathcal{L}(\sigma_1) = \mathcal{L}(\sigma_2)$.

Proof. ($\Leftarrow$) Suppose that there exist two non-fault sequences $\sigma_1, \sigma_2 \in T_N^*$ such that the above conditions hold. Since $M_0 \xrightarrow{\sigma_2} M_2$ and M_2 is a boundary marking, we know that there exists $t_f \in T_F$ such that $M_0 \xrightarrow{\sigma_2 t_f}$. Then, for any non-fault prefix of $\sigma_2 t_f$, say $\beta \in \overline{\sigma_2}$, since $\mathcal{L}(\sigma_1) = \mathcal{L}(\sigma_2)$, we know that there exists a prefix of σ_1, say $\theta \in \overline{\sigma_1}$ such that $\mathcal{L}(\theta) = \mathcal{L}(\beta)$. Since $M_0 \xrightarrow{\theta(\sigma_1/\theta)} M_1$ and M_1 is a non-indicator marking, where (σ_1/θ) is the sequence such that $\theta(\sigma_1/\theta) = \sigma_1$, we know that $(\forall K \in \mathbb{N})(\exists \sigma \in T_N^*)[M_0 \xrightarrow{\theta(\sigma_1/\theta)\sigma} \wedge |\sigma| \geq K]$. Overall, we have

$$\begin{aligned} &(\exists \sigma_2 t_f \in L(\mathcal{N}, M_0) : T_F \in \sigma_2 t_f)(\forall \beta \in \overline{\sigma_2} : T_F \notin \beta) \\ &(\exists \theta \in L(\mathcal{N}, M_0) : \mathcal{L}(\theta) = \mathcal{L}(\beta) \wedge T_F \notin \theta) \\ &(\forall K \in \mathbb{N})(\exists \theta(\sigma_1/\theta)\sigma \in L(\mathcal{N}, M_0))[|\gamma| \geq K \wedge T_F \notin \gamma] \end{aligned} \tag{2}$$

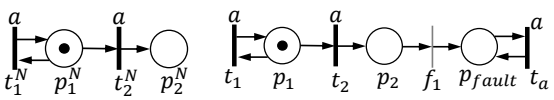
where $\gamma = (\sigma_1/\theta)\sigma$. That is, the system is not prognosable.

($\Rightarrow$) Suppose that the system is not prognosable, i.e.,

$$\begin{aligned} &(\exists \alpha \in L(\mathcal{N}, M_0) : T_F \in \alpha)(\forall \beta \in \overline{\alpha} : T_F \notin \beta) \\ &(\exists \theta \in L(\mathcal{N}, M_0) : \mathcal{L}(\theta) = \mathcal{L}(\beta) \wedge T_F \notin \theta) \\ &(\forall K \in \mathbb{N})(\exists \theta\gamma \in L(\mathcal{N}, M_0))[|\gamma| \geq K \wedge T_F \notin \gamma] \end{aligned} \tag{3}$$

Let α be a string satisfying Equation (3). We take β as the longest prefix of α such that $T_F \notin \beta$, i.e., $\beta t_f \in \overline{\alpha}$ for some $t_f \in T_F$. We know that M_2 is a boundary marking, where $M_0 \xrightarrow{\beta} M_2$. Let θ be a non-faulty sequence such that $\mathcal{L}(\theta) = \mathcal{L}(\beta)$ and $(\forall K \in \mathbb{N})(\exists \theta\gamma \in L(\mathcal{N}, M_0))[|\gamma| \geq K \wedge T_F \notin \gamma]$. By definition, we know that M_1 is a non-indicator marking, where $M_0 \xrightarrow{\theta} M_1$. Therefore, by taking $\sigma_1 = \theta$ and $\sigma_2 = \beta$, all conditions in the lemma hold. □

Finally, we denote by $\langle \mathcal{N}_N, M_0, \mathcal{L} \rangle$ the labeled Petri net obtained by removing transitions in T_F from $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$. Specifically, $\mathcal{N}_N = (P_N, T_N, A_N, w_N)$, where $P_N = P$, A_N is obtained by restricting A to domain $(P \times T_N) \cup (T_N \times P)$ and w_N is obtained by restricting w to domain A_N. This net is also referred to as the *normal net* hereafter. For example, for labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ shown in Figure 1(b), its normal net $\langle \mathcal{N}_N, M_0, \mathcal{L} \rangle$ is shown in Figure 2(a). For the sake of clarity, we add superscript N for each transition and each place in the normal net in order to distinguish them from transitions and places in the original net.



(a) Petri net $\langle \mathcal{N}_N, M_0, \mathcal{L} \rangle$ (b) Petri net $\langle \tilde{\mathcal{N}}, \tilde{M}_0 \rangle$

Fig. 2: Petr nets $\langle \mathcal{N}_N, M_0 \rangle$ and $\langle \tilde{\mathcal{N}}, \tilde{M}_0, \tilde{\mathcal{L}} \rangle$ for the Petri net shown in Figure 1(b).

IV. Verification of Prognosability

In this section, we first provide a necessary and sufficient condition for prognosability in terms of a formula satisfying the syntax in the Yen's problem. Then we show that the verification of prognosability is decidable and it is in EXPSPACE.

By Lemma III.1, to verify prognosability, it suffices to verify the existence of two observationally equivalent sequences such that one goes to a boundary marking and the other goes to a non-indicator marking. Similarly result can also be found in the fault diagnosis problem; see, e.g., [7], [10]. The basic idea to verify this is to use a twin-plant-like approach in order to track all pairs of sequences that look the same. However, to implement this idea, the following difficulty arises. For boundary markings, it is straightforward to obtain a closed-form representation; however, obtaining such a closed-form representation for non-indicator markings seems to be difficult. To resolve this technique challenge, we first define a new net $\langle \tilde{\mathcal{N}}, M_0, \tilde{\mathcal{L}} \rangle$ and then use this net together with the normal net for the twin-plant construction.

Let $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ be the labeled Petri net under consideration. We define a new labeled Petri net $\langle \tilde{\mathcal{N}}, \tilde{M}_0, \tilde{\mathcal{L}} \rangle$, where $\tilde{\mathcal{N}} = (\tilde{P}, \tilde{T}, \tilde{A}, \tilde{w})$, as follows:

- $\tilde{P} = P \cup \{p_{fault}\}$, where p_{fault} is a new place;
- $\tilde{T} = T \cup \{t_e : e \in \Sigma\}$, where each t_e is a new transition;
- $\tilde{A}$ and $\tilde{w}$ are defined by:
  - For any $t \in T_N$, ${}^\bullet t$ and $t^\bullet$ are the same in $\mathcal{N}$ and $\forall p \in P$: $\tilde{w}(p,t) = w(p,t), \tilde{w}(t,p) = w(t,p)$.
  - For any $t \in T_F$, ${}^\bullet t$ is the same in $\mathcal{N}$ and $\forall p \in P : \tilde{w}(p,t) = w(p,t)$, while $t^\bullet = \{p_{fault}\}$ with $\tilde{w}(t, p_{fault}) = 1$.
  - For any $t_e, e \in \Sigma$, we have ${}^\bullet t_e = t_e^\bullet = \{p_{fault}\}$ and $\tilde{w}(t_e, p_{fault}) = \tilde{w}(p_{fault}, t_e) = 1$.

The initial marking is $\tilde{M}_0 = [M_0^\top \ 0]^\top$ (we assume the last place is p_{fault}). The labeling function $\tilde{\mathcal{L}} : \tilde{T} \to \Sigma \cup \{\epsilon\}$ is defined by

$$\tilde{\mathcal{L}}(t) = \begin{cases} \mathcal{L}(t) & \text{if } t \in T_N \\ \epsilon & \text{if } t \in T_f \\ e & \text{if } t = t_e \end{cases} \tag{4}$$

Intuitively, for any non-fault transition, the dynamic of $\tilde{\mathcal{N}}$ is consistent with $\mathcal{N}$. However, for any fault transition, $\tilde{\mathcal{N}}$ will send a token to a new place p_{fault}, which denotes the occurrence of fault. For each event $e \in \Sigma$, a self-loop transition t_e labeled with e is defined at p_{fault}. For example, let us still consider labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ shown in Figure 1(b). Then its corresponding net $\langle \tilde{\mathcal{N}}, \tilde{M}_0, \tilde{\mathcal{L}} \rangle$ is shown in Figure 2(b).

Next, we define a new (unlabeled) Petri net $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$ that "synchronizes" $\langle \mathcal{N}_N, M_0, \mathcal{L} \rangle$ and $\langle \tilde{\mathcal{N}}, \tilde{M}_0, \tilde{\mathcal{L}} \rangle$ based on their labeling functions. Specifically, $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$, where $\mathcal{N}_\parallel = (P_\parallel, T_\parallel, A_\parallel, w_\parallel)$, is defined as follows:

- $P_\parallel = P_N \cup \tilde{P}$;
- $T_\parallel \subseteq (T_N \cup \{\lambda\}) \times (\tilde{T} \cup \{\lambda\}) \setminus \{(\lambda, \lambda)\}$;
- $A_\parallel$ and $w_\parallel$ are defined by

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2017.2756096, IEEE Transactions on Automatic Control

– For any $t_1 \in T_N$ and $t_2 \in \tilde{T}$ such that $\mathcal{L}(t_1) = \tilde{\mathcal{L}}(t_2) \in \Sigma$, we have that $(t_1, t_2) \in T_\parallel$ 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,

$$w_\parallel((t_1, t_2), p) = \begin{cases} w_N(t_1, p) & \text{if } p \in P_N \\ \tilde{w}(t_2, p) & \text{if } p \in \tilde{P} \end{cases} \quad (5)$$

$$w_\parallel(p, (t_1, t_2)) = \begin{cases} w_N(p, t_1) & \text{if } p \in P_N \\ \tilde{w}(p, t_2) & \text{if } p \in \tilde{P} \end{cases} \quad (6)$$

– For any $t_1 \in T_N$ such that $\mathcal{L}(t_1) = \epsilon$, we have $(t_1, \lambda) \in T_\parallel$ with ${}^\bullet(t_1, \lambda) = {}^\bullet t_1$ and $(t_1, \lambda)^\bullet = t_1^\bullet$. Then for any $p \in P_N$, $w_\parallel((t_1, \lambda), p) = w_N(t_1, p)$ and $w_\parallel(p, (t_1, \lambda)) = w_N(p, t_1)$.

– For any $t_2 \in \tilde{T}$ such that $\tilde{\mathcal{L}}(t_2) = \epsilon$, we have $(\lambda, t_2) \in T_\parallel$ with ${}^\bullet(\lambda, t_2) = {}^\bullet t_2$ and $(\lambda, t_2)^\bullet = t_2^\bullet$. Then for any $p \in \tilde{P}$, $w_\parallel((\lambda, t_2), p) = \tilde{w}(t_2, p)$ and $w_\parallel(p, (\lambda, t_2)) = \tilde{w}(p, t_2)$.

- $M_{0,\parallel} = \begin{bmatrix} M_0^\top & \tilde{M}_0^\top \end{bmatrix}^\top$.

Remark IV.1. The construction of $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$ follows the idea of twin-plant (or verifier net) that is used in the literature for the verification of diagnosability; see, e.g., [7], [23], [35]. However, the difference here is that we need to modified one net before the construction in order capture the feature of the fault prognosis problem. Intuitively, $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$ tracks and only tracks all pairs of two sequences, one in $\mathcal{N}_N$ and the other one in $\tilde{\mathcal{N}}$, that have the same observation. For any transition $(t_1, t_2) \in T_\parallel$, if $t_i = \lambda$ for some i, then it means that its corresponding net stays silently when the other net fires an unobservable transition; if $t_1, t_2 \neq \lambda$, then it means that two nets are moved simultaneously by firing observable transitions with a same label. Then for any sequence $\sigma \in L(\mathcal{N}_\parallel, M_{0,\parallel})$, we denote by σ_1 and σ_2 its first and second components, respectively. Then we know that $\mathcal{L}(\sigma_1) = \tilde{\mathcal{L}}(\sigma_2)$. Similarly, for any two sequences $\sigma_1 \in L(\mathcal{N}_N, M_0)$ and $\sigma_2 \in L(\tilde{\mathcal{N}}, \tilde{M}_0)$, such that $\mathcal{L}(\sigma_1) = \tilde{\mathcal{L}}(\sigma_2)$, then there exists a sequence $\sigma \in L(\mathcal{N}_\parallel, M_{0,\parallel})$ such that its first and second components are σ_1 and σ_2, respectively.

Based on net $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$, we are now ready to present a necessary and sufficient condition for prognosability.

Theorem IV.1. Labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is not prognosable w.r.t. T_F, if and only if, there exists a sequence

$$M_{0,\parallel} \xrightarrow{\alpha}_{\mathcal{N}_\parallel} M_1 \xrightarrow{\beta}_{\mathcal{N}_\parallel} M_2 \quad (7)$$

in $\langle \mathcal{N}_\parallel = (P_\parallel, T_\parallel, A_\parallel, w_\parallel), M_{0,\parallel} \rangle$, such that

$$(M_2 \geq M_1) \wedge \Big(\bigvee_{t \in \{\lambda\} \times T_F} \#_\alpha(t) \geq 1 \Big) \wedge \Big(\bigvee_{t \in T_N \times (\tilde{T} \cup \{\lambda\})} \#_\beta(t) \geq 1 \Big) \quad (8)$$

Remark IV.2. Before we formally prove the above theorem, let us first explain intuitively how it works. For a sequence in Equation (7), since $\bigvee_{t \in \{\lambda\} \times T_F} \#_\alpha(t) \geq 1$, we know that a boundary marking can be reached by a prefix of the second component of α. Moreover, the last condition guarantees that β is non-λ for its first component. This condition together with $M_2 \geq M_1$ ensure that any marking reached by a prefix of the first component of α is a non-indicator marking. Therefore, the conditions in Equation (8) essentially guarantee that there are two observationally equivalent sequences that can reach a non-indicator marking and a boundary marking, respectively, which disproves prognosability. On the other hand, suppose that the system is not prognosable, i.e., there exist a non-indicator marking $M'_{1,1}$ and a boundary marking $M'_{1,2}$ that can be reached by α_1 and α_2, respectively, such that α_1 and α_2 look the same. Then we know that $M_{0,\parallel} \xrightarrow{\alpha}_{\mathcal{N}_\parallel} M'_1 = \begin{bmatrix} M'^\top_{1,1} & M'^\top_{1,2} \end{bmatrix}^\top$ for some α whose first and second components are α_1 and α_2, respetively. Since $M'_{1,2}$ is a boundary marking and any fault transition is unobservable in $\tilde{\mathcal{L}}$, we know that $M'_1 \xrightarrow{(\lambda, t_f)}_{\mathcal{N}_\parallel} \begin{bmatrix} M'^\top_{1,1} & M'^\top_{1,2} \end{bmatrix}^\top$ for some $t_f \in T_F$. Since $M'_{1,1} = M_{1,1}$ is a non-indicator marking, we know that β_1 can be extended from $M_{1,1}$ to obtain a covering for places in P_N. Moreover, $M_{1,2}$ contains a token in p_{fault}. Therefore, the self-loops in the form of t_e can "track" the sequence, which contributes to the covering in $\mathcal{N}_N$, without changing markings in $\tilde{P}$. This yields a covering for all places in $P_\parallel$ that satisfies Equation (8). This is also the reason why we add such self-loop transitions at place p_{fault} in $\tilde{\mathcal{N}}$.

With the above explained intuition, we are now ready to formally prove Theorem IV.1.

Proof. ($\Leftarrow$) Let $M_{0,\parallel} \xrightarrow{\alpha}_{\mathcal{N}_\parallel} M_1 \xrightarrow{\beta}_{\mathcal{N}_\parallel} M_2$ be a sequence satisfying the conditions in Equation (8). We denote by α_1 and α_2 the first and the second components of α, respectively. The same for notations β_1 and β_2.

Let us consider the longest non-fault prefix of α_2, say $\alpha'_2 \in \overline{\alpha_2}$, i.e., $\alpha'_2 \in T_N^*$ and $\alpha'_2 t_f \in L(\mathcal{N}, M_0)$ for some $t_f \in T_F$. This means that $M'_{1,2}$ is a boundary marking, where $M_0 \xrightarrow{\alpha'_2}_{\mathcal{N}} M'_{1,2}$. By the construction of $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$, we know that $\mathcal{L}(\alpha_1) = \tilde{\mathcal{L}}(\alpha_2)$. Moreover, since $\alpha'_2 \in T_N^*$, we know that $\tilde{\mathcal{L}}(\alpha'_2) = \mathcal{L}(\alpha'_2)$. Therefore, $\mathcal{L}(\alpha_1) = \mathcal{L}(\alpha_2)$. Then for α'_2, there exists a non-fault sequence $\alpha'_1 \in \overline{\alpha_1}$ such that $\mathcal{L}(\alpha'_1) = \mathcal{L}(\alpha'_2)$. Moreover, since $M_1 \leq M_2$, we know that $M_{1,1} \leq M_{2,1}$, where $M_0 \xrightarrow{\alpha'_1}_{\mathcal{N}_N} M'_{1,1} \xrightarrow{(\alpha/\alpha'_1)}_{\mathcal{N}_N} M_{1,1} \xrightarrow{\beta_1}_{\mathcal{N}_N} M_{2,1}$. Therefore, $M'_{1,1}$ is a non-indicator marking. By Lemma III.1, we know that the system is not prognosable.

($\Rightarrow$) Suppose that system is not prognosable. By Lemma III.1, we know that there exist two non-fault sequences $\sigma_1, \sigma_2 \in T_N$ such that (i) $M_0 \xrightarrow{\sigma_1} M_1$, M_1 is a non-indicator marking; and (ii) $M_0 \xrightarrow{\sigma_2} M_2$, M_2 is a boundary marking; and (iii) $\mathcal{L}(\sigma_1) = \mathcal{L}(\sigma_2)$.

First, since σ_1 only contains non-fault transitions, by the definition of $\tilde{\mathcal{N}}$, we know that $\sigma_1 \in L(\tilde{\mathcal{N}}, \tilde{M}_0)$. Moreover, by the definition of $\tilde{\mathcal{L}}$, we know that $\mathcal{L}(\sigma_1) = \tilde{\mathcal{L}}(\sigma_2)$. Therefore, by the property of $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$, we know that there exists a sequence $\alpha \in L(\mathcal{N}_\parallel, M_{0,\parallel})$ such that its first component is σ_1 and its second component is σ_2. Moreover, since M_2 is a boundary marking, we know that $\sigma_2 t_f \in L(\mathcal{N}, M_0)$ for some $t_f \in T_F$. By the definition of $\tilde{\mathcal{L}}$, we know that $\tilde{\mathcal{L}}(t_f) = \epsilon$. Therefore, by the definition of $\mathcal{N}_\parallel$, we know that $\alpha(\epsilon, t_f) \in L(\mathcal{N}_\parallel, M_{0,\parallel})$.

Also, recall that M_1 is a non-indicator marking. Therefore, there exists a non-fault sequence $v \in T_N^*$, where $v = v_1 v_2 \dots v_{|v|}$, and an integer $k < |v|$, such that

$$M_1 \xrightarrow{v_1 \dots v_k}_{\mathcal{N}_N} M'_1 \xrightarrow{v_{k+1} \dots v_{|v|}}_{\mathcal{N}_N} M''_1 \text{ and } M'_1 \leq M''_1$$

Then we define a sequence in $\mathcal{N}_\parallel$

$$\beta := (v_1, v'_1)(v_2, v'_2) \dots (v_{|v|}, v'_{|v|}) \quad (9)$$

where for each $1 \leq i \leq |v|$, we have

$$v'_i = \begin{cases} t_e & \text{if } \mathcal{L}(v_i) = e \in \Sigma \\ \lambda & \text{if } \mathcal{L}(v_i) = \epsilon \end{cases} \quad (10)$$

Next, we show that $\alpha(\epsilon, t_f)\beta$ is a well-defined sequence in $\langle \mathcal{N}_\parallel, M_{0,\parallel} \rangle$. Note that, since we have shown that $\alpha(\epsilon, t_f) \in L(\mathcal{N}_\parallel, M_{0,\parallel})$, it suffices to show that $M_{\alpha t_f} \xrightarrow{\beta}_{\mathcal{N}_\parallel}$, where $M_{\alpha t_f} = [M_1^\top \ \tilde{M}_2^\top]^\top$, $M_0 \xrightarrow{\sigma_2}_{\tilde{\mathcal{N}}} M_2 \xrightarrow{t_f}_{\tilde{\mathcal{N}}} \tilde{M}_2$. We proceed by induction on the length of β.

Induction Basis: Since $v_1 \in L(\mathcal{N}_N, M_1)$, it suffices to show that $v'_1 \in L(\tilde{\mathcal{N}}, \tilde{M}_2)$ and $\mathcal{L}(v_1) = \tilde{\mathcal{L}}(v'_1)$. We consider the following two cases: (i) $v_1 \in T_o$; and (ii) $v_1 \in T_{uo}$. For Case (i), we have $v'_1 = t_e$, where $e = \mathcal{L}(v_1)$. Therefore, by the definition of $\tilde{\mathcal{L}}$, we have $\tilde{\mathcal{L}}(t_e) = \mathcal{L}(v'_1) = \sigma$. Moreover, since $\tilde{M}_2$ is reached after firing a fault transition t_f, we know that p_{fault} contains a token, which implies that $t_e \in L(\tilde{\mathcal{N}}, \tilde{M}_2)$. For Case (ii), we have $v'_1 = \lambda$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TAC.2017.2756096, IEEE Transactions on Automatic Control

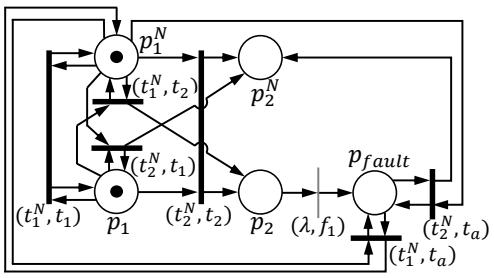


Fig. 3: Petri net $\langle \mathcal{N}_{\|}, M_{0,\|} \rangle$.

Therefore, $\tilde{\mathcal{L}}(v_1) = \mathcal{L}(v_1') = \epsilon$ and $\lambda \in L(\tilde{\mathcal{N}}, \tilde{M}_2)$. Overall, we have $\alpha(\epsilon, t_f)(v_1, v_1') \in L(\mathcal{N}_{\|}, M_{0,\|})$, i.e., the induction basis holds.

Induction Step: We assume that $M_{\alpha t_f} \xrightarrow{(v_1,v_1')\dots(v_i,v_i')}_{\mathcal{N}_{\|}}$ and we want to show that $M_{\alpha t_f} \xrightarrow{(v_1,v_1')\dots(v_i,v_i')(v_{i+1},v_{i+1}')}_{\mathcal{N}_{\|}}$ Since $v_1 \dots v_i v_{i+1} \in L(\mathcal{N}_N, M_1)$, it suffices to show that $v_1' \dots v_i' v_{i+1}' \in L(\tilde{\mathcal{N}}, \tilde{M}_2)$ and $\mathcal{L}(v_{i+1}) = \tilde{\mathcal{L}}(v_{i+1}')$. Note that in marking $M_{\alpha t_f}$, place p_{fault} contains a token and $v_1' \dots v_i'$ does not consume token in p_{fault}. Therefore, following the same reason in the induction basis, we have $\alpha(\epsilon, t_f)(v_1, v_1') \dots (v_i, v_i')(v_{i+1}, v_{i+1}') \in L(\mathcal{N}_{\|}, M_{0,\|})$.

Let $M_\beta' = [M_1'^\top \ \tilde{M}_\beta'^\top]^\top$ and $M_\beta = [M_1''^\top \ \tilde{M}_\beta^\top]^\top$ be markings such that

$$M_{\alpha t_f} \xrightarrow{(v_1,v_1')\dots(v_k,v_k')}_{\mathcal{N}_{\|}} M_\beta' \xrightarrow{(v_{k+1},v_{k+1}')\dots(v_{|v|},v_{|v|}')}_{\mathcal{N}_{\|}} M_\beta$$

Since v_i' is either a self-loop transition in the form of t_e or a λ-transition, we know that $\tilde{M}_2 = \tilde{M}_\beta' = \tilde{M}_\beta$. This together with the fact that $M_1' \le M_1''$ imply that $M_\beta' \le M_\beta$. Also, we know that

$$\bigvee_{t \in T_N \times (\tilde{T} \cup \{\lambda\})} \#_{(v_{k+1},v_{k+1}')\dots(v_{|v|},v_{|v|}')}(t) = |v| - k \ge 1$$

Overall, we have the following sequence in $\langle \mathcal{N}_{\|}, M_{0,\|} \rangle$

$$M_{0,\|} \xrightarrow{\alpha(\lambda,t_f)(v_1,v_1')\dots(v_k,v_k')}_{\mathcal{N}_{\|}} M_\beta' \xrightarrow{(v_{k+1},v_{k+1}')\dots(v_{|v|},v_{|v|}')}_{\mathcal{N}_{\|}} M_\beta \tag{11}$$

satisfying the condition in the theorem. □

Let us show how to use Theorem IV.1 to verify prognosability by the following example.

Example IV.1. Again, let us consider labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ shown in Figure 1(b), where $T_o = \{t_1, t_2\}$ and $T_F = \{f_1\}$. Its corresponding net $\langle \mathcal{N}_{\|}, M_{0,\|} \rangle$ is shown in Figure 3. As we discussed in Example III.2, this system is not prognosable; hereafter we show this using Theorem IV.1.

Let us consider the following sequence in $\langle \mathcal{N}_{\|}, M_{0,\|} \rangle$:

$$\begin{bmatrix}1\\0\\1\\0\\0\end{bmatrix} \xrightarrow{\overbrace{(t_1^N, t_2)(\lambda, f_1)}^{=:\alpha}} \underbrace{\begin{bmatrix}1\\0\\0\\0\\1\end{bmatrix}}_{=:M_1} \xrightarrow{\overbrace{(t_1^N, t_a)}^{=:\beta}} \underbrace{\begin{bmatrix}1\\0\\0\\0\\1\end{bmatrix}}_{=:M_2} \tag{12}$$

where the places in each marking are ordered by $\{p_1^N, p_2^N, p_1, p_2, p_{fault}\}$. Now, let us check that the above sequence satisfies Equation (8). Clearly, we have $M_4 = M_3$. Also, we know that $(\lambda, f_1) \in \{\lambda\} \times T_F$ and $\#_\alpha((\lambda, f_1)) = 1$. For β, we know that $(t_1^N, t_a) \in T_N \times (\tilde{T} \cup \{\lambda\})$ and $\#_\beta((t_1^N, t_a)) = 1$. Therefore, we know that the sequence in Equation (12) satisfies the conditions in Equation (8), i.e., the system is not prognosable.

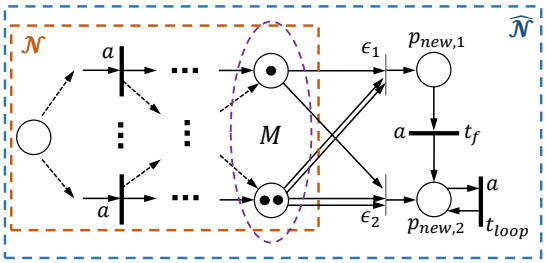


Fig. 4: Suppose that, in $\mathcal{N}$, we are interested in whether or not a marking M, in which each place in the red cycle contains the depicted number of tokens, can be covered. Then a conceptual illustration of $\langle \hat{\mathcal{N}}, M_0, \hat{\mathcal{L}} \rangle$ is shown in the figure.

Note that the conditions in Equation (8) is a valid formula in Definition II.1. Moreover, it satisfies the constraint that $F \Rightarrow M_1 \le M_k$. Since the length of the formula in Equation (8) is a constant, by Theorem IV.1 and [2] we have the following result immediately.

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

V. EXPSPACE-Completeness of Prognosability

In the preceding section, we have shown that the verification of prognosability for unbounded Petri nets can be done in EXPSPACE. However, this is still an extremely high complexity and one may ask whether or not this complexity can be further improved. In this section, we show that this complexity is actually tight, i.e., the verification of prognosability is EXPSPACE-complete.

For unbounded Petri nets, it is well-known that the *coverability problem*, stated as follows, is EXPSPACE-complete [22], [25].

- Given: A Petri net $\langle \mathcal{N}, M_0 \rangle$ and a marking M.
- To Decide: Whether or not there exists $M_0 \xrightarrow{\sigma} M'$ such that $M \le M'$.

Next, we show that checking prognosability is EXPSPACE-hard by reducing the coverability problem to the prognosability verification problem.

Theorem V.1. Checking prognosability for unbounded labeled Petri nets is EXPSPACE-complete.

Proof. In Theorem IV.2, we have shown that this problem is in EXPSPACE. Therefore, it remains to show that it is also EXPSPACE-hard. To this end, we use the coverability problem for the purpose of reduction.

Let $\langle \mathcal{N} = (P, T, A, w), M_0 \rangle$ and M be the instance of the coverability problem. We construct a new labeled Petri net $\langle \hat{\mathcal{N}} = (\hat{P}, \hat{T}, \hat{A}, \hat{w}), \hat{M}_0, \hat{\mathcal{L}} \rangle$ as follows. First, $\hat{P}$ is obtained by adding two new places, $p_{new,1}$ and $p_{new,2}$, to P; $\hat{T}$ is obtained by adding four new transitions, $\epsilon_1, \epsilon_2, t_f$ and t_{loop}, to T. For any $t \in T$ and $p \in P$, $\hat{A}$ and $\hat{w}$ are the same as A and w. However, for the newly added transitions and places, we have the followings:

- For each $i = 1, 2$, ${}^\bullet\epsilon_i = \{p \in P : M(p) \ne 0\}$ with $\forall p \in P : \hat{w}(p, \epsilon_i) = M(p)$; and $\epsilon_i^\bullet = \{p_{new,i}\}$ with $\hat{w}(\epsilon_i, p_{new,i}) = 1$;
- ${}^\bullet t_f = \{p_{new,1}\}$ and $t_f^\bullet = \{p_{new,2}\}$ with $\hat{w}(p_{new,1}, t_f) = \hat{w}(t_f, p_{new,2}) = 1$;
- ${}^\bullet t_{loop} = t_{loop}^\bullet = \{p_{new,2}\}$ with $\hat{w}(p_{new,2}, t_{loop}) = \hat{w}(t_{loop}, p_{new,2}) = 1$.

The initial marking $\hat{M}_0 = [M_0^\top \ 0 \ 0]^\top$ (we assume newly added places $p_{new,1}$ and $p_{new,2}$ are the last two places in the marking). Finally, the labeling function $\hat{\mathcal{L}}$ is defined by: (i) the only two unobservable transitions are ϵ_1 and ϵ; (ii) all other transitions are observable with the same label a. A conceptual illustration showing

the construction of $\hat{\mathcal{N}}$ is provided in Figure 4. Intuitively, transitions ϵ_1 and ϵ_2 can be fired iff marking M is covered in $\langle \mathcal{N}, M_0 \rangle$.

For the above constructed $\langle \hat{\mathcal{N}}, \hat{M}_0, \hat{\mathcal{L}} \rangle$, now, let us assume that t_f is the only fault transition we want to predict. Next, we show that M can be covered in $\langle \mathcal{N}, M_0 \rangle$ if and only if $\langle \hat{\mathcal{N}}, \hat{M}_0, \hat{\mathcal{L}} \rangle$ is not prognosable w.r.t. $T_F = \{t_f\}$.

The "if" part is straightforward. Suppose that M cannot be covered in $\langle \mathcal{N}, M_0 \rangle$. Then by the construction of $\langle \hat{\mathcal{N}}, \hat{M}_0, \hat{\mathcal{L}} \rangle$, we know that the only fault transition t_f can never fire in $\langle \hat{\mathcal{N}}, \hat{M}_0, \hat{\mathcal{L}} \rangle$. Therefore, $\langle \hat{\mathcal{N}}, \hat{M}_0, \hat{\mathcal{L}} \rangle$ is prognosable immediately.

To see the "only if" part, we assume that M can be covered in $\langle \mathcal{N}, M_0 \rangle$. Let $\sigma \in L(\mathcal{N}, M_0)$ be a sequence such that $M_0 \xrightarrow{\sigma}_{\mathcal{N}} M'$, where $M' \geq M$. By the construction of $\langle \hat{\mathcal{N}}, \hat{M}_0, \hat{\mathcal{L}} \rangle$, we know that $\hat{M}_0 \xrightarrow{\sigma}_{\hat{\mathcal{N}}} [M'^{\top} \ 0 \ 0]^{\top}$. Therefore, we know that $\hat{M}_0 \xrightarrow{\sigma\epsilon_1}_{\hat{\mathcal{N}}} M_1$ and $\hat{M}_0 \xrightarrow{\sigma\epsilon_2}_{\hat{\mathcal{N}}} M_2$, where $M_1(p_{new,1}) = 1$ and $M_2(p_{new,2}) = 1$. Since $M_1 \xrightarrow{t_f}_{\hat{\mathcal{N}}}$, we know that M_1 is a boundary marking. Since $M_2 \xrightarrow{(t_{loop})^K}_{\hat{\mathcal{N}}}$ for any $K \in \mathbb{N}$, we know that M_2 is a non-indicator marking. Moreover, $\hat{\mathcal{L}}(\sigma\epsilon_1) = \hat{\mathcal{L}}(\sigma\epsilon_2) = a^{|\sigma|}$. By Lemma III.1, we know that $\langle \hat{\mathcal{N}}, \hat{M}_0, \hat{\mathcal{L}} \rangle$ is not prognosable w.r.t. $\{t_f\}$. □

VI. Conclusion

In this technical note, we presented new results for prognosability analysis of partially-observed DES. A necessary and sufficient condition of prognosability for unbounded Petri nets was presented. In particular, this condition is stated in terms of a special class of formulas that can be effectively checked by existing model checking techniques. Moreover, we showed that the complexity of verifying prognosability for Petri nets is EXPSPACE-complete. This result reveals that extremely high computation complexity seems to be unavoidable in this verification problem. To mitigate the computational challenges, one possible direction is to find simple but sufficient conditions for prognosability using structural analysis. Another potential direction is to identify sub-classes of Petri nets for which the necessary and sufficient condition for prognosability can be verified more efficiently.

Note that the main purpose of verifying prognosability is to determine *a priori* if fault can be correctly predicted. When prognosability holds, how to design an efficient *online prognosis* mechanism is also an interesting and important problem. In fact, some works have been done on this direction; see, e.g., [1], [20]. Another future direction is to extend our results to the decentralized setting by considering the verification of co-prognosability; see., e.g., [18], [19], [36] for the case of finite-state automata.

Acknowledgments

The author would like to thank an anonymous reviewer for bringing [2] to our attention.

References

[1] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre. Faults prognosis using partially observed stochastic Petri nets. In *13th Int. Workshop on Discrete Event Systems*, pages 472–477, 2016.

[2] M. Atig and P. Habermehl. On Yen's path logic for Petri nets. *Reachability Problems*, pages 51–63, 2009.

[3] F. Basile, M.P. Cabasino, and C. Seatzu. State estimation and fault diagnosis of labeled time Petri net systems with unobservable transitions. *IEEE Trans. Automatic Control*, 60(4):997–1009, 2015.

[4] F. Basile, P. Chiacchio, and G. De Tommasi. On K-diagnosability of Petri nets via integer linear programming. *Automatica*, 48(9):2047–2058, 2012.

[5] A. Benveniste, E. Fabre, S. Haar, and C. Jard. Diagnosis of asynchronous discrete-event systems: a net unfolding approach. *IEEE Trans. Automatic Control*, 48(5):714–727, 2003.

[6] N. Bertrand, S. Haddad, and E. Lefaucheux. Foundation of diagnosis and predictability in probabilistic systems. In *34th IARCS Annual Conf. FSTTCS*, pages 417–429, 2014.

[7] M.P. Cabasino, A. Giua, S. Lafortune, and C. Seatzu. A new approach for diagnosability analysis of Petri nets using verifier nets. *IEEE Trans. Automatic Control*, 57(12):3104–3117, 2012.

[8] M.P. Cabasino, A. Giua, and C. Seatzu. Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46(9):1531–1539, 2010.

[9] C. G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, 2nd edition, 2008.

[10] F. Cassez. A note on fault diagnosis algorithms. In *48th IEEE Conf. Decision and Control*, pages 6941–6946, 2009.

[11] F. Cassez and A. Grastien. Predictability of event occurrences in timed systems. In *Formal Mod. Anal. Timed Syst.*, pages 62–76. 2013.

[12] J. Chen and R. Kumar. Stochastic failure prognosability of discrete event systems. *IEEE Trans. Autom. Contr.*, 60(6):1570–1581, 2015.

[13] L. Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American J. Mathematics*, 35(4):413–422, 1913.

[14] M. Dotoli, M.P. Fanti, A.M. Mangini, and W. Ukovich. On-line fault detection in discrete event systems by Petri nets and integer linear programming. *Automatica*, 45(11):2665–2672, 2009.

[15] S. Genc and S. Lafortune. Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, 45(2):301–311, 2009.

[16] T. Jéron, H. Marchand, S. Genc, and S. Lafortune. Predictability of sequence patterns in discrete event systems. In *Proc. 17th IFAC World Congress*, pages 537–543, 2008.

[17] G. Jiroveanu and R.K. Boel. The diagnosability of Petri net models using minimal explanations. *IEEE TAC*, 55(7):1663–1668, 2010.

[18] A. Khoumsi and H. Chakib. Conjunctive and disjunctive architectures for decentralized prognosis of failures in discrete-event systems. *IEEE Trans. Autom. Sci. Engin.*, 9(2):412–417, 2012.

[19] R. Kumar and S. Takai. Decentralized prognosis of failures in discrete event systems. *IEEE Trans. Autom. Contr.*, 55(1):48–59, 2010.

[20] D. Lefebvre. Fault diagnosis and prognosis with partially observed Petri nets. *IEEE Trans. S.M.C.: Syst.*, 44(10):1413–1424, 2014.

[21] D. Lefebvre. On-line fault diagnosis with partially observed Petri nets. *IEEE Trans. Automatic Control*, 59(7):1919–1924, 2014.

[22] R. Lipton. The reachability problem requires exponential space. *Research Report 62, Dep. Comput. Sci., Yale Univ.*, 1976.

[23] A. Madalinski, F. Nouioua, and P. Dague. Diagnosability verification with Petri net unfoldings. *Int. J. Knowledge-Based and Intelligent Engineering Syst.*, 14(2):49–55, 2010.

[24] F. Nouioua, P. Dague, and L. Ye. Predictability in probabilistic discrete event systems. In *Soft Methods for Data Sci.*, pages 381–389. 2017.

[25] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Sci.*, 6(2):223–231, 1978.

[26] 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. Engin.*, 4(1):31–39, 2007.

[27] Y. Ru and C.N. Hadjicostis. Fault diagnosis in discrete event systems modeled by partially observed Petri nets. *Discrete Event Dynamic Syst.: Theo. & Appl.*, 19(4):551–575, 2009.

[28] S. Takai. Robust prognosability for a set of partially observed discrete event systems. *Automatica*, 51:123–130, 2015.

[29] S. Takai and R. Kumar. Distributed failure prognosis of discrete event systems with bounded-delay communications. *IEEE Trans. Autom. Contr.*, 57(5):1259–1265, 2012.

[30] Y. Tong, Z. Li, C. Seatzu, and A. Giua. Decidability of opacity verification problems in labeled Petri net systems. *Automatica*, 80:48–53, 2017.

[31] L. Ye, P. Dague, and F. Nouioua. Predictability analysis of distributed discrete event systems. In *52nd CDC*, pages 5009–5015, 2013.

[32] L. Ye, P. Dague, and F. Nouioua. A predictability algorithm for distributed discrete event systems. In *International Conference on Formal Engineering Methods*, pages 201–216. Springer, 2015.

[33] H.-C. Yen. A unified approach for deciding the existence of certain Petri net paths. *Inform. Computation*, 96(1):119–137, 1992.

[34] X. Yin and S. Lafortune. A general approach for solving dynamic sensor activation problems for a class of properties. In *54th IEEE Conference on Decision and Control*, pages 3610–3615, 2015.

[35] X. Yin and S. Lafortune. On the decidability and complexity of diagnosability for labeled Petri nets. *IEEE Trans. Autom. Contr.*, 2017.

[36] X. Yin and Z.-J. Li. Decentralized fault prognosis of discrete event systems with guaranteed performance bound. *Automatica*, 69:375–379, 2016.

[37] X. Yin and Z.-J. Li. Reliable decentralized fault prognosis of discrete-event systems. *IEEE Trans. S.M.C.: Syst.*, 49(10), 2016.