2018 IEEE Conference on Decision and Control (CDC)
Miami Beach, FL, USA, Dec. 17-19, 2018

Verification of Coprognosability in Decentralized Fault Prognosis of Labeled Petri Nets

Wenqing Wu, Xiang Yin and Shaoyuan Li

Abstract— **We investigate the problem of decentralized fault prognosis of discrete-event systems modeled by unbounded labeled Petri nets. We assume that the system is monitored by a set of local agents (prognosers) with local observations so that they can predict the occurrence of fault in the system as a team. It is known in the literature that the notion of *coprognosability* provides the necessary and sufficient condition for the existence of a set of decentralized prognosers so that any fault can be predicted before its occurrence without false alarm. In this paper, we investigate the verification of coprognosability for systems modeled by labeled Petri nets. We show that coprognosability is decidable even when the Petri net is unbounded. Specifically, we provide an approach to transform the coprognosability verification problem to a model checking problem that can be effectively solved. Our result extends existing works on coprognosability analysis in decentralized fault prognosis from regular languages to Petri net languages.**

I. Introduction

We investigate the fault prognosis problem for Discrete-Event Systems (DES) modeled by labeled Petri nets. This problem is crucial in many safety-critical systems, where we need to *predict* the occurrences of faults so that some protecting actions can be taken before a fault is encountered. In this paper, we consider DES modeled by labeled Petri nets, a computational model that is widely used in modeling cyber-physical systems with concurrency, e.g., flexible manufacturing systems and software systems. Our goal is to analyze *a priori* whether or not any fault in the system can be successfully predicted.

In the context of DES, the problem of fault prognosis has drawn considerable attention in the past years; see, e.g., [6], [11], [15], [16], [19], [21], [23]–[25], [28], [30]–[32]. This problem was initially studied in [12], [13], where the authors consider DES modeled by finite-state automata and a language-based property called *predictability* (or prognosability) was proposed. Specifically, predictability is the necessary and sufficient condition under which there exists a centralized prognoser that can always issue a fault alarm before the occurrence of fault without any false alarm. Recently, predictability/prognosability analysis has also been investigated for DES modeled by timed automata [9], stochastic automata [6], [10], [11], [19] and fuzzy automata [5]. For DES modeled by Petri nets, the authors in [1]–[3], [17] have investigated online fault prognosis methods for both logical and stochastic Petri nets. Recently, it has been shown in [27] the condition of prognosability is decidable for unbounded Petri nets in the centralized setting.

In many large-scale systems, the information structure are naturally decentralized. For such systems, centralized fault prognosis algorithms cannot be directly applied due to the information constraint and we need to develop corresponding decentralized prognostic protocols for the purpose of fault prognosis. Therefore, the problem of decentralized fault prognosis has also drawn many attention in the past years; see, e.g., [14]–[16], [22], [31], [32]. In this setting, it is assumed that the system is monitored by a set of *local prognosers*. Each local prognoser has its own observation and can send local prognostic decision to a *coordinator*, where a final global prognostic decision is issued. Particularly, in [16], the authors investigated the decentralized fault prognosis problem under the *disjunctive architecture*, where a global fault alarm is issued if one local fault alarm is issued. Moreover, the authors in [16] proposed a notion called *coprognosability* as the necessary and sufficient condition under which there exists a decentralized prognoser that can successfully predict fault in the disjunctive architecture. In [15] and [22], the decentralized fault prognosis problem has been further studied in the conjunctive architecture and the inference-based architecture, respectively.

In this paper, we study the decentralized fault prognosis problem for (unbounded) labeled Petri nets in the disjunctive architecture. Specifically, we are interested in verifying coprognosability for Petri net languages in order to determine a priori if the fault prognostic task can be accomplished. Note that, since the state space of a Petri net is unbounded in general, existing approaches based on finite-state automata cannot be applied to our problem. The main contributions of this paper are as follows. First, we provide a necessary and sufficient condition for coprognosability of DES modeled by unbounded Petri nets. Then we present an effective construction to capture this condition and show that checking coprognosability is decidable for general unbounded labeled Petri nets. To the best of our knowledge, existing works on decentralized fault prognosis mentioned above assume that the systems are modeled by finite-state automata and the decentralized fault prognosis problem has never been studied for systems modeled by Petri nets. Therefore, our results extend existing works on coprognosability analysis from regular languages to Petri nets languages.

The remaining part of the paper is organized as follows. Section II presents necessary preliminaries. In Section III,

This work is supported by the National Natural Science Foundation of China (61803259, 61833012).

W. Wu, X. Yin and S. Li are with Department of Automation and Key Laboratory of System Control and Information Processing, Shanghai Jiao Tong University, Shanghai 200240, China. {wuwenqing.dy,yinxiang,syli}@sjtu.edu.cn.

978-1-5386-1394-8/18/$31.00 ©2018 IEEE

coprognosability is defined for labeled Petri nets and its relevant properties are also discussed. In Section IV, we provide a necessary and sufficient condition for coprognosability, which can be effectively checked by an existing model checking problem. Finally, we conclude the paper in Section V.

II. Preliminaries

A (place/transition) *net* is a directed bipartite graph, defined as a 4-tuple $\mathcal{N} = (P, T, A, \omega)$, where $P = \{p_1, p_2, \ldots, p_n\}$ is the finite set of places, $T = \{t_1, t_2, \ldots, t_m\}$ is the finite set of transitions, $A \subseteq (P \times T) \cup (T \times P)$ is the set of directed arcs from places to transitions and from transitions to places and $\omega : A \to \mathbb{N}$ is the weight function that assigns to each arc a non-negative integer. We introduce the empty transition λ and we define $T^+ := T \cup \{\lambda\}$. For any place $p \in P$, we denote its preset by $I(p)$, i.e., $I(p) := \{t \in T : (t,p) \in A\}$; we denote its postset by $O(p)$, i.e., $O(p) := \{t \in T : (p,t) \in A\}$. For any transition t, we define its preset $I(t)$ and its postset $O(t)$ 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 $M(p_i)$ is the number of tokens in place $p_i \in P$. A *Petri net* is a pair $\langle \mathcal{N}, M_0 \rangle$, where $\mathcal{N}$ is a net and $M_0 \in \mathbb{N}^n$ is the initial markings. We say that transition t is enabled at state M if $\forall p \in I(t) : M(p) \geq \omega(p,t)$. If t is enabled at M, then it can *fire* and yields a new marking M' defined by $\forall p \in P : M'(p) = M(p) - \omega(p,t) + \omega(t,p)$. We denote by $M \xrightarrow{t}_{\mathcal{N}}$ that transition $t \in T$ is enabled at M in net $\mathcal{N}$ and denote by $M \xrightarrow{t}_{\mathcal{N}} M'$ that the firing of t at M yields M' in net $\mathcal{N}$. We will also omit the subscript $\mathcal{N}$ when the net is clear from the context.

Let T^* denote the set of all finite sequences of transitions (or, for simplicity, sequences) including the empty transition λ. For any $\sigma \in T^*$, we have $\sigma\lambda = \lambda\sigma = \sigma$. A sequence $\sigma = t_1 t_2 \ldots t_k \in T^*$ is said to be enabled at M if $\forall i \in \{1, \ldots, k\} : M_i \xrightarrow{t_i}$, where $M_1 = M$ and $M_i \xrightarrow{t_i} M_{i+1}$. Analogously, we denote by $M \xrightarrow{\sigma}_{\mathcal{N}}$ that $\sigma \in T^*$ can be fired at M and by $M \xrightarrow{\sigma}_{\mathcal{N}} M'$ that firing σ yields M'. For a Petri net $\langle \mathcal{N}, M_0 \rangle$, $L(\mathcal{N}, M_0)$ denotes the set of finite sequences which are enabled at M_0, i.e., $L(\mathcal{N}, M_0) = \{\sigma \in T^* : M_0 \xrightarrow{\sigma}_{\mathcal{N}}\}$. Given a sequence $\sigma \in T^*$, we denote by $\overline{\sigma}$ the set of prefixes of σ, i.e., $\overline{\sigma} = \{\sigma_i \in T^* : \exists \sigma_j \in T^*\ s.t.\ \sigma_i \sigma_j = \sigma\}$. Finally, the length of sequence σ is denoted by $|\sigma|$. For any sequence $\sigma \in T^*$ and any transition $t \in T$, we denote by $\#_\sigma(t)$ the number of times t occurs in σ.

We define Σ as a finite set of events (or alphabets). A string is a finite sequence of events and we use Σ^* to denote 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 assigns to each transition an event. In other words, for any $t \in T$, $\mathcal{L}(t)$ indicates the event that can be observed when t fires. Given a transition $t \in T$, we call it *observable* if $\mathcal{L}(t) \in \Sigma$; otherwise, t is *unobservable*. Therefore, T is naturally partitioned as $T = T_o \dot{\cup} T_{uo}$, where T_o and T_{uo} are the sets of observable and unobservable transitions, respectively. The labeling function $\mathcal{L}$ can also be extended from T to T^* recursively by:

(i) $\mathcal{L}(\lambda) = \epsilon$; and
(ii) $\forall \sigma \in T^*, t \in T : \mathcal{L}(\sigma t) = \mathcal{L}(\sigma)\mathcal{L}(t)$.

Therefore, the language generated by labeled Petri net $\langle \mathcal{N}, M_0, \mathcal{L} \rangle$ is a set of strings defined by $\mathcal{L}(L(\mathcal{N}, M_0)) := \{\mathcal{L}(\sigma) : \sigma \in L(\mathcal{N}, M_0)\}$.

Finally, we make the following standard assumption in the analysis of partially-observed DES

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

III. Coprognosability of Labeled Petri Nets

In the decentralized fault prognosis problem, the system $\langle \mathcal{N}, M_0 \rangle$ is monitored by a set of n local agents (or local prognosers). We denote by $\mathcal{I} = \{1, 2, \ldots, n\}$ the index set. We assume that each agent has its own observation. Formally, for any $i \in \mathcal{I}$, we denote by $\mathcal{L}_i : T \to \Sigma \cup \{\epsilon\}$ its local labeling function and we denote by $T_{o,i}$ and $T_{uo,i}$ the sets of observable transitions and unobservable transitions w.r.t. $\mathcal{L}_i$, respectively. Therefore, a labeled Petri net in the decentralized setting is written by $\langle \mathcal{N}, M_0, \{\mathcal{L}_i\}_{i \in \mathcal{I}} \rangle$.

In the fault prognosis problem, we assume that the set of transitions is further partitioned as two disjoint sets $T = T_N \cup T_F$, where T_N is the set of normal transitions and T_F is the set of fault transitions. For any sequence $\sigma = t_1 t_2 \ldots t_k \in T^*$, with a slight abuse of notation, we denote by $T_F \in \sigma$ if σ contains a fault transition, i.e., $\exists i \in \{1, \ldots, k\} : t_i \in T_F$.

In order to characterize whether or not any fault in a decentralized system can be predicted, a language-based condition called coprognosability was proposed in [16]. This definition is reviewed as follows in the context of Petri net languages.

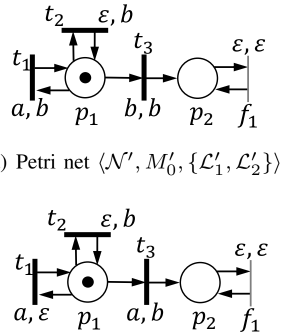
Definition 3.1 (Coprognosability): Let $\langle \mathcal{N}, M_0, \{\mathcal{L}_i\}_{i \in \mathcal{I}} \rangle$ be a labeled Petri net with decentralized information. Then $\langle \mathcal{N}, M_0, \{\mathcal{L}_i\} \rangle$ is said to be coprognosable w.r.t. T_F if

$$\begin{aligned} &(\forall \alpha \in L(\mathcal{N}, M_0) : T_F \in \alpha)(\exists \beta \in \overline{\alpha} : T_F \notin \beta) \\ &(\exists i \in \mathcal{I})(\forall \theta \in L(\mathcal{N}, M_0) : \mathcal{L}_i(\theta) = \mathcal{L}_i(\beta) \wedge T_F \notin \theta) \\ &(\exists K \in \mathbb{N})(\forall \theta\gamma \in L(\mathcal{N}, M_0))[|\gamma| \geq K \Rightarrow T_F \in \gamma] \qquad (1) \end{aligned}$$

Intuitively, coprognosability requires that for any fault sequence, it must have a non-fault prefix such that, at least one local prognoser knows for sure that a fault will occur in a finite number of steps. Therefore, a local fault alarm can be sent to the coordinator in order to issue a global fault alarm before the fault actually occurs. It has been shown in [16] that coprognosability provides the necessary and sufficient condition under which there exists a set of local prognosers that can achieve the following two requirements under the disjunctive architecture[1]

- Any fault can be predicted before its occurrence; and
- A fault will occur in a finite number of steps once a fault alarm is issued.

[1] In the disjunctive architecture, the coordinator issues a global alarm if one local prognoser issues a fault alarm.



(a) Petri net $\langle \mathcal{N}', M_0', \{\mathcal{L}_1', \mathcal{L}_2'\}\rangle$

(b) Petri net $\langle \mathcal{N}, M_0, \{\mathcal{L}_1, \mathcal{L}_2\}\rangle$

Fig. 1. For each transition t, its associated label (e_1, e_2) means that $\mathcal{L}_1(t) = e_1$ and $\mathcal{L}_3(t) = e_2$.

The reader is referred to [16] for how coprognosability guarantees the above two conditions. In this paper, we will focus on how to verify this condition for languages generated by unbounded Petri nets.

First, let us illustrate the notion of coprognosability in Petri nets by the following examples.

Example 3.1: Let us consider labeled Petri net $\langle \mathcal{N}', M_0', \{\mathcal{L}_1', \mathcal{L}_2'\}\rangle$ with two local agents shown in Figure 1(a), where $T_{o,1} = \{t_1, t_3\}$, $T_{o,2} = \{t_1, t_2, t_3\}$ and $T_F = \{f_1\}$. Let $\Sigma = \{a, b\}, \mathcal{L}_1'(t_1) = a, \mathcal{L}_1'(t_3) = b$ and $\mathcal{L}_2'(t_1) = \mathcal{L}_2'(t_2) = \mathcal{L}_2'(t_3) = b$. Note that transition t_3 has to fire before the occurrence of fault transition f_1, and once it fires, the first agent observes event $\mathcal{L}_1'(t_3) = b$, which can only be generated by transition t_3 through labeling function $\mathcal{L}_1'$, it can issue a fault alarm unambiguously. Hence, the system is coprognosable, although from the perspective of $\mathcal{L}_2'$, all the observable transitions have the same label.

Example 3.2: Let us consider labeled Petri net $\langle \mathcal{N}, M_0, \{\mathcal{L}_1, \mathcal{L}_2\}\rangle$ with two local agents shown in Figure 1(b), where $T_{o,1} = \{t_1, t_3\}$, $T_{o,2} = \{t_2, t_3\}$ and $T_F = \{f_1\}$. Let $\Sigma = \{a, b\}, \mathcal{L}_1(t_1) = \mathcal{L}_1(t_3) = a$ and $\mathcal{L}_2(t_2) = \mathcal{L}_2(t_3) = b$. This system is not coprognosable. To reveal this, we consider fault sequence $t_3 f_1 \in L(\mathcal{N}, M_0)$. Then for $t_3 \in \overline{t_3 f_1} : T_F \notin t_3$, we can find $t_1 \in L(\mathcal{N}, M_0)$ such that $\mathcal{L}_1(t_1) = \mathcal{L}_1(t_3) = a$ and an arbitrarily long non-fault sequence $M_0 \xrightarrow{t_1 (t_1)^k}, k \in \mathbb{N}$ is defined. Similarly, for t_3, we can also find $t_2 \in L(\mathcal{N}, M_0)$ such that $\mathcal{L}_2(t_2) = \mathcal{L}_2(t_3) = b$ and an arbitrarily long non-fault sequence $M_0 \xrightarrow{t_2 (t_2)^k}, k \in \mathbb{N}$ is defined. In other words, when t_3 occurs, none of the agents knows for sure that a fault will occur in a finite number of steps. Therefore, fault sequence $t_3 f_1$ cannot be alarmed correctly, i.e., the system is not coprognosable.

Before proceeding to the verification of coprognosability for labeled Petri nets, we first define the notions of *boundary marking* and *non-indicator marking*, which was originally introduced in [27].

Definition 3.2: 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 $(\forall K \in \mathbb{N})(\exists \sigma \in T_N^* : |\sigma| \geq K)[M \xrightarrow{\sigma}]$.

Intuitively, a boundary marking is a marking from which a fault can occur in the next step and a non-indicator marking is a marking from which an arbitrarily long non-fault sequence can occur. Since a marking is a vector with its elements taken from the set of nonnegative integers, there does not exist a strictly monotone decreasing sequence of markings. Hence, under the deadlock-free assumption, M is a non-indicator marking if and only if

$$\exists \sigma_1, \sigma_2 \in T_N^* \text{ s.t. } M \xrightarrow{\sigma_1} M_1 \xrightarrow{\sigma_2} M_2 \text{ and } M_1 \leq M_2.$$

Next, we introduce the following lemma that illustrates how to decide coprognosability according to the above two notions.

Lemma 3.1: Labeled Petri net $\langle \mathcal{N}, M_0, \{\mathcal{L}_i\}_{i \in \mathcal{I}}\rangle$ is not coprognosable w.r.t. T_F, if and only if, there exist $n+1$ non-fault sequences $\sigma_B, \sigma_1, \ldots, \sigma_n \in T_N^*$ such that

1) M_B is a boundary marking, where $M_0 \xrightarrow{\sigma_B} M_B$; and
2) For any $i \in \mathcal{I}$, M_i is a non-indicator marking, where $M_0 \xrightarrow{\sigma_i} M_i$; and
3) For any $i \in \mathcal{I}$, $\mathcal{L}_i(\sigma_B) = \mathcal{L}_i(\sigma_i)$.

IV. Verification of Coprognosability

In the previous section, we have discussed the definition of coprognosability for labeled Petri nets and provided an approach to characterize coprognosability based on the notions of boundary markings and non-indicator markings. In this section, we provide a verifiable necessary and sufficient condition for coprognosability in order to show that coprognosability is indeed decidable.

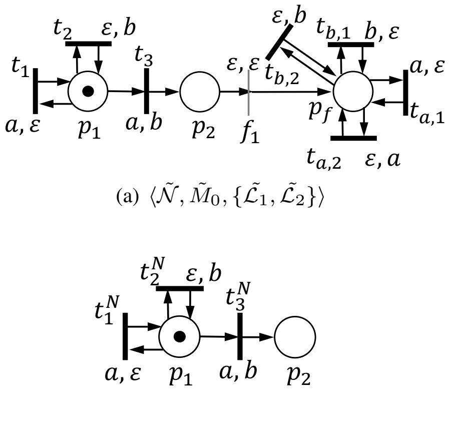
According to Lemma 3.1, the verification of coprognosability is equivalent to checking the existence of a "leading" sequence σ_B that goes to a boundary marking and n "following" sequences $\sigma_1, \ldots, \sigma_n$, each of them goes to a non-indicator marking, such that σ_B and σ_i look the same under the i-th observation mapping $\mathcal{L}_i$. To this end, we need to construct a new net that tracks all such leading and following sequences.

Let $\langle \mathcal{N}, M_0, \{\mathcal{L}_i\}_{i \in \mathcal{I}}\rangle$ be the original labeled Petri net. First, we define a new labeled Petri net

$$\langle \tilde{\mathcal{N}}, \tilde{M}_0, \{\tilde{\mathcal{L}}_i\}_{i \in \mathcal{I}}\rangle, \text{ where } \tilde{\mathcal{N}} = (\tilde{P}, \tilde{T}, \tilde{A}, \tilde{\omega})$$

as follows:

- $\tilde{P} = P \cup \{p_f\}$, where p_f is a new place;
- $\tilde{T} = T \cup \{t_{e,i} : e \in \Sigma, i \in \mathcal{I}\}$, where each $t_{e,i}$ is a new transition defined for each $e \in \Sigma, i \in \mathcal{I}$;
- $\tilde{A}$ and $\tilde{\omega}$ are defined by:
  - For any $t \in T_N$, $I(t)$ and $O(t)$ are the same as in $\mathcal{N}$ and $\forall p \in P : \tilde{\omega}(p, t) = \omega(p, t), \tilde{\omega}(t, p) = \omega(t, p)$.
  - For any $t \in T_F$, $I(t)$ is the same as in $\mathcal{N}$ with $\forall p \in P : \tilde{\omega}(p, t) = \omega(p, t)$, while $O(t) = \{p_f\}$ with $\tilde{\omega}(t, p_f) = 1$.
  - For any $t_{e,i}, e \in \Sigma, i \in \mathcal{I}$, $I(t) = O(t) = \{p_f\}$ and $\tilde{\omega}(t_{e,i}, p_f) = \tilde{\omega}(p_f, t_{e,i}) = 1$.
- The initial marking is $\tilde{M}_0 = [M_0^\top \; 0]^\top$, where we assume that the last place is p_f.



(a) $\langle \tilde{\mathcal{N}}, \tilde{M}_0, \{\tilde{\mathcal{L}_1}, \tilde{\mathcal{L}_2}\}\rangle$

(b) $\langle \mathcal{N}_N, M_0, \{\mathcal{L}_1, \mathcal{L}_2\}\rangle$

Fig. 2. $\mathcal{N}_N$ and $\tilde{\mathcal{N}}$ are constructed based on the system in Figure 1(b).

- For each $i \in \mathcal{I}$, the labeling function $\tilde{\mathcal{L}}_i : \tilde{T} \to \Sigma \cup \{\epsilon\}$ is defined as

$$\tilde{\mathcal{L}}_i(t) = \begin{cases} \mathcal{L}_i(t) & \text{if } t \in T_N \\ \epsilon & \text{if } t \in T_f \\ e & \text{if } t = t_{e,j} \text{ and } i = j \\ \epsilon & \text{if } t = t_{e,j} \text{ and } i \neq j \end{cases}$$

Intuitively, $\tilde{\mathcal{N}}$ is a copy of $\mathcal{N}$ except for new place p_f and new transitions $t_{e,i}$. In particular, for any non-fault sequence, the dynamic of $\tilde{\mathcal{N}}$ and the labels coincide with those of $\tilde{\mathcal{N}}$. On the other hand, when a fault transition fires, a token is sent to new place p_f in $\tilde{\mathcal{N}}$ signifying the occurrence of fault. For each $e \in \Sigma$, a series of self-loop transitions $t_{e,i}$ are correspondingly defined at p_f, so that for any non-fault sequence whose label is $e_1 \dots e_k$ under $\mathcal{L}_i$, there always exists a self-loop sequence in the form of $t_{e_1,i} \dots t_{e_k,i}$ that produces the same observation under $\tilde{\mathcal{L}}_i$ but produces empty observation under $\tilde{\mathcal{L}}_j, j \neq i$. As will become clear later, such a construction is used to find non-indicator markings.

Also, we denote by $\mathcal{N}_N$ the *normal net* obtained by removing transitions in T_F from $\mathcal{N}$. We denote by $\mathcal{N}_{N,1}, \dots, \mathcal{N}_{N,n}$ n copies of $\mathcal{N}_N$ with disjoint places $P_{N,i}$ and transitions $T_{N,i}$. For example, for Petri net shown in Figure 1(b), $\tilde{\mathcal{N}}$ and $\mathcal{N}_N$ are shown in Figures 2(a) and 2(b), respectively.

Next, we build a new (unlabeled) Petri net $\langle \mathcal{N}_{||}, M_{0,||}\rangle$ which synchronizes $\langle \tilde{\mathcal{N}}, \tilde{M}_0\rangle$ with $\langle \mathcal{N}_{N,1}, M_0\rangle, \dots, \langle \mathcal{N}_{N,n}, M_0\rangle$ so that the move in the first component and the $(i+1)$-th component have the same label under $\tilde{\mathcal{L}}_i$ and $\mathcal{L}_i$. Formally, Petri net

$$\langle \mathcal{N}_{||}, M_{0,||}\rangle, \text{ where } \mathcal{N}_{||} = (P_{||}, T_{||}, A_{||}, \omega_{||}),$$

is defined as follows:

- $P_{||} = \tilde{P} \cup P_{N,1} \cup P_{N,2} \cup \dots \cup P_{N,n}$;
- $T_{||} \subseteq \tilde{T}^+ \times T_{N,1}^+ \times T_{N,2}^+ \times \dots \times T_{N,n}^+ \setminus \{\underbrace{(\lambda, \dots, \lambda)}_{(n+1)\text{times}}\}$;
- $A_{||}$ and $\omega_{||}$ are defined by:
  - For any $t_0 \in \tilde{T}$, transition $(t_0, t_1, t_2, \dots, t_n) \in T_{||}$ is defined if, for any $i \in \mathcal{I}$, we have

$$[\tilde{\mathcal{L}}_i(t_0) = \mathcal{L}_i(t_i)] \wedge [\tilde{\mathcal{L}}_i(t_0) = \epsilon \Rightarrow t_i = \lambda] \quad (2)$$

  - For each $i \in \mathcal{I}$, transition $(t_0, t_1, \dots, t_i, \dots, t_n) \in T_{||}$ is defined if

$$[\mathcal{L}_i(t_i) = \epsilon] \wedge [t_i \neq \lambda] \wedge [\forall j \geq 0, j \neq i : t_j = \lambda] \quad (3)$$

  - For each defined transition $t_{||} = (t_0, t_1, t_2, \dots, t_n) \in T_{||}$, we have $I(t_{||}) = \bigcup_{i \geq 0} I(t_i)$ and $O(t_{||}) = \bigcup_{i \geq 0} O(t_i)$. Also,

$$\omega_{||}(t_{||}, p) = \begin{cases} \tilde{\omega}(t_0, p) & \text{if } p \in \tilde{P} \\ \omega_{N,i}(t_i, p) & \text{if } p \in P_{N,i} \end{cases}$$

$$\omega_{||}(p, t_{||}) = \begin{cases} \tilde{\omega}(p, t_0) & \text{if } p \in \tilde{P} \\ \omega_{N,i}(p, t_i) & \text{if } p \in P_{N,i} \end{cases}$$

- $M_{0,||} = [\tilde{M}_0^\top \; M_0^\top \; M_0^\top \dots M_0^\top]^\top$.

Intuitively, net $\langle \mathcal{N}_{||}, M_{0,||}\rangle$ consists of $(n+1)$ components, where the first component $\tilde{\mathcal{N}}$ is used to track the "leading" sequence that goes to a boundary marking and the $(i+1)$-th component $\mathcal{N}_{N,i}$ is used to track the i-th "following" sequence that goes to a non-indicator marking. Moreover, for each $i \in \mathcal{I}$, the leading sequence and the i-th following sequence have the same observation based on the labeling functions $\mathcal{L}_i$ and $\tilde{\mathcal{L}}_i$.

More specifically, for any transition $t_{||} = (t_0, t_1, \dots, t_i, \dots, t_n) \in T_{||}$ satisfying Equation (2), the leading transition moves and each following component should move accordingly if this transition does not look as ϵ locally. Similarly, for any transition $t_{||} = (t_0, t_1, \dots, t_i, \dots, t_n) \in T_{||}$ satisfying Equation (3), each following component can execute a locally unobservable transition without involving the leading component and other following components. To sum up, for any sequence $\sigma = (\sigma', \sigma_1, \sigma_2, \dots, \sigma_n) \in L(\mathcal{N}_{||}, M_{0,||})$ we can conclude that $\tilde{\mathcal{L}}_i(\sigma') = \mathcal{L}_i(\sigma_i)$. On the other hand, for any $\sigma' \in L(\tilde{\mathcal{N}}, \tilde{M}_0)$, $\sigma_i \in L(\mathcal{N}_{N,i}, M_0)$ and $\tilde{\mathcal{L}}_i(\sigma') = \mathcal{L}_i(\sigma_i), i \in \mathcal{I}$, there exists a sequence $\sigma \in L(\mathcal{N}_{||}, M_{0,||})$ such that it is in the form of $(\sigma', \sigma_1, \sigma_2, \dots, \sigma_n)$ (possibly by inserting λ). This construction of $\mathcal{N}_{||}$ is motivated by the so called verifier (or twin-machine) construction for diagnosability and prognosability analysis; see, e.g., [7], [8], [18], [20], [29]. Here, we adopt the basic idea and extend it to the decentralized setting for the purpose of fault prognosis with non-trivial modifications.

The following theorem reveals how to use net $\langle \mathcal{N}_{||}, M_{0,||}\rangle$ to test coprognosability for labeled Petri nets.

Theorem 4.1: Labeled Petri net $\langle \mathcal{N}, M_0, \{\mathcal{L}_i\}_{i \in \mathcal{I}}\rangle$ is not coprognosable w.r.t. T_F, if and only if, there exists a sequence

$$M_{0,||} \xrightarrow{\alpha}_{\mathcal{N}_{||}} M_1 \xrightarrow{\beta}_{\mathcal{N}_{||}} M_2 \quad (4)$$

in $\langle \mathcal{N}_{||}, M_{0,||}\rangle$ such that

$$(M_2 \geq M_1) \wedge (\bigvee_{t\in T_F\times\{\lambda\}\times\cdots\times\{\lambda\}} \#_\alpha(t) \geq 1) \wedge \tag{5}$$
$$(\bigwedge_{i\in\mathcal{I}} \bigvee_{t\in\tilde{T}^+\times T^+_{N,1}\times\cdots\times T^+_{N,i-1}\times T_{N,i}\times T^+_{N,i+1}\times\cdots\times T^+_{N,n}} \#_\beta(t) \geq 1)$$

Let us explain the basic idea of the above theorem. Note that, each sequence in $\mathcal{N}_{||}$ consists of $(n+1)$ components. We denote by α_{i-1} the i-th components in α, i.e., $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_n)$; the same for β. Now, let us assume that there exists a sequence satisfying the three conditions. Since $\bigvee_{t\in T_F\times\{\lambda\}\times\cdots\times\{\lambda\}} \#_\alpha(t) \geq 1$, we know that the first component α_0 must contain a fault transition and there exists a prefix of α_0 leading to a boundary marking. Also, the last condition implies that $\beta_1, \beta_2, \ldots, \beta_n$ are all non-λ. This together with $M_2 \geq M_1$ guarantees that any marking reached by a prefix of $\alpha_1, \alpha_2, \ldots, \alpha_n$ is a non-indicator marking, since each β_i can be fired for an arbitrary number of times. Thus, the conditions in Theorem 4.1 essentially ensure that there exist a leading sequence that goes to a boundary marking and n following sequences, each of them goes to a non-indicator marking, such that the leading sequence and each following sequence are locally observationally equivalent, which disproves coprognosability. On the other hand, if the labeled Petri net is not coprognosable, i.e., there exists a sequence α such that $M_{0,||} \xrightarrow{\alpha}_{\mathcal{N}_{||}} M_1 = [M_B^\top\ M_{1,1}^\top\ M_{1,2}^\top \ldots M_{1,n}^\top]^\top$, where M_B is a boundary marking and $M_{1,1}, M_{1,2}, \ldots, M_{1,n}$ are n non-indicator markings. As all fault transitions are unobservable in $\tilde{\mathcal{L}}$, we have $M_1 \xrightarrow{(t_f,\lambda,\ldots,\lambda)}_{\mathcal{N}_{||}} M_2$ for some $t_f \in T_F$. Such an M_2 can be extended to a covering since the self-loops in the form of $t_{e,i}$ in $\tilde{\mathcal{N}}$ can track any sequence that forms a covering for each component.

In the above theorem, we have shown that checking non-coprognosability is equivalent to checking the existence of a specific sequence in net $\langle \mathcal{N}_{||}, M_{0,||}\rangle$. This condition is actually verifiable as a special case of Petri nets model checking problem called the Yen's problem [26]. This leads to the decidability of coprognosability.

Theorem 4.2: Checking coprognosability is decidable for label Petri nets. Moreover, it is in EXPSPACE.

Proof: The decidability comes from the fact that the existence of a sequence satisfying the Equation (5) is expressed by a logic called Yen's formula [4], [26], whose satisfiability is decidable. Specifically, Yen's result allows us to test the existence of a sequence $M_0 \xrightarrow{\sigma_1} M_1 \xrightarrow{\sigma_2} \cdots M_{k-1} \xrightarrow{\sigma_k} M_k$ such that a predicate $\phi(M_1, \ldots, M_k, \sigma_1, \ldots, \sigma_k)$ holds, where $\phi(M_1, \ldots, M_k, \sigma_1, \ldots, \sigma_k)$ is a marking and transition predicate obtained by conjunctions and disjunctions of the terms in the form of $M_i(p) = M_j(p')$, $M_i(p) > M_j(p')$, $\#_{\sigma_i}(t) \leq c$, $\#_{\sigma_i}(t) \geq c$ and $\#_{\sigma_i}(t) \leq \#_{\sigma_j}(t')$, where $t, t' \in T$, $p, p' \in P$ and c is an arbitrary constant. For more detail on Yen's logic, the reader is referred to the literature [26]. Moreover, it has been shown in [4] that the Yen's problem can be decided in EXPSPACE if $M_1 \leq M_k$ and there is no transition predicate. Clearly, our condition in Theorem 4.1 satisfies the Yen's formula and $M_1 \leq M_2$. Note that transitions predicates can be replaced by marking predicates. Moreover, the size of our condition is polynomial in the number of transitions and number of places in the original net (but is exponential in the number of local agents). Therefore, we conclude that checking coprognosability is decidable and it can be solved in EXPSPACE in the size of the original net when the number of local agents is fixed. ■

Finally, we show how to use Theorem 4.1 to verify coprognosability by the following example.

Example 4.1: Let us again consider labeled Petri net $\langle \mathcal{N}, M_0, \{\mathcal{L}_1, \mathcal{L}_2\}\rangle$ shown in Figure 1(b), where $T_{o,1} = \{t_1, t_3\}$, $T_{o,2} = \{t_2, t_3\}$ and $T_F = \{f_1\}$. Its corresponding net $\langle \mathcal{N}_{||}, M_{0,||}\rangle$ is depicted in Figure 3. For the sake of clarity, we use super-script N_i for each transition and place that correspond to $\tilde{\mathcal{N}}_{N,i}$. For example, transition is defined $(t_1, t_1^{N_1}, \lambda)$ in defined in $\mathcal{N}_{||}$, since $\mathcal{L}_1(t_1) = a$ and $\mathcal{L}_2(t_1) = \epsilon$, i.e., Equation (2) is fulfilled. Similarly, transition $(\lambda, \lambda, t_1^{N_2})$ is also defined since $\mathcal{L}_2(t_1) = \epsilon$; this is the situation described in Equation (3).

As we discussed in Example 3.2, the system is not coprognosable. Next we show this using Theorem 4.1. Specifically, we have the following sequence in $\langle \mathcal{N}_{||}, M_{0,||}\rangle$:

$$\underbrace{\begin{bmatrix}1\\0\\0\\1\\0\\1\\0\end{bmatrix}}_{=:M_{0,||}} \xrightarrow{\overbrace{(t_3,t_1^{N_1},t_2^{N_2})(f_1,\lambda,\lambda)}^{=:\alpha}} \underbrace{\begin{bmatrix}0\\0\\1\\1\\0\\1\\0\end{bmatrix}}_{=:M_1} \xrightarrow{\overbrace{(t_{a,1},t_1^{N_1},\lambda)}^{=:\beta_1}} \begin{bmatrix}0\\0\\1\\1\\0\\1\\0\end{bmatrix} \xrightarrow{\overbrace{(t_{b,2},\lambda,t_2^{N_2})}^{=:\beta_2}} \underbrace{\begin{bmatrix}0\\0\\1\\1\\0\\1\\0\end{bmatrix}}_{=:M_2}$$

where places in each marking are ordered by $\{p_1, p_2, p_f, p_1^{N_1}, p_2^{N_1}, p_1^{N_2}, p_2^{N_2}\}$. This sequence satisfies the condition in Equation (5). First, we have $(f_1, \lambda, \lambda) \in T_F \times \{\lambda\} \times \{\lambda\}$ and $\#_\alpha((f_1, \lambda, \lambda)) = 1$. Second, for $\beta := \beta_1\beta_2$, we have $(t_{a,1}, t_1^{N_1}, \lambda) \in \tilde{T}^+ \times T_{N,1} \times T^+_{N,2}$, $(t_{b,2}, \lambda, t_2^{N_2}) \in \tilde{T}^+ \times T^+_{N,1} \times T_{N,2}$ and $\#_\beta((t_{a,1}, t_1^{N_1}, \lambda)) = \#_\beta((t_{b,2}, \lambda, t_2^{N_2})) = 1$. Also, it is clear that $M_2 = M_1$. Therefore, the system is not coprognosable according to Theorem 4.1.

V. Conclusion

In this paper, we solve the problem of verifying coprognosability for decentralized labeled Petri nets. We show that this problem is decidable by providing a verifiable necessary and sufficient condition. Our result extends existing results on coprognosability analysis from regular languages to Petri net languages. Note that coprognosability is a condition that determines *a priori* whether or not any fault can be predicted. How to develop efficient online prognostic algorithms for decentralized Petri nets is an important future direction.

References

[1] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre. Fault prognosis of timed stochastic discrete event systems with bounded estimation error. *Automatica*, 82:35–41, 2017.

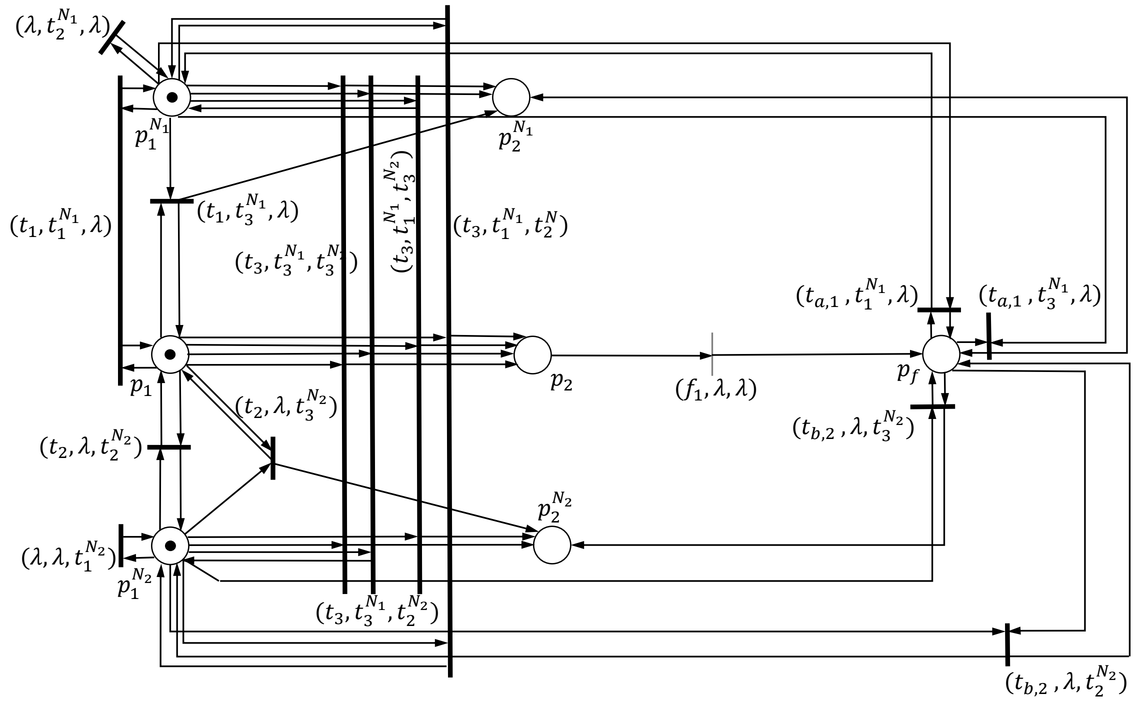

$(\lambda, t_2^{N_1}, \lambda)$
$p_1^{N_1}$
$p_2^{N_1}$
$(t_1, t_3^{N_1}, \lambda)$
$(t_1, t_1^{N_1}, \lambda)$
$(t_3, t_1^{N_1}, t_3^{N_2})$
$(t_3, t_1^{N_1}, t_2^{N})$
$(t_3, t_3^{N_1}, t_3^{N_2})$
$(t_{a,1}, t_1^{N_1}, \lambda)$
$(t_{a,1}, t_3^{N_1}, \lambda)$
p_1
p_2
(f_1, λ, λ)
p_f
$(t_2, \lambda, t_3^{N_2})$
$(t_{b,2}, \lambda, t_3^{N_2})$
$(t_2, \lambda, t_2^{N_2})$
$p_2^{N_2}$
$(\lambda, \lambda, t_1^{N_2})$
$p_1^{N_2}$
$(t_3, t_3^{N_1}, t_2^{N_2})$
$(t_{b,2}, \lambda, t_2^{N_2})$

Fig. 3. $\langle \mathcal{N}_{||}, M_{0,||} \rangle$ for the system in Figure 1(b).

[2] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre. State estimation of discrete event systems for rul prediction issue. *International Journal of Production Research*, 55(23):7040–7057, 2017.

[3] R. Ammour, E. Leclercq, E. Sanlaville, and D. Lefebvre. Faults prognosis using partially observed stochastic petri-nets: an incremental approach. *Discrete Event Dynamic Systems*, pages 1–21, 2018.

[4] M. F. Atig and P. Habermehl. On Yen's path logic for Petri nets. *International Journal of Foundations of Computer Science*, 22(4):783–799, 2011.

[5] B. Benmessahel, M. Touahria, and F. Nouioua. Predictability of fuzzy discrete event systems. *Discrete Event Dynamic Systems*, 27(4):641–673, 2017.

[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] B. Bordbar, A. Al-Ajeli, and M. Alodib. On diagnosis of violations of constraints in Petri net models of discrete event systems. In *26th International Conference on Tools with Artificial Intelligence*, pages 673–680. IEEE, 2014.

[8] 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.

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

[10] M. Chang, W. Dong, Y. Ji, and L. Tong. On fault predictability in stochastic discrete event systems. *Asian journal of Control*, 15(5):1458–1467, 2013.

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

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

[13] 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.

[14] A. Khoumsi. Multi-decision prognosis: Decentralized architectures cooperating for predicting failures in discrete event systems. In *20th Mediterranean Conference on Control & Automation*, pages 309–314. IEEE, 2012.

[15] 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.

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

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

[18] 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.

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

[20] N. Ran, H. Su, A. Giua, and C. Seatzu. Codiagnosability analysis of bounded Petri nets. *IEEE Transactions on Automatic Control*, 2017.

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

[22] S. Takai and R. Kumar. Inference-based decentralized prognosis in discrete event systems. *IEEE Transactions on Automatic Control*, 56(1):165–171, 2011.

[23] 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.

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

[25] 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.

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

[27] X. Yin. Verification of prognosability for labeled petri nets. *IEEE Transactions on Automatic Control*, 2018.

[28] 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.

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

[30] X. Yin and Z. Li. Decentralized fault prognosis of discrete-event systems using state-estimate-based protocols. *IEEE Transactions on Cybernetics*, 2018. DOI: TCYB.2018.2799961.

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

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