IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 15, 2019, accepted December 2, 2019, date of publication December 16, 2019, date of current version December 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2959904

K-Codiagnosability Verification of Labeled Petri Nets

NING RAN[1], (Member, IEEE), JINYUAN HAO[2], ZIJIAN DONG[3], ZHOU HE[4], (Member, IEEE), ZHIHENG LIU[1], YUAN RUAN[5], AND SHOUGUANG WANG[5], (Senior Member, IEEE)

[1]College of Electronic and Information Engineering, Hebei University, Baoding 071000, China
[2]HBU-UCLan School of Media, Communication and Creative Industries, Hebei University, Baoding 071000, China
[3]Department of Automation, North China Electric Power University, Baoding 071003, China
[4]College of Mechanical and Electrical Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China
[5]School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China

Corresponding author: Shouguang Wang (wsg5000@hotmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61903119 and Grant 61803246, in part by the Natural Science Foundation of Hebei Province under Grant F2019201088 and Grant F2019201095, in part by the Hebei Province Foundation for Returned Overseas Chinese Scholars under Grant C20190319, in part by the Hebei University One Province One University Special Fund under Grant 801260201137, and in part by the Foundation of Hebei Education Department under Grant QN2019202.

ABSTRACT Fault detection is quite important for discrete event systems. We investigate K-codiagnosability of Petri nets in this paper under the framework that some local sites monitor the operation of the system using their own masks. They exchange information with a coordinator while do not communicate with each other. A fault is detected when there exists a site can diagnose it. We recall the notion of Modified Verifier Nets (MVNs), and prove that K-codiagnosability can be verified looking at some special cycles in the reachability or coverability graph of the MVN. In particular, the proposed approach is available for bounded and unbounded nets. Finally, we give an algorithm to compute the minimum value of K.

INDEX TERMS Petri nets, fault detection, K-codiagnosability.

I. INTRODUCTION

Any abnormal behavior can be viewed as a fault, and is unavoidable in discrete event systems (DESs). Performing diagnosability analysis of DESs is to confirm whether faults can be diagnosed after finite steps. The diagnosability problem was extensively researched in a centralized framework and a series of approaches has been proposed [1]–[11]. However, a lot of large DESs are usually physically decentralized, and the centralized approaches are not available for this case. Therefore, in recent years some decentralized approaches are developed [12]–[18].

Debouk *et al.* [12] first propose the definition of diagnosability in a decentralized framework and introduce three different protocols, which are the extensions of the results in [1]. Qiu and Kumar [13] first give the notion of codiagnosability under the protocol that some local sites exchange information with a coordinator while do not communicate with each other. A fault is diagnosed if there exists a site is able to detect it.

Petri nets (PNs) are extensively used in the problems of supervisory control [19]–[21], performance optimization [22]–[25] and codiagnosability analysis. Cabasino *et al.* [14] first prove that a PN is codiagnosable, under the same protocol as [13], if and only if it contains arbitrarily long *failure ambiguous sequences (FASs)*. Note that a failure sequence is said to be failure ambiguous if it is faulty for some sites and non-faulty for other sites. A special PN structure, called Modified Verifier Net (MVN), was constructed as the synchronization of given system with the nonfailure part with respect to (w.r.t.) all sites. Therefore, the existence of FASs can be computed using the reachability graph of the MVN. However, this approach may be unfeasible in real situations since its complexity may grow exponentially in the worst case. In order to deal with this limitation, some authors of this paper [15] analyze codiagnosability of bounded PNs taking advantage of *Basis Markings*, which allows ones to look for FASs without enumerating the entire state space. This approach can also be used to analyze K-codiagnosability of bounded PNs. In [16], Basile *et al.* analyze K-codiagnosability dealing with integer linear programming (ILP) problems. Although its complexity is NP-complete, it can be solved by some off-the-shelf tools such as LINGO and GUROBI. In particular, a common assumption of acyclicity of unobservable transitions is not required.

The associate editor coordinating the review of this manuscript and approving it for publication was Francesco Tedesco.

In this paper, we study K-codiagnosability of labeled PN systems (LPNSs) employing the notion of MVN proposed in [14]. We first investigate the relationship between codiagnosability and K-codiagnosability, and show that the approach in [14] is not applicable for the case of K-codiagnosability. Then a necessary and sufficient condition for K-codiagnosability is presented, and some procedures to verify K-codiagnosability are given for both bounded and unbounded LPNSs. Finally, a method to compute the minimum value of K is given.

II. PRELIMINARIES

We recall some basic knowledge in this section. For more details, we refer to [26].

A. LABELED PETRI NETS

A Petri net (PN) is a four-tuple $N = (P, T, F, W)$. P is the set of places and T is the set of transitions, where they are represented by circles and bars respectively. The connected relation between the places and transitions is described by F. The mapping W attaches a nonnegative number to each arc: $W(a, b) > 0$ if $(a, b) \in F$ and $W(a, b) = 0$ if $(a, b) \notin F$, where $a, b \in P \cup T$. The notation $[N]$ is used to denote the *incidence matrix*.

The *postset* of a is indicated as

$$a^{\bullet} = \{b \in P \cup T \mid (a, b) \in F\},$$

and the *preset* of a is indicated as

$$^{\bullet}a = \{b \in P \cup T \mid (b, a) \in F\},$$

where a is a node of $P \cup T$.

A marking m assigns a nonnegative number to each place. A transition t is called *enabled* at the marking m if it holds that

$$\forall p \in {}^{\bullet}t, \quad m(p) \geq W(p, t), \tag{1}$$

and is written as $m[t\rangle$. If a new marking m' is reached by the firing of t, then it is

$$\forall p \in P, \; m'(p) = m(p) + [N](p, t)$$

and written as $m[t\rangle m'$. $L(N, m_0)$ is used to denote the set of transition sequences that are enabled at the initial marking m_0, i.e.,

$$L(N, m_0) = \{\sigma \in T^* \mid m_0[\sigma\rangle\}.$$

In addition, ε denotes the empty sequence. $R(N, m_0)$ is used to denote the set of markings that are reachable from the initial marking m_0, i.e.,

$$R(N, m_0) = \{m \mid m_0[\sigma\rangle m, \sigma \in L(N, m_0)\}.$$

Let $\sigma \in L(N, m_0)$ be a transition sequence and $T' \subseteq T$ be a set of transitions. We denote by $\pi(\sigma)$ the *Parikh vector* of σ. The fact that a transition t is contained in σ is written as $t \in \sigma$. Furthermore, we use $T' \cap \sigma \neq \emptyset$ to denote that there exists at least a transition in T' contained in σ, and use $T' \cap \sigma = \emptyset$ to denote that there exist no transitions in T' contained in σ.

Definition 1 ([5]): Given a reachable marking m_1, a transition sequence $s \in L(N, m_0)$ is called a *repetitive sequence* if it is able to fire infinitely at m_1, i.e., it holds that

$$m_1[s\rangle m_2[s\rangle m_3[s\rangle \ldots \tag{2}$$

where $m_2, m_3, \ldots$ are reachable markings.

Let s be a repetitive sequence that satisfies (2). s is called a *stationary sequence* if it holds that $m_i = m_{i+1}$, called an *increasing sequence* if it holds that $m_i \lneq m_{i+1}$, where $i = 1, 2, \ldots$.

Given a PN N, a subnet $N' = (P, T', F', W)$ is called the *T'-induced subnet* of N if T' is a subset of T, where $F' \in F$ denotes the connected relation. We can compute N' by removing the transitions in $T \setminus T'$.

A *labeled PN system (LPNS)* is a triple $(N, m_0, \mathcal{L})$, where $\mathcal{L}$ is a labeling function $\mathcal{L} : T \to A \cup \{\varepsilon\}$, and A is an alphabet.

We indicate as T_u and T_o the set of *unobservable* transitions and the set of *observable* transitions, respectively.

We extend the function $\mathcal{L}$:

(1) $\mathcal{L}(\varepsilon) = \varepsilon$;

(2) $\mathcal{L}(t) = l$, where $t \in T_o$ and $l \in A$;

(3) $\mathcal{L}(t) = \varepsilon$, where $t \in T_u$;

(4) $\mathcal{L}(\sigma t) = \mathcal{L}(\sigma)\mathcal{L}(t)$, if $\sigma \in T^*$ and $t \in T$.

Furthermore, given an observed sequence $w \in A^*$, the set of sequences that correspond to w is written as $\mathcal{L}^{-1}(w)$, namely, it is

$$\mathcal{L}^{-1}(w) = \{s \in L(N, m_0) \mid \mathcal{L}(s) = w\}.$$

Two transition sequences s and s' producing the same observation, i.e., $\mathcal{L}(s) = \mathcal{L}(s')$, are said to be *undistinguishable*. Otherwise, they are said to be *distinguishable*.

The *post-language* of K after σ is written as K/σ, namely, it is

$$K/\sigma = \{\sigma' \in T^* \mid \sigma\sigma' \in K\},$$

where $K \subseteq T^*$ is a language.

B. COVERABILITY GRAPH

Coverability Graphs (CGs) are used for the analysis of unbounded PNs. Every node is attached with a $|P|$ dimensional vector, in which every entry is either a nonnegative number or a symbol ω that denotes "arbitrarily large". It should be notice that $\omega > n$ and $\omega \pm n = \omega$, where n is a nonnegative number. Every arc is attached with $(t, \mathcal{L}(t))$, where $\mathcal{L}$ is the labeling function. An ω-marking is a marking that contains ω-places.

A *Coverability Tree* is constructed using Algorithm 1, which is detailed in Fig. 1. Then the CG can be obtained by merging identical nodes in the tree.

Remark 1: In the CG, a cycle may correspond either to a non-repetitive sequence or a repetitive sequence. It is able to fire infinitely if it corresponds to a repetitive sequence. On the other hand, if it corresponds to a non-repetitive sequence,

Algorithm 1 [Construction of a Coverability Tree]

Input: A PN system (N, m_0).
Output: A Coverability Tree.

1. Let m_0 be the root node of the tree
2. While nodes with no tag exist
- **2.1** select a node m with no tag
- **2.2 for** all t enabled at m, **do**
  - compute the marking $m' = m + [N](\cdot, t)$
  - **for** all markings $m'' \lneq m'$ on the path from the root node to the node m and for all $p \in P$, **do**
    - **if** $m''(p) < m'(p)$, **then** $m'(p) = \omega$

    end for
  - add a new node m' to the tree
  - add a new arc $(t, \mathcal{L}(t))$ from m to m'
  - **if** there already exists a node m' in the tree, **then** label the new node m' "duplicated"
- **2.3** label node m "old"

  end for

End While

FIGURE 1. Algorithm 1: Construction of a coverability tree.

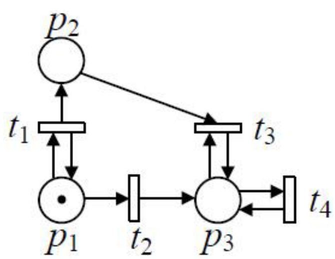


FIGURE 2. An unbounded PN.

it must contains ω-markings (otherwise it corresponds to a repetitive sequence), which means that the PN contains an increasing sequence whose firing can put an indeterminate amount of tokens in ω-places [5]. Hence, the cycle is able to fire until all tokens in those places are consumed. This fact can be illustrated using Example 1.

Example 1: Fig. 2 shows an unbounded PN, where $m_0 = (1\ 0\ 0)^T$. Its CG is shown in Fig. 3. The cycle (sequence) $\sigma_1 = t_4$ is a repetitive sequence since it can fire infinitely starting from $(0\,0\,1)^T$ or from $(0\,\omega\,1)^T$. The cycle $\sigma_2 = t_3$ is a non-repetitive sequence since it cannot fire infinitely starting from any marking. In fact, t_3 can fire at most as many times as t_1 due to the fact that t_3 cannot fire when the tokens in p_2 are consumed.

III. PROBLEM FORMULATION

The set of unobservable transitions T_u is partitioned as follows:

$$T_u = T_f \cup T_{reg},$$

where T_f denotes the set of faults, and T_{reg} denotes the set of faulty-free transitions. Furthermore, the set T_f contains r

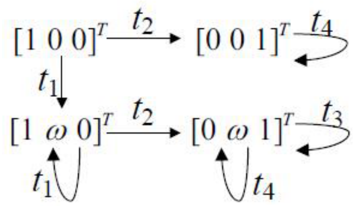


FIGURE 3. CG of the PN of Example. 1.

types of faults, i.e.,

$$T_f = \bigcup_{i=1}^{r} T_f^i,$$

where T_f^i denotes the i-th type of faults.

Let $T' = T \setminus T_f$. $(N', m_0, \mathcal{L}')$ indicates the *nonfailure subnet* of $(N, m_0, \mathcal{L})$, where N' is the T'-induced subnet and $\mathcal{L}'$ is the corresponding labeling function.

In this paper, we perform K-codiagnosability analysis under the protocol that the net is observed by some sites $\mathcal{J} = \{1, 2, \ldots, \nu\}$. They observe the PN's evolution, but have no communication with each other. It should be noted that such a protocol is identical with the one in [15].

We indicate as $T_{o,j} \subseteq T_o$ the set of observable transitions for $j \in \mathcal{J}$, and $T_{u,j} = T \setminus T_{o,j}$ the set of unobservable transitions for $j \in \mathcal{J}$. Every observable transition can be monitored by at least one site, i.e.,

$$T_o = \bigcup_{j \in \mathcal{J}} T_{o,j}.$$

We denote by $A_j \subseteq A$ the alphabet for the j-th site, and

$$\mathcal{L}_j(t) = \begin{cases} \mathcal{L}(t), & \text{if } \mathcal{L}(t) \in A_j \\ \varepsilon, & \text{otherwise} \end{cases} \tag{3}$$

the labeling function with respect to the j-th site.

The following assumption is adopted in this paper:

(A1) The LPNS is not dead after the occurrence of a fault.

This assumption is common in the research of diagnosability, which avoids dealing with the technicality that PNs may be dead after the occurrence of faults.

Hereinafter, $\Psi(T_f^i)$ indicates the set of transition sequences in which the last transition is a fault in T_f^i, where $i = 1, \ldots, r$.

Definition 2: An LPNS $(N, m_0, \mathcal{L})$ that satisfies Assumption A1 is *codiagnosable* w.r.t. T_f^i if

$$(\forall s \in \Psi(T_f^i)), (\exists K \in \mathbb{N}), (\forall \sigma \in L(N, m_0)/s), |\sigma| \geq K$$
$$\Rightarrow (\exists j \in \mathcal{J}), (\forall \sigma' \in \mathcal{L}_j^{-1}(\mathcal{L}_j(s\sigma))), T_f^i \cap \sigma' \neq \emptyset.$$

By Definition 2, an LPNS is codiagnosable w.r.t. T_f^i if there exists a site that is able to diagnose the firing of a fault in T_f^i after a sequence of finite length.

Definition 3: An LPNS $(N, m_0, \mathcal{L})$ that satisfies Assumption A1 is K-*codiagnosable* w.r.t. T_f^i if $\exists K \in \mathbb{N}$ such that

$$(\forall s \in \Psi(T_f^i)), (\forall \sigma \in L(N, m_0)/s), (|\sigma| \geq K)$$
$$\Rightarrow (\exists j \in \mathcal{J}), (\forall \sigma' \in \mathcal{L}_j^{-1}(\mathcal{L}_j(s\sigma))), T_f^i \cap \sigma' \neq \emptyset.$$

By Definition 3, an LPNS is K-codiagnosable w.r.t. T_f^i if there exists a site that is able to diagnose the firing of a fault in T_f^i after at most K transitions. Given a number $K' > K$, an LPNS is obviously K'-codiagnosable if it is K-codiagnosable.

The objective of the paper is to analyze K-codiagnosable of LPNSs under Assumption A1.

IV. MAIN RESULT

This section will show that K-codiagnosability of an LPNS can be analyzed using a structure, called *Modified Verifier Net (MVN)*, which is defined as the synchronization of the LPNS with its nonfailure subnets w.r.t. all local sites, where the composition is performed on the alphabet A. In this section, we assume that there exists a single fault type T_f in the LPNS.

A. MODIFIED VERIFIER NET (MVN)

Definition 4: A sequence $\sigma \in L(N, m_0)$ that satisfies $T_f \cap \sigma \neq \emptyset$ is called a *failure ambiguous sequence (FAS)* if the LPNS contains ν transition sequences $s_1, s_2, \ldots, s_\nu$ that satisfies:

$$T_f \cap s_j = \emptyset \quad \text{and} \quad \mathcal{L}_j(s) = \mathcal{L}_j(s_j),$$

where $j \in \{1, 2, \ldots, \nu\}$.

By Definition 4, a faulty sequence σ is failure ambiguous if it is faulty for some sites and non-faulty for other sites.

It has been proved in [14] that an LPNS is codiagnosable if and only if it contains no FASs with arbitrary length. Therefore, codiagnosability can be studied by finding some cycles in the R/CG of the MVN. For the sake of simplicity, the formal notion and construction algorithm of MVNs are not recalled here.

Given an LPNS, we denote by G its MVN. A node in the R/CG of G is called *faulty* if it can be reached via a path in which there exists at least a fault. The set of all faulty nodes is written as $F(G)$.

Proposition 1 ([14]): Let G be the MVN of an LPNS and $F(G)$ be the set of faulty nodes. The LPNS is codiagnosable *iff* starting from any faulty node, the R/CG of G contains no cycles corresponding to repetitive sequences.

Example 2: Fig. 4 shows an unbounded LPNS, where $T_o = \{t_1, t_3, t_5, t_6, t_8, t_9\}$, $T_u = \{t_2, t_4, t_7\}$, $T_f = \{t_4\}$ and $m_0 = (0\ 1\ 0\ 0\ 0)^T$. Let $\mathcal{L}(t_1) = a$, $\mathcal{L}(t_5) = \mathcal{L}(t_8) = b$, $\mathcal{L}(t_6) = c$, $\mathcal{L}(t_9) = d$ and $\mathcal{L}(t_3) = e$. Assume that the LPNS is locally monitored by two sites, where $A_1 = \{a, b, c, e\}$ and $A_2 = \{a, d, e\}$.

The MVN is constructed using Algorithm 5.1 in [14], and is shown in Fig. 5. Note that in this figure we use a double arrow arc if a place has a self-loop. Then we construct the CG of the MVN, which is not reported here. We can observe that starting from any faulty node, there exist no cycles corresponding to repetitive sequences in the MVN. Therefore, the LPNS is codiagnosable.

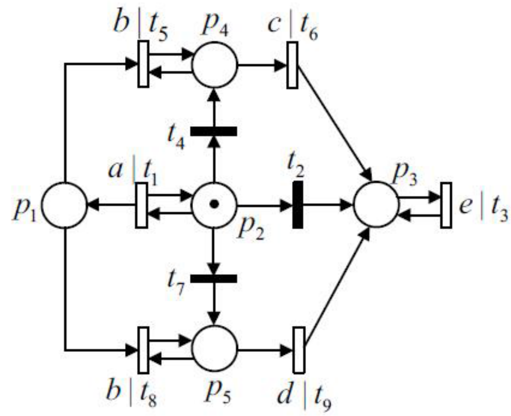


FIGURE 4. An unbounded LPNS.

B. K-CODIAGNOSABILITY VERIFICATION USING MVN

We propose a necessary and sufficient condition for K-codiagnosability in this section.

We first show that the result in [14] is not available for the case of K-codiagnosability. Consider again the unbounded LPNS shown in Fig. 4. We can infer that it is not K-codiagnosable. In fact, for any sequence $s = t_1^x t_4 \in \Psi(T_f)$, we can choose a number $K_s \geq x + 1$ to verify codiagnosability, where x is a nonnegative number. However, since K_s may grow arbitrarily large with x, we cannot find a *fixed* number K so that it is K-codiagnosable.

By Definition 3, we know that the detection delay after each fault has a uniform bound, i.e., K. While by Definition 2, we know that such a uniform bound does not necessarily exist. In particular, the following two properties hold.

Property 1: An LPNS $(N, m_0, \mathcal{L})$ is codiagnosable *if* it is K-codiagnosable.

Proof: The result obviously holds since Definition 3 is stronger than Definition 2. □

Property 2: A bounded LPNS $(N, m_0, \mathcal{L})$ is codiagnosable *iff* it is K-codiagnosable.

Proof: (If) Straightforward from Property 1.

(Only if) Let m_s be the marking reached by firing a sequence $s \in \Psi(T_f)$ from m_0, i.e., $m_0[s\rangle m_s$. By Definition 2, there exists a number, say K_s, such that the fault can be detected after K_s steps starting from m_s. Since the LPNS is bounded, it has a finite number of markings reached by firing sequences in $\Psi(T_f)$ from m_0. Therefore, the LPNS is K-codiagnosable by taking the largest K over all such markings. □

Now we give the main result of this paper.

Proposition 2: Let $(N, m_0, \mathcal{L})$ be an LPNS. Let G be the MVN and $F(G)$ be the set of faulty nodes. The LPNS is K-codiagnosable *iff* staring from any node of $F(G)$, the R/CG of G contains no cycles.

Proof: Let $(N, m_0, \mathcal{L})$ be a bounded LPNS. By Proposition 1, we know that the result obviously holds for this case since in the RG of G, a cycle must correspond to a repetitive sequence.

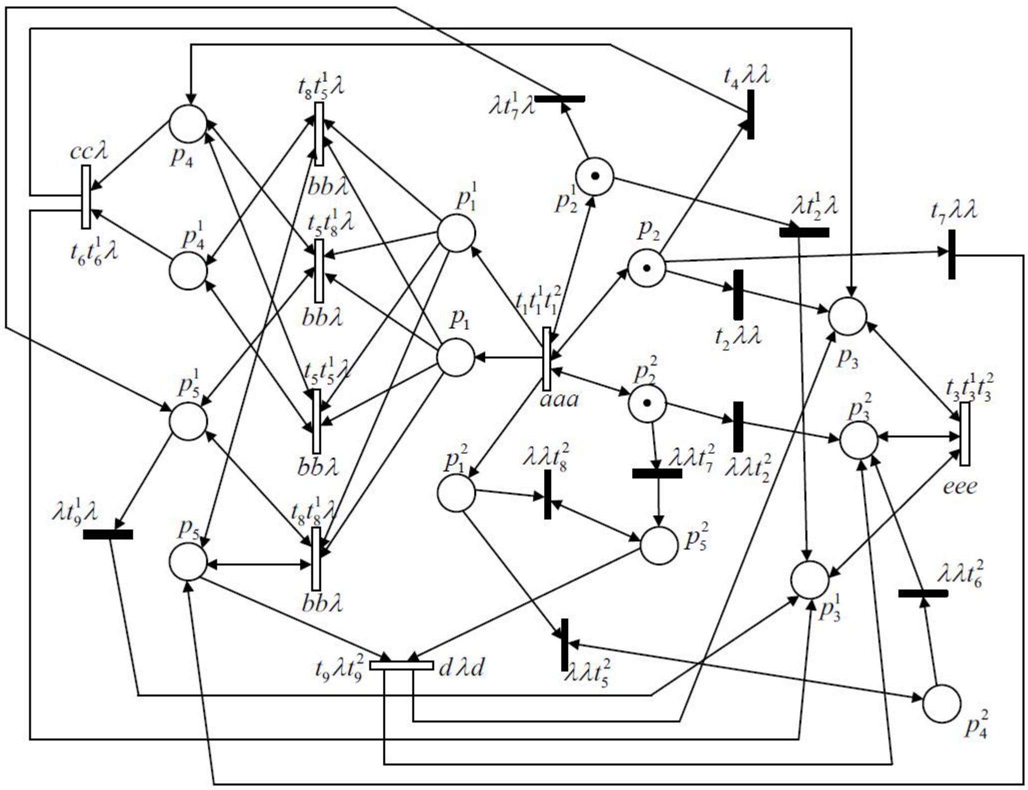


FIGURE 5. The MVN of the LPNS in Fig. 4.

Let $(N, m_0, \mathcal{L})$ be an unbounded LPNS. We prove the if part and only if part respectively.

(If) When constructing the CG of G, we can always compute that after how many transitions the firing of a fault can be diagnosed by a site since there exist no cycles starting from a node in $F(G)$. Therefore, the LPNS is K-codiagnosable, where K is taken as any number that is larger than the longest subsequent path of a faulty node.

(Only if) By contradiction, assume that there exists a cycle after a faulty node. It may correspond to either a repetitive sequence or a non-repetitive sequence:

1) If it corresponds to a repetitive sequence, then $(N, m_0, \mathcal{L})$ is not codiagnosable by Proposition 1. As a consequence, it is not K-codiagnosable by Property 1. This is a contradiction.
2) If it corresponds to a non-repetitive sequence, then it cannot occur arbitrarily. By Remark 1, we know that it contains ω-markings for sure, which means that there exists an increasing sequence whose firing can put an uncertain number of tokens in ω-places. Therefore, the non-repetitive sequence can occur until the tokens in ω-places are consumed. However, we cannot find a fixed number of transitions can occur after a fault since the number of tokens in those places is uncertain. Therefore, it is not K-codiagnosable by Definition 3. This is a contradiction.

□

Algorithm 2, which is detailed in Fig. 6, summarizes the main procedures for verifying K-codiagnosability of LPNSs.

Algorithm 2 [Verification of K-codiagnosability]

Input: An LPNS $(N, m_0, \mathcal{L})$.
Output: K-codiagnosability of $(N, m_0, \mathcal{L})$.
1. Construct the MVN G
2. Compute the R/CG of G
3. Compute the the set $F(G)$ of faulty nodes of G
4. Check if there exists a node of the R/CG in $F(G)$ from which a cycle exists
4.1 **if** the answer is yes, **then**
- output the LPNS is not K-codiagnosable

else
- output the LPNS is K-codiagnosable

end if

FIGURE 6. Algorithm 2: Verification of *K*-codiagnosability.

Example 3: Reconsider the unbounded LPNS shown in Fig. 4. A part of the CG of the MVN is detailed in Fig. 7, where the markings are detailed in Table 1. Note that $m_2 \in F(G)$. According to Proposition 2, we can infer that it is not K-codiagnosable due to the fact that the CG contains cycle $c_1 = bb\lambda/t_5t_8^1\lambda$ after m_2. In fact, the sequence associated to the cycle is not repetitive, and thus it cannot fire for an infinite number of times.

In previous discussion, we only consider one fault type. When the LPNS contains r fault types, r MVNs need to be constructed, i.e., for every fault type a MNV is constructed.

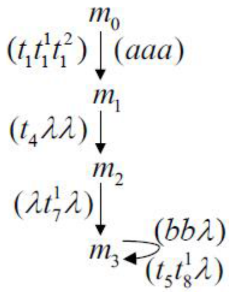


FIGURE 7. A part of the CG of the MVN w.r.t. T_f.

TABLE 1. The set of markings in Fig. 7.

Nodes	Markings
m_0	$(0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0)^T$
m_1	$(\omega\,1\,0\,0\,0\,\omega\,1\,0\,0\,0\,\omega\,1\,0\,0\,0)^T$
m_2	$(\omega\,0\,0\,1\,0\,\omega\,1\,0\,0\,0\,\omega\,1\,0\,0\,0)^T$
m_3	$(\omega\,0\,0\,1\,0\,\omega\,0\,0\,0\,1\,\omega\,1\,0\,0\,0)^T$

If we analyze K-codiagnosability w.r.t. a fault type, all faults that are not in this type should be viewed as faulty-free transitions.

The complexity of Algorithm 2 is briefly discussed here. The most burdensome part of Algorithm 2 consists in the construction of the R/CG. It is well known that constructing a CG is still an open issue since it requires even more than exponential space [27]. In fact, Yin and Lafortune have proved that checking diagnosability of LPNSs is EXPSPACE-complete, and the high complexity seems unavoidable [27]. Fortunately, some efficient software such as Time Petri Nets Analyzer can be used for constructing an R/CG. Moreover, the structure properties for some types of PNs may also improve the computational efficiency of diagnosability analysis, which could be an attractive research area for future study on fault diagnosis.

V. COMPUTATION OF THE MINIMUM VALUE OF K

From Definition 3, we know that an LPNS that is K-codiagnosable is also K'-codiagnosable with $K' > K$. This section presents a method to compute the minimum value of K, denoted by K_{min}, for an LPNS that is K-codiagnosable. In particular, the computation is still based on the R/CG of MVN.

By Proposition 2, we can easily infer that an LPNS is K-codiagnosable if and only if starting from any faulty node, each subsequent path in the R/CG of the MVN must end in a deadlock (otherwise the LPNS is not K-codiagnosable by Proposition 2). Therefore, K_{min} is the largest number of transitions after a faulty node in the R/CG of the MVN, plus one.

Given a faulty node $x \in F(G)$, we denote by $D(x)$ the largest number of transitions starting from x in the R/CG of the MVN. Clearly, if the LPNS is K-codiagnosable, $D(x)$ is finite and it is

$$K_{min} = 1 + \max_{x \in F(G)} D(x).$$

Algorithm 3 [Computation of K_{min}]

Input: An LPNS $(N, m_0, \mathcal{L})$.
Output: K_{min}.
1. Analyse K-codiagnosability of $(N, m_0, \mathcal{L})$ using Algorithm 2
2. If $(N, m_0, \mathcal{L})$ is K-codiagnosable, **then** goto Step 3 **else** exit
3. Let $k = 0$, $i = 1$, $\Gamma = \emptyset$ and $\forall x \in F(G) : D_0(x) = 0$
4. While i, **do**
  4.1 $k = k + 1$
  4.2 for all $x \in F(G)$, **do**
   • **for** all $x' \in F(G)$ s.t. x' can be reached from x in the R/CG of the MVN with $(\gamma, \gamma_1, \gamma_2)$, **do**
    – **if** $\gamma \neq \lambda$, **then** $\alpha = D_{k-1}(x') + 1$
     else $\alpha = D_{k-1}(x')$
     $\Gamma = \Gamma \cup \{\alpha\}$
   end for
   • let $D_k(x) = \max\{D_{k-1}(x), \max_{\alpha \in \Gamma} \alpha\}$
   • let $\Gamma = \emptyset$
  end for.
  4.3 if $\forall x \in F(G)$, $D_k(x) = D_{k-1}(x)$, **then** $i = 0$
 End While
5. Let $K_{min} = 1 + \max_{x \in F(G)} D_k(x)$

FIGURE 8. Algorithm 3: Computation of K_{min}.

Algorithm 3, which is reported in Fig. 8, provides the procedures to compute K_{min}.

The idea behind Algorithm 3 can be detailed as follows. Steps 1 and 2 analyze K-codiagnosability of the given LPNS using Algorithm 2. Step 3 means that the occurrence of a fault can be detected after at least one (i.e., $D_0(x) + 1$) transition. Step 4 iteratively computes the number of transitions starting from all faulty nodes. It stops if such a number w.r.t. each faulty node no longer increases. The value K_{min} is finally computed by Step 5 taking the largest number of transitions w.r.t. all faulty nodes.

Here we consider the case of multiple fault types. When the LPNS contains r fault types, Algorithm 3 needs to be respectively applied r times. The minimum value of K w.r.t. each fault type, say $K_{i,min}$, is obtained, where $i = 1, 2, \ldots, r$. Therefore, the value of K_{min} is

$$K_{min} = \max_{i \in \{1,2,\ldots,r\}} K_{i,min}.$$

VI. CONCLUSION

Fault detection is an essential task of PNs. This paper presents an approach for analysing K-codiagnosability of LPNSs using the R/CG of a special structure called Modifier Verifier Net, which is first introduced in [14]. A necessary and sufficient condition for K-codiagnosability is given. In particular, the presented approach is available for bounded and unbounded PNs. We finally propose an algorithm to compute the minimum value of K for an LPNS that is K-codiagnosable.

As a future plan, we will analyze codiagnosability and K-codiagnosability of unbounded PNs while avoiding computing the entire state space [28].

REFERENCES

[1] F. Lin, "Diagnosability of discrete event systems and its applications," *Discrete Event Dyn. Syst.*, vol. 4, no. 2, pp. 197–212, 1994.

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

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

[4] T.-S. Yoo and S. Lafortune, "Polynomial-time verification of diagnosability of partially observed discrete-event systems," *IEEE Trans. Autom. Control*, vol. 47, no. 9, pp. 1491–1495, Sep. 2002.

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

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

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

[8] M. P. Cabasino, A. Giua, M. Pocci, and C. Seatzu, "Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems," *Control Eng. Pract.*, vol. 19, no. 9, pp. 989–1001, 2011.

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

[10] F. Basile, P. Chiacchio, and G. De Tommasi, "Decentralized K-diagnosability of Petri nets," *IFAC Proc. Volumes*, vol. 45, no. 29, pp. 214–220, 2012.

[11] N. Ran, A. Giua, and C. Seatzu, "Enforcement of diagnosability in labeled Petri nets via optimal sensor selection," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2997–3004, Jul. 2019.

[12] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dyn. Syst.*, vol. 10, nos. 1–2, pp. 33–86, 2000.

[13] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 36, no. 2, pp. 384–395, Mar. 2006.

[14] M. P. Cabasino, A. Giua, A. Paoli, and C. Seatzu, "Decentralized diagnosability analysis of discrete event systems using Petri nets," in *Proc. 18th IFAC World Congr.*, vol. 18, 2011, pp. 6060–6066.

[15] N. Ran, H. Su, A. Giua, and C. Seatzu, "Codiagnosability analysis of bounded Petri nets," *IEEE Trans. Autom. Control*, vol. 63, no. 4, pp. 1192–1199, Apr. 2018.

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

[17] X. Yin and S. Lafortune, "Codiagnosability and coobservability under dynamic observations: Transformation and verification," *Automatica*, vol. 61, pp. 241–252, Nov. 2015.

[18] S. Takai and T. Ushio, "Verification of codiagnosability for discrete event systems modeled by Mealy automata with nondeterministic output functions," *IEEE Trans. Autom. Control*, vol. 57, no. 3, pp. 798–804, Mar. 2012.

[19] S. Wang, D. You, and C. Seatzu, "A novel approach for constraint transformation in Petri nets with uncontrollable transitions," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 8, pp. 1403–1410, Aug. 2018.

[20] S. Wang, D. You, and M. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S 4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017.

[21] X. Guo, S. Wang, D. You, Z. Li, and X. Jiang, "A siphon-based deadlock prevention strategy for S^3 PR," *IEEE Access*, vol. 7, pp. 86863–86873, 2019.

[22] Z. He, M. Liu, N. Ran, and Z. Li, "Firing rate optimization of deterministic timed event graphs by server performance improvement," *IEEE Access*, vol. 6, pp. 70866–70873, 2018.

[23] Z. He, Z. W. Li, and A. Giua, "Performance optimization for timed weighted marked graphs under infinite server semantics," *IEEE Trans. Autom. Control*, vol. 63, no. 8, pp. 2573–2580, Aug. 2018.

[24] Z. He, Z. Li, and A. Giua, "Cycle time optimization of deterministic timed weighted marked graphs by transformation," *IEEE Trans. Control Syst. Technol.*, vol. 25, no. 4, pp. 1318–1330, Jul. 2017.

[25] N. Ran, S. G. Wang, and W. H. Wu, "Event feedback supervision for a class of Petri nets with unobservable transitions," *IEEE Access*, vol. 6, pp. 6920–6926, 2018.

[26] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.

[27] X. Yin and S. Lafortune, "On the decidability and complexity of diagnosability for labeled Petri nets," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5931–5938, Nov. 2018.

[28] E. Lefaucheux, A. Giua, and C. Seatzu, "Basis coverability graph for partially observable Petri nets with application to diagnosability analysis," in *Proc. 39th Int. Conf. Appl. Theory Petri Nets Concurrency*. Cham, Switzerland: Springer, 2018, pp. 164–183.

NING RAN (S'16–M'18) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2017.

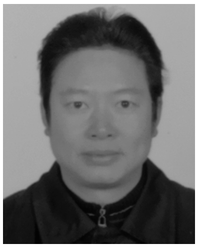
He was a Visiting Ph.D. Student with the Electrical and Electronic Engineering Department, University of Cagliari, Cagliari, Italy, from 2015 to 2016. He joined Hebei University, in 2017, where he is currently a Lecturer with the College of Electronic and Information Engineering. His current research interests include fault diagnosis of Petri nets and supervisory control of discrete event systems.

JINYUAN HAO received the B.A. degree in radio and television journalism from Hebei University, China, in 2013, and the M.A. degree in journalism and communication from the University of New South Wales, Sydney, Australia, in 2015. She joined Hebei University, in 2015, where she is currently a Lecturer with the HBU-UCLan School of Media, Communication and Creative Industries. Her current research interests include Petri nets and new media communication.

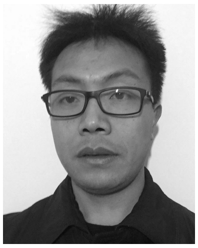
ZIJIAN DONG received the Ph.D. degree in control theory and control engineering from North China Electric Power University, Beijing, China, in 2015. He joined North China Electric Power University, in 2015, where he is currently a Lecturer with the College of Control and Computer Engineering. His current research interests include networked control systems, thermal system modeling, and performance analysis.

ZHOU HE (S'16–M'18) received the B.S. degree in mechanical engineering from the Shaanxi University of Science and Technology, Xi'an, China, in 2012, the Ph.D. degree in mechatronic engineering from Xidian University, Xi'an, in 2017, and the Ph.D. degree in automatique from Aix-Marseille University, Marseille, France, in 2017. He is currently a Lecturer with the Shaanxi University of Science and Technology. His current research interests include Petri nets, fault diagnosis, and control and optimization of timed discrete event systems.

ZHIHENG LIU received the Ph.D. degree in electric machines and electric apparatus from the Dalian University of Technology, in 2017. He joined Hebei University, in 2018, where he is currently a Lecturer with the College of Electronic and Information Engineering. His current research activities are intelligent high-voltage apparatus and intelligent testing technology, and emphasis on electrical equipment on-line detection and insulation diagnosis, and electromagnetic immunity testing technology.



YUAN RUAN received the B.S. and M.A. degrees in electrical engineering from Zhejiang University, Hangzhou, China, in 1989 and 1992, respectively. He joined Zhejiang Gongshang University, in 1992, where he is currently an Associate Professor with the School of Information and Electronic Engineering. His current research interest includes embedded systems.



SHOUGUANG WANG (M'10–SM'12) received the B.S. degree in computer science from the Changsha University of Science and Technology, Changsha, China, in 2000, and the Ph.D. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2005.

He joined Zhejiang Gongshang University, in 2005, where he is currently a Professor with the School of Information and Electronic Engineering, the Director of the Discrete-Event Systems Group, and the Dean of the System modeling and Control Research Institute. He was a Visiting Professor with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, from January 2011 to January 2012. He was the Dean of the Department of Measuring and Control Technology and Instrument, from July 2011 to July 2014. He was a Visiting Professor with the Electrical and Electronic Engineering Department, University of Cagliari, Cagliari, Italy, from December 2014 to December 2015. He has over ten articles in the IEEE Transactions. His current research interests include Petri net theory and application and supervisory control of discrete event systems. He is currently an Associate Editor of IEEE Access and the IEEE/CAA Journal of Automatica Sinica.