Codiagnosability Verification of Bounded Petri Nets Using Basis Markings

Ning Ran [a], Hongye Su [a], Alessandro Giua [b,c], Carla Seatzu [c]

[a] *Institute of Cyber-Systems and Control, Zhejiang University, China. (ranning87@hotmail.com, hysu@iipc.zju.edu.cn)*

[b] *Aix Marseille Univ, Université de Toulon, CNRS, ENSAM, LSIS, Marseille, France (giua@diee.unica.it)*

[c] *DIEE, University of Cagliari, Italy (seatzu@diee.unica.it)*

Abstract

In this paper we propose a novel approach to perform codiagnosability analysis of bounded Petri nets with arbitrary labeling functions. In more detail, a set of sites observe the system evolution, each one with its own observation mask. Sites do not exchange information with each other but communicate with a coordinator. The coordinator is able to detect a fault if and only if at least one site is able to do that. The proposed approach is based on a necessary and sufficient condition for codiagnosability, namely the absence of sequences that are "ambiguous" with respect to all sites and whose length may grow indefinitely after the occurrence of some fault (i.e., sequences of infinite length that could be observed either in the presence of a fault and with no fault). The novelties of the approach consist in using the notion of basis markings to avoid exhaustive enumeration of the set of reachable markings, and in the construction of an automaton, called Verifier, that enables to detect the presence of ambiguous sequences.

Published as:
Ning Ran, Hongye Su, Alessandro Giua, Carla Seatzu, "Codiagnosability Verification of Bounded Petri Nets Using Basis Markings," *2016 IEEE 55th Conference on Decision and Control (CDC)*, 12-14 Dec. 2016, Las Vegas, NV, USA. DOI: 10.1109/CDC.2016.7798866.

1 Introduction

In the fault detection framework two different problems need to be solved: the problem of diagnosis and the problem of diagnosability. Solving a diagnosability problem consists in determining if, once a fault has occurred, the occurrence of the fault can be detected in a finite number of steps. In the past few decades, the diagnosability problem has been extensively studied in a centralized setting [2, 5, 8–10, 14, 17–19]. Since most large complex systems are physically distributed, centralized fault diagnosis methods may not be appropriate in practice, and decentralized diagnosis techniques are needed. In recent years, a series of decentralized approaches have been developed both in automata and Petri nets frameworks [1, 3, 6, 7, 11, 13, 16].

Debouk *et al.* [6] propose a coordinated decentralized architecture consisting of local sites communicating with a coordinator that is responsible for diagnosing the failures occurring in the system. The definition of diagnosability, which was originally introduced in [14] for centralized systems, is extended to the proposed decentralized architecture. Three protocols that realize the proposed architecture are specified, and their diagnostic properties are analyzed.

The notion of codiagnosability has been first introduced in [13] under the assumption that all local diagnosers do not communicate with each other and only send information to a coordinator. Algorithms with polynomial complexity in the size of the automaton and the nonfaulty specification are provided for verifying codiagnosability and computing the bound in the delay of diagnosis.

Due to the intuitive graphical representation and powerful algebraic formulation, Petri nets have been recently used in decentralized diagnosis. Cabasino *et al.* [3] present a procedure to analyze the diagnosability of a Petri net system in a decentralized framework. They first prove that the absence of failure ambiguous sequences is a necessary and sufficient condition for codiagnosability, and give a procedure to verify the absence of such kind of sequences for both bounded and unbounded Petri net systems. The verification is based on the analysis of the reachability/coverability graph of a particular Petri net called *Modified Verifier Net*, which is an extension of the *Verifier Net* introduced in [2] to analyze diagnosability in a centralized setting. However, the number of reachable markings may increase exponentially with the size of the net (structure and number of tokens in the initial marking) thus such an approach could be unfeasible in practical situations.

To address the state explosion problem, this paper uses the notion of basis marking [4] to analyze codiagnosability, thus avoiding exhaustive enumeration of the state space. An automaton called *Verifier* is constructed making the parallel composition of $\nu + 1$ graphs whose number of nodes is a subset of the set of reachable markings, where ν is the number of local sites. The Verifier enables us to detect faulty sequences that lead to the same observation of non-faulty sequences with respect to all sites. If such sequences may have infinite length after the fault, then the system is not codiagnosable. Such sequences correspond to special cycles, called F-cycles, in the Verifier. Thus the problem of codiagnosability analysis is reduced to the problem of looking for F-cycles in the Verifier.

2 Background on labeled Petri nets

In this section, basic definitions of Petri nets are reviewed. For more details we refer the reader to [5] and [12].

A Petri net (PN) is a 4-tuple $N = (P, T, F, W)$, where P and T are finite, non-empty, and disjoint sets, $F \subseteq (P \times T) \cup (T \times P)$ is called the flow relation of the net, W is a mapping that assigns a weight to an arc: $W(x, y) > 0$ iff $(x, y) \in F$, and $W(x, y) = 0$ otherwise, where $x, y \in P \cup T$. The incidence matrix $[N]$ of N is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. Let $x \in P \cup T$ be a node of net N. The preset of x is defined as ${}^\bullet x = \{y \in P \cup T | (y, x) \in F\}$ while the postset of x is defined as $x^\bullet = \{y \in P \cup T | (x, y) \in F\}$.

A marking m of a PN N is a mapping from P to $\mathbb{N} = 0, 1, 2, ...$: $m(p)$ denotes the number of tokens in

place p. (N, m_0) denotes a PN system with an initial marking m_0.

A transition t is enabled at a marking m if $\forall p \in {}^\bullet t, m(p) \geq W(p,t)$. This fact is denoted by $m[t\rangle$ while $m[\sigma\rangle$ is used to denote that the transition sequence $\sigma = t_1 t_2 ... t_k$ is enabled at m. The Parikh vector of σ is denoted by $\pi(\sigma)$. The set of all sequences that are enabled at the initial marking m_0 is denoted by $L(N, m_0)$, i.e., $L(N, m_0) = \{\sigma \in T^* | m_0[\sigma\rangle\}$. We write $t \in \sigma$ to denote that a transition t is contained in σ, $T' \cap \sigma \neq \emptyset$ to denote that there is at least one transition in T' contained in σ and $T' \cap \sigma = \emptyset$ to denote that there is no transition in T' contained in σ, where T' is a set of transitions.

Firing t yields a new marking m' such that $\forall p \in P, m'(p) = m(p) + [N](p,t)$, which is denoted by $m[t\rangle m'$. Marking m'' is said to be reachable from m if there exists a transition sequence σ such that $m[\sigma\rangle m''$. The set of markings reachable from m in N is called the reachability set of (N, m) and is denoted by $R(N, m)$.

A PN is said to be bounded if there exists a positive constant k such that $\forall p \in P$, $m(p) \leq k$, where $m \in R(N, m_0)$. It is unbounded if it is not bounded.

Given a PN system (N, m_0), a transition $t \in T$ is live under m_0 if $\forall m \in R(N, m_0)$, $\exists m' \in R(N, m)$, $m'[t\rangle$. A PN system (N, m_0) is: *live* if $\forall t \in T$, t is live under m_0; *dead* under m_0 if $\nexists t \in T$, $m_0[t\rangle$; *deadlock-free* if $\forall m \in R(N, m_0)$, $\exists t \in T$, $m[t\rangle$.

Given a PN $N = (P, T, F, W)$ and a set $T' \subseteq T$ of transitions, we define T'-induced subnet of N the new PN $N' = (P, T', F', W)$, where F' is the restriction of F to $(P \times T') \cup (T' \times P)$. The net N' can be obtained from N by removing all transitions in $T \setminus T'$.

A Petri net with no directed circuits is said to be acyclic.

A labeled PN system is a triple $(N, m_0, \mathcal{L})$, where (N, m_0) is a PN system, $\mathcal{L}$ is a labeling function $\mathcal{L} : T \to A \cup \{\varepsilon\}$ that assigns to each transition in T either a symbol from a given alphabet A or the empty sequence ε.

We use T_u to denote the set of transitions whose labels are ε, and T_o to denote the set of transitions whose labels are the symbols in A. T_u and T_o are called the set of unobservable and observable transitions, respectively. $[N]_u$ (or $[N]_o$) is used to denote the restriction of the incidence matrix $[N]$ to T_u (or T_o). Given $\sigma \in T^*$, we denote $P_u(\sigma)$ (or $P_o(\sigma)$) the projection of σ over T_u (or T_o).

The labeling function is extended to define the projection operator $\mathcal{L} : T^* \to A^*$ as follows:

1) $\mathcal{L}(t) = l$ for some $l \in A$, if $t \in T_o$;

2) $\mathcal{L}(t) = \varepsilon$, if $t \in T_u$; and

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

Moreover, $\mathcal{L}^{-1}(w)$ is used to denote the set of all transition sequences consistent with $w \in L^*$, i.e., $\mathcal{L}^{-1}(w) = \{\sigma \in L(N, m_0) | \mathcal{L}(\sigma) = w\}$. Using the extended labeling function, the language of transition labels is therefore denoted by $\mathcal{L}(L(N, m_0))$.

Let $K \subseteq T^*$ be a language, we use K/σ to denote the post-language of K after σ, i.e., $K/\sigma = \{\sigma' \in T^* | \ \sigma\sigma' \in K\}$.

3 Problem statement

The unobservable transition set is partitioned as $T_u = T_f \cup T_{reg}$, where T_f is the set of fault transitions and T_{reg} is the set of unobservable but regular transitions. We use $[N]_{reg}$ to denote the restriction of the incidence matrix to T_{reg}. The fault transition set T_f is partitioned into r different subsets T_f^i that model different fault classes, where $i = 1, 2, ..., r$.

The PN is monitored by a set $\mathcal{J} = \{1, 2, ..., \nu\}$ of sites. Each site knows the structure of the net and observes the evolution of the system by its own mask. Sites may send information to a coordinator but do not communicate with the other sites. In particular we assume that the coordinator follows protocol 3 in [6], i.e., a fault in a given class is diagnosed if and only if at least one local site detects its occurrence.

The set of transitions that are observable (or unobservable) for site $j \in \mathcal{J}$ is denoted by $T_{o,j} \subseteq T_o$ (or $T_{u,j} \subseteq T$). The alphabet of the j-th site is denoted A_j, and

$$\mathcal{L}_j(t) = \begin{cases} \mathcal{L}(t), \text{ if } \mathcal{L}(t) \in A_j \\ \varepsilon, \text{ otherwise} \end{cases} \tag{1}$$

is the labeling function associated with the j-th site. Given a transition sequence $\sigma \in L(N, m_0)$, $w_j = \mathcal{L}_j(\sigma)$ is used to denote the sequence of labels in A_j associated with σ by the j-th site.

We make the following assumptions that are commonly adopted in the field of decentralized diagnosability.

A1) The PN system is deadlock-free after the occurrence of any fault;

A2) The PN system is diagnosable in a centralized setting;

A3) The PN system net is bounded;

A4) The $T_{u,j}$-induced subnet is acyclic for any $j \in \mathcal{J}$.

We use $\Psi(T_f^i)$ to denote the set of all sequences in $L(N, m_0)$ that end with a transition in T_f^i.

Definition 1 *Let $(N, m_0, \mathcal{L})$ be a labeled PN system that is deadlock-free after the occurrence of any fault $t_f \in T_f$. Assume that $(N, m_0, \mathcal{L})$ is monitored by a set $\mathcal{J} = \{1, 2, ..., \nu\}$ of local sites. The labeled PN system $(N, m_0, \mathcal{L})$ is* codiagnosable *wrt the i-th fault class T_f^i if*

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

The labeled PN system $(N, m_0, \mathcal{L})$ is codiagnosable *if it is codiagnosable wrt all fault classes.*

In simple words, $(N, m_0, \mathcal{L})$ is codiagnosable wrt T_f^i if, once a fault in T_f^i has occurred, there exists at least one site that detects it within a finite delay.

Let us now recall the definition of *failure ambiguous sequence* first proposed in the Petri net framework by Cabasino *et al.* in [3].

Definition 2 *Consider a labeled PN system $(N, m_0, \mathcal{L})$ whose labeling function $\mathcal{L}$ is defined over an alphabet A. Assume that $(N, m_0, \mathcal{L})$ is monitored by a set $\mathcal{J} = \{1, 2, ..., \nu\}$ of local sites. A sequence $\sigma \in T^*$ such that $T_f^i \cap \sigma \neq \emptyset$ is said to be* failure ambiguous *wrt T_f^i if there exist ν sequences $\sigma_1, \sigma_2, ..., \sigma_\nu \in T^*$, not necessarily distinct, such that*

(1) $\forall \sigma' \in \mathcal{L}^{-1}(\mathcal{L}(\sigma))$, $T_f^i \cap \sigma' \neq \emptyset$; and

(2) $T_f^i \cap \sigma_j = \emptyset$ and $\mathcal{L}_j(\sigma) = \mathcal{L}_j(\sigma_j)$, $j = 1, 2, ..., \nu$.

In other words, a sequence σ containing some fault transitions in T_f^i is failure ambiguous wrt T_f^i if σ is not ambiguous for the centralized diagnoser but it is ambiguous for all sites.

Example 1 *Consider the labeled PN system $(N, m_0, \mathcal{L})$ in Fig. 1, where $T_o = \{t_3, t_6, t_{10}\}$, $T_u = \{t_1, t_2, t_4, t_5, t_7 - t_9\}$, $T_f = \{t_9\}$, $m_0 = [k\ 0\ 0\ 0\ 0\ 0\ 0\ 0]^T$, and k is a positive constant. The labeling function is defined as follows: $\mathcal{L}(t_3) = a$, $\mathcal{L}(t_6) = b$ and $\mathcal{L}(t_{10}) = c$. Assume that the PN is monitored by two local sites whose alphabets are equal to $A_1 = \{a, c\}$ and $A_2 = \{b, c\}$, respectively. The sequence $\sigma = t_7 t_8 t_9 t_{10}$ is failure ambiguous wrt T_f. In fact, $\mathcal{L}^{-1}(\mathcal{L}(\sigma)) = \{t_7 t_8 t_9 t_{10}\}$ and there exist two sequence $\sigma_1 = t_4 t_5 t_6 t_{10}$ and $\sigma_2 = t_1 t_2 t_3 t_{10}$ such that $\mathcal{L}_1(\sigma) = \mathcal{L}_1(\sigma_1) = c$ and $\mathcal{L}_2(\sigma) = \mathcal{L}_2(\sigma_2) = c$.*

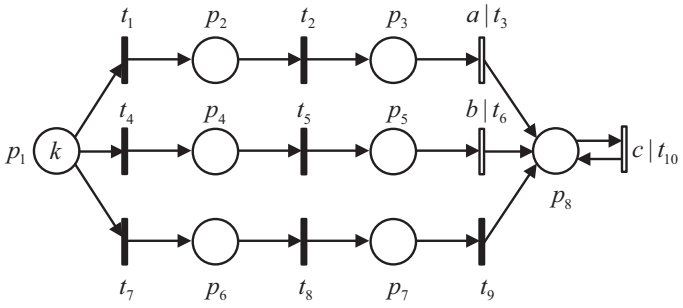

Fig. 1. A Labeled PN system $(N, m_0, \mathcal{L})$.

Theorem 1 *Consider a labeled PN system $(N, m_0, \mathcal{L})$ whose labeling function $\mathcal{L}$ is defined over an alphabet A. Assume that $(N, m_0, \mathcal{L})$ is monitored by a set $\mathcal{J} = \{1, 2, ..., \nu\}$ of local sites. The system is codiagnosable iff there do not exist failure ambiguous sequences that are arbitrarily long after the occurrence of any fault in T_f^i, for $i = 1, \ldots, r$.*

4 Extended Basis Reachability Graph

In this section we first introduce a particular graph, called *Extended Basis Reachability Graph*. Then, we prove some properties that are the starting point for the proposed approach of codiagnosability analysis.

For the sake of simplicity, in the rest of the paper we assume that there is a single fault class T_f. In the following section it is clearly discussed how to deal with the case of several fault classes.

Definition 3 *Given a marking m and an observable transition t, the set of* explanations *of t at m is denoted by*

$$\Sigma(m, t) = \{\sigma \in T_u^* | m[\sigma > m', m'[t >\},$$

and the set of e-vectors *is denoted by*

$$Y(m, t) = \pi(\Sigma(m, t)).$$

Definition 4 *Given a marking m and an observable transition t, the set of* minimal explanations *of t at m is denoted by*

$$\Sigma_{min}(m, t) = \{\sigma \in \Sigma(m, t) | \nexists \sigma' \in \Sigma(m, t) : \pi(\sigma') \lneq \pi(\sigma)\},$$

and the set of minimal e-vectors *is denoted by*

$$Y_{min}(m, t) = \pi(\Sigma_{min}(m, t)).$$

Definition 5 *Let $(N, m_0, \mathcal{L})$ be a labeled PN system and $w \in L^*$ be an observation, where $N = (P, T, F, W)$ and $T = T_o \cup T_u$. The set of pairs ($\sigma_o \in T_o^*$ with $\mathcal{L}(\sigma_o) = w$ and the justification) is denoted by*

$$\hat{\mathcal{J}}(w) = \{(\sigma_o, \sigma_u), \sigma_o \in T_o^*, \mathcal{L}(\sigma_o) = w, \sigma_u \in T_u^* |$$

$$[\exists \sigma \in \mathcal{L}^{-1}(w) : \sigma_o = P_o(\sigma), \sigma_u = P_u(\sigma)]$$
$$\wedge [\nexists \sigma' \in \mathcal{L}^{-1}(w) : \sigma_o = P_o(\sigma'), \sigma'_u = P_u(\sigma')$$
$$\wedge \pi(\sigma'_u) \lneq \pi(\sigma_u)]\},$$

and the set of pairs ($\sigma_o \in T_o^*$ with $\mathcal{L}(\sigma_o) = w$ and the j-vector) is denoted by

$$\hat{Y}_{min}(m_0, w) = \{(\sigma_o, y), \sigma_o \in T_o^*, \mathcal{L}(\sigma_o) = w, y \in \mathbb{N}^{|T_u|} | \exists (\sigma_o, \sigma_u) \in \hat{\mathcal{J}}(w) : \pi(\sigma_u) = y\}.$$

Definition 6 *Let $(N, m_0, \mathcal{L})$ be a labeled PN system, $w \in L^*$ be an observation and $\hat{\mathcal{J}}(w)$ be a set of pairs. The set of* basis markings *of w is denoted by*

$$M_b(w) = \{m \in \mathbb{N}^{|P|} | m = m_0 + [N]_u \cdot \pi(\sigma_u) + [N]_o \cdot \pi(\sigma_o), (\sigma_o, \sigma_u) \in \hat{\mathcal{J}}(w)\},$$

and the set of all basis markings are denoted by M_b, i.e.,

$$M_b = \bigcup_{w \in L^*} M_b(w).$$

In simple words, a basis marking is a marking that can be reached from the initial marking firing a sequence of transitions that is consistent with the observation and a sequence of unobservable transitions, interleaved with the previous sequence, whose firing is strictly necessary to enable it (in the sense that its firing vector is minimal) [4]. The set of basis markings is a subset (usually a strict subset) of the set of reachable markings. Therefore, if the net is bounded, the set of basis markings is finite.

In [5] it has been proved that when performing centralized diagnosability, it is useful to compute basis markings assuming that fault transitions are observable.

Definition 7 *An* extended basis marking (EBM) *is a basis marking computed assuming that all transitions in T_f are observable. The set of all EBMs is denoted by M_e.*

The set M_e can be computed by restricting the minimal explanations to the set of regular unobservable transitions T_{reg}. In the following, we denote $Y_{min}^{reg}(m, t)$ the set of minimal e-vectors restricted to T_{reg}. The set $Y_{min}^{reg}(m, t)$ can be computed using Algorithm 4.4 in [4].

Example 2 *Let us consider the labeled PN system in Fig. 1 previously introduced in Example 1. The set of EBMs is $\{m_i | m_i = [k - i \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ i]^T, i = 0, 1, ..., k\}$.*

Let us now define a graph whose nodes are uniquely associated with EBMs and edges are labeled with either observable transitions (and their labels) or with fault transitions.

Definition 8 *Let $(N, m_0, \mathcal{L})$ be a labeled PN system, T_f be the set of fault transitions and M_e be the set of EBMs. The* Extended Basis Reachability Graph (EBRG) *is a (non-deterministic) finite state automaton $G_e = (M_e, E, \Delta, m_0)$, where M_e is the set of states; $E \subseteq (T_o \times A) \cup T_f$ is the set of event labels; $\Delta \subseteq M_e \times E \times M_e$ is the transition relation; and m_0 is the initial state. In particular, $(m, e, m') \in \Delta$ where $e = t(a) \in T_o \times A$ or $e = t \in T_f$, if and only if $\exists y \in Y_{min}^{reg}(m, t)$ and $m' = m + [N]_{reg} \cdot y + [N](\cdot, t)$.*

Note that a similar graph, called *Modified Basis Reachability Graph* (MBRG) has been proposed in [5] to perform centralized diagnosis. In the MBRG, as well as in the EBRG, a different node is associated with each extended basis marking and edges are labeled either with an observable transition (and its label) or with a fault transition. However, the MBRG contains some information on nodes and edges that are omitted in the EBRG. This implies that the number of edges of the EBRG is a subset of the number of edges of the MBRG. In more detail, if there exist two minimal explanations of a given transition that lead to the same extended basis marking, in the MBRG two different edges are associated with it, while only one edge appears in the EBRG.

Algorithm 1 summarizes the main steps for the construction of the EBRG.

Algorithm 1: [EBRG construction]

Input: A labeled PN system $(N, m_0, \mathcal{L})$.

Output: The EBRG G_e.

1. Let m_0 be the initial node.

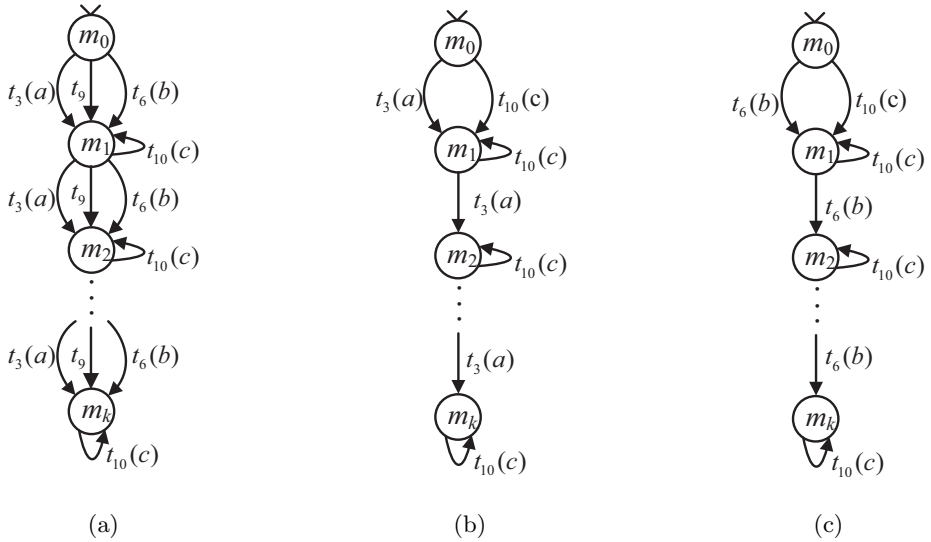

Fig. 2. a) G_e: EBRG of $(N, m_0, \mathcal{L})$, b) G_e^1: Nonfailure-EBRG wrt site 1, and c) G_e^2: Nonfailure-EBRG wrt site 2.

2. While nodes with no tag exist

2.1. select a node m with no tag,

2.2. for all $t \in T_o \cup T_f$, do

2.2.1. if $Y_{min}^{reg}(m,t) \neq \emptyset$, then

- for all $y \in Y_{min}^{reg}(m,t)$, do
  - let $m' = m + [N]_{reg} \cdot y + [N](\cdot, t)$,
  - if $\nexists$ a node m', then
    - add a node m',
  - if $t \in T_o \wedge \nexists$ an arc $t(e)$ from m to m', where

  $e = \mathcal{L}(t)$, then
    - add an arc $t(e)$ from m to m',
  - if $t \in T_f \wedge \nexists$ an arc t from m to m', then
    - add an arc t from m to m',

2.3. tag the node m "old".

3. Remove all tags.

Example 3 *Consider again the labeled PN system in Example 1. The EBRG G_e is shown in Fig. 2a, where* $\{m_i | m_i = [k-i \ 0 \ 0 \ 0 \ 0 \ 0 \ i]^T, i = 0, 1, ..., k\}$.

Property 1 *Let $(N, m_0, \mathcal{L})$ be a labeled PN system, G_e be its EBRG and $L(G_e)$ be the language generated by G_e. It holds that*

$$\rho(L(N, m_0)) = L(G_e)$$

where $\rho(L(N, m_0))$ is the projection of $L(N, m_0)$ over $T_o \cup \{T_f\}$.

Proof. *See [Arxiv paper].*

In words, the above property claims that set of transition sequences in $L(G_e)$ coincides with the projection of $L(N, m_0)$ over the set $T_o \cup T_f$.

5 Verifier

In this section, we show that the codiagnosability of a bounded PN can be verified by analyzing a special automaton called *Verifier*. For the sake of simplicity, and without loss of generality, we assume that the PN is monitored by only two local sites.

In the following we denote by $(N', m_0, \mathcal{L}')$ the T'-induced subnet of $(N, m_0, \mathcal{L})$, where $T' = T \setminus T_f$, i.e., $(N', m_0, \mathcal{L}')$ is the nonfailure subnet of $(N, m_0, \mathcal{L})$. Therefore, $L(N', m_0)$ is the language formed with all sequences of $L(N, m_0)$ that do not contain faults, and $\mathcal{L}'$ is equal to $\mathcal{L}$ restricted to $T \setminus T_f$.

Definition 9 *Let $(N, m_0, \mathcal{L})$ be a labeled PN system and $(N', m_0, \mathcal{L}')$ be its nonfailure subnet. The* nonfailure-EBRG *wrt site j, denoted by $G_e^j = (M^j, E^j, \Delta^j, m_0)$, is the EBRG of $(N', m_0, \mathcal{L}')$ constructed under the assumption that the set of observable transitions is equal to $T_{o,j}$, and all transitions in $T' \setminus T_{o,j} = T \setminus T_f \setminus T_{o,j}$ are unobservable.*

Obviously, G_e^j can be computed using Algorithm 1 assuming that the set of observable transitions is equal to the set of transitions observable by the j-th site, namely $T_{o,j}$, and restricting minimal explanations to the set $T' \setminus T_{o,j} = T \setminus T_f \setminus T_{o,j}$.

Example 4 *Consider again the Petri net in Example 1. The nonfailure-EBRGs G_e^1 and G_e^2 are shown in Figs. 2b and 2c, respectively, where $\{m_i | m_i = [k-i \ 0 \ 0 \ 0 \ 0 \ 0 \ i]^T, i = 0, 1, ..., k\}$.*

Property 2 *Let $(N, m_0, \mathcal{L})$ be a labeled PN system, G_e^j be its nonfailure-EBRG wrt site j and $L(G_e^j)$ be the language generated by G_e^j. It holds that*

$$\rho_j(L(N', m_0)) = L(G_e^j)$$

where $\rho_j(L(N', m_0))$ is the projection of $L(N', m_0)$ over $T_{o,j}$.

Proof. *See [Arxiv paper].*

We now introduce a (non-deterministic) finite state automaton, called *Verifier*, that is defined as the parallel composition of the EBRG of a given labeled PN system and the nonfailure-EBRGs G_e^1 and G_e^2 of the two sites that monitor it, where synchronization is performed on the set of labels A. We denote it $V = (M^V, E^V, \Delta^V, m_0^V)$ and compute it using the following algorithm.

Algorithm 2: [Construction of the Verifier]

Input: G_e, G_e^1 and G_e^2.

Output: The Verifier $V = (M^V, E^V, \Delta^V, m_0^V)$.

1. Let $M^V = M \times \{\text{F,N}\} \times M^1 \times M^2$.

2. Let $E^V = (T_o \cup T_f \cup \{\varepsilon\}) \times (T_{o,1} \cup \{\varepsilon\}) \times (T_{o,2} \cup \{\varepsilon\})$.

3. Let $m_0^V = (m_0, N; m_0; m_0)$.

4. $\Delta^V \subseteq M^V \times E^V \times M^V$ is defined as follows:

- $((m, l; m_1, m_2), t\varepsilon\varepsilon, (m', \text{F}; m_1; m_2)) \in \Delta^V$ if

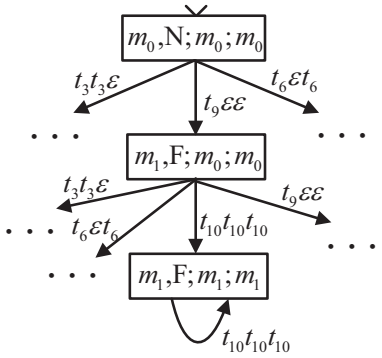

Fig. 3. The Verifier of the PN system in Example 1.

  - $t \in T_f$ and $(m, t, m') \in \Delta$.
- $((m, l; m_1, m_2), tt_1t_2, (m', l; m_1'; m_2')) \in \Delta^V$ if
  - $t \in T_{o,1} \cap T_{o,2}$, $(m, t, m') \in \Delta$, $(m_1, t_1, m_1') \in \Delta^1$, $(m_2, t_2, m_2') \in \Delta^2$, $L_1(t) = L_1(t_1)$ and $L_2(t) = L_2(t_2)$.
- $((m, l; m_1, m_2), tt_1\varepsilon, (m', l; m_1'; m_2)) \in \Delta^V$ if
  - $t \in T_{o,1} \setminus T_{o,2}$, $(m, t, m') \in \Delta$, $(m_1, t_1, m_1') \in \Delta^1$ and $L_1(t) = L_1(t_1)$.
- $((m, l; m_1, m_2), t\varepsilon t_2, (m', l; m_1; m_2')) \in \Delta^V$ if
  - $t \in T_{o,2} \setminus T_{o,1}$, $(m, t, m') \in \Delta$, $(m_2, t_2, m_2') \in \Delta^2$ and $L_2(t) = L_2(t_2)$.

5. Trim the automaton $V = (M^V, E^V, \Delta^V, m_0^V)$ by removing the states that are not reachable from the initial state m_0^V and all their input and output edges.

By construction, the Verifier captures a triple of sequences $(\sigma, \sigma_1, \sigma_2)$ satisfying the following conditions: $\sigma \in L(G_e)$, $\sigma_1 \in L(G_e^1)$, $\sigma_2 \in L(G_e^2)$, $\mathcal{L}_1(\sigma) = \mathcal{L}_1(\sigma_1)$, and $\mathcal{L}_2(\sigma) = \mathcal{L}_2(\sigma_2)$. N(resp., F) indicates that σ does not(resp., does) include a fault in T_f.

A state $(m, l; m_1; m_2)$ in the Verifier is called an l-state. For example, the initial state m_0^V is an N-state.

A cycle in the Verifier is called an l-cycle if each state in the cycle is an l-state.

Example 5 *Let us consider again the PN system in Example 1. Fig. 3 shows a part of the Verifier. The cycle* $((m_1, F; m_1; m_1), t_{10}t_{10}t_{10}, (m_1, F; m_1; m_1))$ *is an F-cycle.*

Now, since sequences in $L(G_e)$ also include faults, while sequences in $L(G_e^j)$'s do not, looking at sequences in the Verifier, we could establish if there exists any faulty sequence in $L(G_e)$ whose observable projection in all sites could be explained without the firing of any fault. This is formalized in the following result part a).

Property 3 *Let* $(N, m_0, \mathcal{L})$ *be a labeled PN system with EBRG* G_e. *Let* G_e^1 *and* G_e^2 *be the nonfailure-ERBGs wrt site 1 and site 2. The Verifier* V *constructed using Algorithm 2 has the following properties.*

a) The language of V *is:*

$$\begin{aligned} L(V) = \{\ & (\sigma, \sigma_1, \sigma_2) \mid \exists \bar{\sigma}, \bar{\sigma}_1, \bar{\sigma}_2 \in L(N, M_0) \\ & \sigma \in \rho(\bar{\sigma}), \sigma_1 \in \rho_1(\bar{\sigma}_1), \sigma_2 \in \rho_2(\bar{\sigma}_2), \\ & L_1(\bar{\sigma}) = L_1(\bar{\sigma}_1), L_2(\bar{\sigma}) = L_2(\bar{\sigma}_2), \\ & \bar{\sigma}_1 \cap T_f = \bar{\sigma}_2 \cap T_f = \emptyset\} \end{aligned}$$

b) If state $(m, l; m_1; m_2)$ is reached in V from the initial state with a sequence $(\sigma, \sigma_1, \sigma_2)$, then

$$l = \begin{cases} N & \text{iff } (\forall \bar{\sigma} \in \rho^{-1}(\sigma)),\ \bar{\sigma} \cap T_f = \emptyset \\ F & \text{iff } (\forall \bar{\sigma} \in \rho^{-1}(\sigma)),\ \bar{\sigma} \cap T_f \neq \emptyset \end{cases}$$

Proof. *See [Arxiv paper].*

In the following, we use $m_0^V \xrightarrow{(\sigma,\sigma_1,\sigma_2)} m^{V'}$ to denote that state $m^{V'}$ is reached in V from m_0^V with a sequence $(\sigma, \sigma_1, \sigma_2)$.

Theorem 2 *Let $V = (M^V, E^V, \Delta^V, m_0^V)$ be the Verifier of a given PN system constructed by Algorithm 2. The net has failure ambiguous sequences of arbitrary length after the occurrence of some fault in T_f iff V contains F-cycles.*

Proof. *Consider an evolution of V such that*

$$m_0^V \xrightarrow{(\sigma',\sigma_1',\sigma_2')} m^{V'} \xrightarrow{(t',t_1',t_2')} m^V$$

where $m^{V'}$ is an N-state and m^V is an F-state. Hence there exists a sequence $\bar{\sigma}' \in \rho^{-1}(\sigma' t')$ that ends with a fault by Property 3. By assumption A1 this sequence can be continued indefinitely and by assumption A2 there exists an integer $K > 0$ such that all continuations of length greater than or equal to $|\bar{\sigma}'| + K$ can be correctly diagnosed in a centralized setting.

This means that all sequences in the set

$$A(\bar{\sigma}') = \{\bar{\sigma} = \bar{\sigma}'\bar{\sigma}'' \mid |\bar{\sigma}''| \geq K, (\rho(\bar{\sigma}), \rho_1(\bar{\sigma}), \rho_2(\bar{\sigma})) \in L(V)\}$$

are failure ambiguous, since they can be correctly diagnosed as faulty in a centralized setting but can be explained by nonfaulty sequences by the two sites. Additionally for any such sequence $\bar{\sigma}$, the sequence $(\rho(\bar{\sigma}), \rho_1(\bar{\sigma}), \rho_2(\bar{\sigma}))$ drives the Verifier to an F-state by Property 3.

Now, there exist failure ambiguous sequences of arbitrary length after the fault, if and only if there exists a sequence $\bar{\sigma}'$ of the net that ends with a fault and is such that $A(\bar{\sigma}')$ is an infinite set. From this set we can extract an infinite increasing sequence $\bar{\sigma}_0$, $\bar{\sigma}_1 = \bar{\sigma}_0\bar{t}_1$, $\bar{\sigma}_2 = \bar{\sigma}_0\bar{t}_1\bar{t}_2, \ldots$. Obviously the chain of states of the Verifier reached by sequences $(\rho(\bar{\sigma}_0), \rho_1(\bar{\sigma}_0), \rho_2(\bar{\sigma}_0))$, $(\rho(\bar{\sigma}_1), \rho_1(\bar{\sigma}_1), \rho_2(\bar{\sigma}_1))$, $(\rho(\bar{\sigma}_2), \rho_1(\bar{\sigma}_2), \rho_2(\bar{\sigma}_2))$, ... belongs to an infinite path of F-states by assumption A4. Since the set of states of V is finite by assumption A3, this is possible if and only if there exists an F-cycle.

Corollary 1 *A labeled PN system $(N, m_0, \mathcal{L})$ monitored by two local sites is codiagnosable iff its Verifier has no F-cycles.*

Proof. *Straightforward from Theorems 1 and 2.*

Example 6 *Let us consider again Example 5. According to Corollary 1, we conclude that the PN system is not codiagnosable since there exists an F-cycle in the Verifier.*

In the discussion so far, we only considered one fault class. In the case of r fault classes we need to construct r different Verifiers, one for each fault class. When verifying the codiagnosability wrt a given fault class T_f^i, all fault transitions in $T_f \setminus T_f^i$ should be considered as regular unobservable transitions.

We conclude this section with a brief discussion on the complexity of our method. The size of the state space of the EBRG, in the worst case, is equal to that of the reachability graph. However, the EBRG has significantly fewer states than the reachability graph in most cases. For example, the number of reachable markings of the PN in Example 1 is $\binom{8+k-1}{k}$, while the number of states in the EBRGs is $k + 1$.

Let x be the number of nodes in G_e, i.e., $x = |M_e|$. Assume that the PN system is monitored by ν local

sites and has r fault classes. According to Algorithm 2, the number of nodes and edges in the Verifier are at most equal to $2x^{\nu+1}$ and $2x^{\nu+1} \times |T|^{\nu+1}$, respectively. Moreover, we need to check all cycles in the Verifier. This can be computed by Tarjan's strongly connected components algorithm [15], whose complexity is linear in the sum of the number of nodes and arcs in the Verifier, i.e., $O((x \times |T|)^{\nu+1})$. Hence, the overall complexity is $O((x \times |T|)^{\nu+1} \times r)$.

6 Conclusions

This paper proposes a new approach to verify codiagnosability of labeled bounded Petri nets. It is based on the result that a necessary and sufficient condition for codiagnosability is the absence of failure ambiguous sequences that are arbitrarily long after the occurrence of any fault. An automaton, called Verifier, is constructed to detect the presence of such kind of sequences. The main feature of the proposed method is that it uses the notion of basis markings thus avoiding exhaustive enumeration of the state space.

References

[1] J.C. Basilio and S. Lafortune. Robust codiagnosability of discrete event systems. In *American Control Conference, 2009. ACC '09.*, pages 2202–2209, June 2009.

[2] M.P. Cabasino, A. Giua, S. Lafortune, and C. Seatzu. A New Approach for Diagnosability Analysis of Petri Nets Using Verifier Nets. *Automatic Control, IEEE Transactions on*, 57(12):3104–3117, Dec 2012.

[3] 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.*, volume 18, pages 6060–6066, Aug 2011.

[4] M.P. Cabasino, A. Giua, M. Pocci, and C. Seatzu. Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems. *Control Engineering Practice*, 19(9):989 – 1001, Sep 2011.

[5] M.P. Cabasino, A. Giua, and C. Seatzu. Diagnosability of Discrete-Event Systems Using Labeled Petri Nets. *Automation Science and Engineering, IEEE Transactions on*, 11(1):144–153, Jan 2014.

[6] R. Debouk, S. Lafortune, and D. Teneketzis. Coordinated Decentralized Protocols for Failure Diagnosis of Discrete Event Systems. *Discrete Event Dynamic Systems*, 10(1):33–86, January 2000.

[7] S. Genc and S. Lafortune. Distributed Diagnosis of Place-Bordered Petri Nets. *Automation Science and Engineering, IEEE Transactions on*, 4(2):206–219, April 2007.

[8] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial algorithm for testing diagnosability of discrete-event systems. *Automatic Control, IEEE Transactions on*, 46(8):1318–1321, Aug 2001.

[9] G. Jiroveanu and R.K. Boel. The Diagnosability of Petri Net Models Using Minimal Explanations. *Automatic Control, IEEE Transactions on*, 55(7):1663–1668, July 2010.

[10] F. Lin. Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems*, 4(2):197–212, May 1994.

[11] M.V. Moreira, T.C. Jesus, and J.C. Basilio. Polynomial Time Verification of Decentralized Diagnosability of Discrete Event Systems. *Automatic Control, IEEE Transactions on*, 56(7):1679–1684, July 2011.

[12] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, Apr 1989.

[13] W. Qiu and R. Kumar. Decentralized failure diagnosis of discrete event systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 36(2):384–395, March 2006.

[14] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *Automatic Control, IEEE Transactions on*, 40(9):1555–1575, Sep 1995.

[15] R. Tarjan. Depth-First Search and Linear Graph Algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[16] Y. Wang, T.-S. Yoo, and S. Lafortune. Diagnosis of Discrete Event Systems Using Decentralized Architectures. *Discrete Event Dynamic Systems*, 17(2):233–263, 2007.

[17] Y. Wen and M. Jeng. Diagnosability analysis based on T-invariants of Petri nets. In *Networking, Sensing and Control, 2005. Proceedings. 2005 IEEE*, pages 371–376, March 2005.

[18] Y. Wen, C. Li, and M. Jeng. A polynomial algorithm for checking diagnosability of Petri nets. In *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, volume 3, pages 2542–2547 Vol. 3, Oct 2005.

[19] T.-S. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially observed discrete-event systems. *Automatic Control, IEEE Transactions on*, 47(9):1491–1495, Sep 2002.