K-diagnosability of labeled Petri nets

Baisi Liu, Mohamed Ghazel, Armand Toguyeni

MajecSTIC 2012
Lille, France, du 29 au 31 octobre 2012

K-diagnosability of labeled Petri nets

Baisi Liu[1,3], Mohamed Ghazel[1,2] et Armand Toguyéni[1,3]

1 : Univ Lille Nord de France, F-59000 Lille, France
2 : Univ Lille Nord de France, F-59000 Lille, IFSTTAR, ESTAS, F-59650 Villeneuve d'Ascq, France
3 : EC LILLE, LAGIS, F-59651, Villeneuve d'Ascq, France

Contact : baisi.liu@ec-lille.fr

Résumé
Dans cet article, nous présentons une approche qui permet de résoudre le problème de la *K*-diagnosticabilité des réseaux de Petri (RdP) synchronisés et bornés. Tout d'abord, nous introduisons deux nouveaux concepts : la matrice d'incidence étendue qui intègre les informations relatives aux affectations des événements aux transitions, et le marquage étendu ainsi que l'équation d'état étendue, qui permet de transcrire le nombre d'occurrences de chaque événement depuis l'état initial, jusqu'à un certain marquage étendu cible. Sur la base de ces concepts, nous élaborons un nouvel algorithme d'analyse, à la volée, de la *K*-diagnosticabilité des RdP synchronisés partiellement observables. L'algorithme développé se base sur des procédures récursives et ne nécessite pas une construction a priori du graphe de marquage ni d'un diagnostiqueur.

Abstract
This paper presents an approach to solve the problem of *K*-diagnosability of bounded labeled Petri nets (PNs). We first introduce extended incidence matrix, marking-eventing and extended state equation, which make it possible to record the number of event occurrences for any marking reachable from the initial one. Based on the proposed conceptions, we give a mathematical representation for labeled PNs to check diagnosability, and describe *K*-diagnosability problem of partially observed PNs. Then we propose a recursive algorithm for testing *K*-diagnosability of PNs, without construction of marking graph or diagnoser.

Mots-clés : *K*-diagnosticabilité, systèmes à événements discrets, réseaux de Petri synchronisés

Keywords: *K*-diagnosability, discrete event systems, labeled Petri nets

---

1. Introduction

Diagnosability is a critical issue that has received a great deal of attention. Simply speaking diagnosability refers to the ability to detect and locate any fault within a finite delay of observation after its occurrence. In the framework of discrete event systems (DESs), diagnosability was first formally proposed in [8]. The authors provided a systematic methodology of checking diagnosability, where a diagnoser automaton must be first constructed. Then other automata-based approaches [6, 12], aiming at reducing computational complexity, have been proposed. In [12] a polynomial-time algorithm for deciding diagnosability was presented. In [6] an algorithm based on the parallel composition of the investigated automaton with itself was proposed. These methods are based on algorithms which investigate whether the system is not diagnosable by seeking for some specific cycles. Hence the system is diagnosable if such cycles do not exist.
Furthermore, some works on diagnosability of DESs turned to PN technology, benefiting from mathematical and graphical representative capability and well-developed theory of PNs. The definition of diagnosability under formal languages was first extended to unbounded PNs in [10], with propositions of a simple ω diagnoser and sufficient conditions for diagnosability of unbounded PNs. In [11] the authors proposed a sufficient condition for testing diagnosability by checking the structure of T-invariant of the PN.

The classical diagnosability problem deals with determining qualitatively the existence of a finite delay to make the system diagnosable. Practically this delay may be so big that it is not advisable to obtain the solution at cost of much time. Thus some "practical" and "quantitative" versions of diagnosability are developed. *K*-diagnosability [3] is one of the developed version of diagnosability in which we must be able to determine with certainty that a fault has occurred within *K* (observable) events after any fault event. Another similar version, $\mathcal{K}$-diagnosability for PNs, was proposed in [1], where $\mathcal{K}$ refers to the number of both observable and unobservable transitions after any failure transition. Besides the former two untimed versions, diagnosability is also developed for timed systems. In [9] the authors discussed Δ-diagnosability for timed DES, i.e. to announce a fault within a delay of at most Δ time units after the fault occurred, and gave necessary and sufficient conditions for the diagnosability of a timed automaton. Generally speaking, there are mainly two problems on *K*-diagnosability. The first is to study *K*-diagnosability of a system under a given value *K*, i.e. if any fault can be detected and located within at most *K* steps after its occurrence. The second is to find the minimum *K* for a diagnosable system. In this paper we will deal with the former one.

The remainder of this paper is organized as follows. In Section 2 we briefly present some terms and notations that will be used in this paper. Then we introduce extended incidence matrix, marking-eventing and extended state equation for labeled PNs in Section 3. Finally, based on the proposed conceptions, we state the problem of *K*-diagnosability in Section 4, propose an algorithm to check *K*-diagnosability of labeled PN and provide an example.

2. Preliminaries

2.1. PNs and language

Petri nets, also named Place/Transition nets or P/T nets, are a graphical and mathematical modeling notation for DESs. A PN is a structure $N = (P, T, Pre, Post)$, where P is a finite set of places (circles in a PN graph) ; T is a finite set of transitions (boxes or bars in a PN graph) ; $Pre\colon P \times T \to \mathbb{N}$ and $Post\colon P \times T \to \mathbb{N}$ are the pre- and post-incidence functions that specify the weight of the arcs directed from places to transitions and from transitions to places, respectively. $C = Post - Pre$ is the incidence matrix.

A state of a PN is called marking, presented by a distribution of tokens (dots in a PN graph) in the places of the net. A marking is a vector $M\colon P \to \mathbb{N}$ that assigns a non-negative integer to each place. A marked PN (N, M_0) is a net N with an initial marking M_0.

The dynamics of PNs are presented by a movement or redistribution of tokens according to the firing rules. A transition t is enabled at marking M if $M \geq Pre(\cdot, t)$, denoted by $M[t >$. We may fire an enable transition t at marking M, yielding to a marking $M' = M + C \cdot \vec{t}$, where $\vec{t}\colon T \to \{0, 1\}$ is a vector in which only the entry associated with transition t is equal to 1. Marking M' is said to be reachable from marking M by firing transition t, and written by $M[t > M'$. A sequence of transitions $\sigma = t_1 t_2 ... t_k$ is executable at marking M, if $M[t_1 > M_1[t_2 > ...M_{k-1}[t_k >$, and we write it as $M[\sigma >$. The reached marking M' is computed by $M' = M + C \cdot \pi(\sigma)$, and denoted by $M[\sigma > M'$, where $\pi(\sigma) = \sum_{i=1}^{k} \vec{t_i}$ is the firing vector of σ.

A Petri net (N, M_0) is said to be bounded if the number of tokens in each place does not exceed a finite number k for any marking reachable from M_0. A PN is said to be live if, for any marking reachable from M_0, there exists an enabled transition to bring the PN to a future state.

A language over an event set Σ is a set of finite-length strings formed from events in Σ. In order to represent a language by a PN, each transition of the PN is associated with an event by a labeling function, therefore labeled PNs are introduced.

A labeled PN is a structure $G = (N, M_0, \Sigma, \varphi)$, where (N, M_0) is a marked PN with an initial marking M_0 ; Σ is a finite set of events for transition labeling ; $\varphi\colon T \to \Sigma$ is the transition labeling function, and it is extended to sequence of transitions, $\varphi\colon T^* \to \Sigma^*$. A labeled PN graph is presented as a PN graph of which each transition is labeled by an event in Σ, e.g. Figure 1 is a labeled PN graph where $a, b, f \in \Sigma$.

A PN language defined over event set Σ is a set $L = \{\varphi(\sigma) \in \Sigma^* : \sigma \in T^* \text{ and } M_0[\sigma >\}$. If string s is an concatenation of $s_1, s_2, ..., s_n$, where $s_1, s_2, ..., s_n \in \Sigma^*$, we write $s = s_1 s_2 ... s_n$.

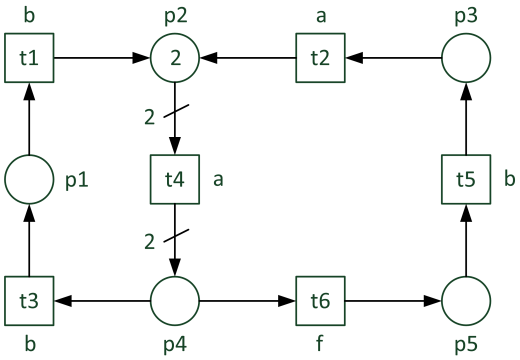

FIGURE 1 – A labeled PN

2.2. *K*-diagnosability

In event-based diagnosis of DES, event set Σ is partitioned into two disjoint sets, $\Sigma = \Sigma_o \biguplus \Sigma_{uo}$, where Σ_o is a finite set of observable events and Σ_{uo} is a finite set of unobservable events. Failure events are regarded as unobservable, thus the set of failure events $\Sigma_f \subseteq \Sigma_{uo}$. Accordingly, the set of transitions of a labeled PN T is partitioned into the set of observable and unobservable transitions, $T = T_o \dot{\cup} T_{uo}$, and the set of failure transitions $T_f \subseteq T_{uo}$. Moreover, for diagnosis on multiple faults, the set of failure events is partitioned into multiple disjoint sets, $\Sigma_f = \Sigma_{f_1} \biguplus \Sigma_{f_2} \biguplus ... \biguplus \Sigma_{f_m}$, where $\Sigma_{f_i} (i = 1, 2, ..., m)$ denotes one class of faults with fault label f_i.

Let $P_{\Sigma_o} : \Sigma^* \to \Sigma_o^*$ be the projection, which "erases" the unobservable events in a sequence $s \in \Sigma^*$. Define the inverse projection operator P_L^{-1} as $P_L^{-1}(r) = \{s \in L \; such \; that \; P_{\Sigma_o}(s) = r\}$. Given a live and prefix-closed language $L \in \Sigma^*$ and a string $s \in L$, the post-language of L after s denoted by L/s, is the language $L/s = \{s' \in \Sigma^* such \; that \; ss' \in L\}$. We denote the length of string s by $|s|$, and the i^{th} event of sequence s by s^i.

We now give an extension of the definition of the diagnosability in [8].

Definition 1 *(K-diagnosable fault) Given a labeled PN* G *and* $K \in \mathbb{N}$, $f \in \Sigma_f$ *is said to be* K*-diagnosable if* $\forall u \in L, u^{|u|} \in \Sigma_f$ *and* $\forall v \in L/u$ *such that* $|P_{\Sigma_o}(v)| \geq K$*, then it is* $r \in P_L^{-1}(P_{\Sigma_o}(uv)) \Rightarrow f \in r$.

3. Representation of labeled PNs to check diagnosability

The static and dynamic behaviors of a PN can be thoroughly described by the classical mathematical representation, with the help of markings, incidence matrix and state equations. This technology, however, is not sufficient for the description of labeled PNs, as there is no mathematical expression for events and their behaviors. In this context we propose a novel mathematical representation for labeled PNs, based on some new notions that we introduce, namely extended incidence matrix, marking-eventing and extended state equation, to present both mapping relationship between transitions and events and the quantitative record of event occurrences. Let $G = (N, M_0, \Sigma, \varphi)$ be a labeled PN, and $\Sigma' \subseteq \Sigma$.

Definition 2 *An event-incidence matrix is a* $|\Sigma'| \times |T|$ *matrix* $C_e : \Sigma' \times T \to \{0, 1\}$*, where* $C_e(i, j) = 1$ *if transition* t_j *is associated with event* e_i*, otherwise* $C_e(i, j) = 0$.

When $\Sigma' = \Sigma$, matrices C_e, Pre, Post, and initial marking M_0, provide a complete description of a labeled PN structure. That is, we can rebuild a labeled PN by the above matrices and vector.

Definition 3 *An extended incidence matrix* C_x *is the orderly composition of incidence matrix and event-incidence matrix,*

$$C_x = \begin{bmatrix} C \\ C_e \end{bmatrix}$$

.

Definition 4 *The eventing of marking* M *is a vector* $E : \Sigma' \to \mathbb{N}$ *that assigns a non-negative integer number of event occurrences from* M_0 *to each event.*

Let the initial eventing be $E_0 = \vec{0}$, since there is no occurrence of any event at its corresponding marking M_0.

Definition 5 *A marking-eventing* ME *is the orderly composition of a marking* M *and its corresponding eventing E,*

$$ME = \begin{bmatrix} M \\ E \end{bmatrix}$$

Let the initial marking-eventing be

$$ME_0 = \begin{bmatrix} M_0 \\ E_0 \end{bmatrix} = \begin{bmatrix} M_0 \\ \vec{0} \end{bmatrix}$$

We compute the successive marking-eventing of ME_0 by

$$ME_k = ME_0 + C_x \cdot \pi(\sigma)$$

and we write it as $ME_0[\sigma > ME_k$.
In event-based diagnosis, we focus on the occurrences of observable events and certain types of faults that we are interested in, rather than the other unobservable events. If we take into account multiple faults, we make $\Sigma' = \Sigma_o \cup \Sigma_f$; whereas for only one fault class f_i, let $\Sigma' = \Sigma_o \cup \Sigma_{f_i}$. For simplicity, we denote $mark(ME) = M$ and $R(ME_0)$ the set of all the marking-eventings that can be reached from ME_0. Let $obs(ME)\colon \Sigma_o \to \mathbb{N}$ and $fault(ME)\colon \Sigma_{f_i} \to \mathbb{N}$, then we have

$$ME = \begin{bmatrix} M \\ E \end{bmatrix} = \begin{bmatrix} mark(ME) \\ obs(ME) \\ fault(ME) \end{bmatrix}$$

Example
For the labeled PN in Figure 1, $M_0 = [0\ 2\ 0\ 0\ 0]^T, \Sigma_o = \{a, b\}, \Sigma_{uo} = \Sigma_f = \{f\}$. $\Sigma' = \Sigma_o \cup \Sigma_f$ is the set of events that we are interested in. For an executable sequence $\sigma = t_4t_6t_5t_2t_6t_5$ we have $ME_0[t_4 > ME_1[t_6 > ME_2[t_5 > ME_3[t_2 > ME_4[t_6 > ME_5[t_5 > ME_6$ and $ME_6 = ME_0 + C_x \cdot \pi(\sigma)$,

$$\begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ \hdashline 0 \\ 0 \\ 0 \end{bmatrix} [t_4 > \begin{bmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 0 \\ \hdashline 1 \\ 0 \\ 0 \end{bmatrix} [t_6 > ...[t_2 > \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ \hdashline 2 \\ 1 \\ 1 \end{bmatrix} [t_6 > \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ \hdashline 2 \\ 1 \\ 2 \end{bmatrix} [t_5 > \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ \hdashline 2 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 0 \\ 0 \\ 0 \\ \hdashline 0 \\ 0 \\ 0 \end{bmatrix} + \left[\begin{array}{cccccc} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & -2 & 0 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 2 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ \hdashline 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right] \cdot \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 2 \\ 2 \end{bmatrix}$$

$$mark(ME_0) = [0\ 2\ 0\ 0\ 0]^T,\ obs(ME_4) = obs(ME_5) = [2\ 1]^T,\ fault(ME_5) = fault(ME_6) = [2]^T$$

4. *K*-diagnosability of labeled Petri nets

The formal definition of *K*-diagnosability of a fault is recalled in Section 2. We can extend *K*-diagnosability from one fault to multiple faults, by assigning to each class of fault Σ_{f_i} a K_i, and transform the diagnosability problem of multiple faults into a series of problems of K_i-diagnosability of Σ_{f_i}, as K_i could be different according to Σ_{f_i}. This gives a finer and more quantitative analysis of the diagnosability problem than that of the general approach. In the sequel, we will first discuss the *K*-diagnosability issue of a fault f.

4.1. Problem statement

Before the discussion, we make the following assumptions :

1. the labeled PN is live and bounded ;
2. there is no executable cycle of unobservable transitions, i.e. $\forall\, ME \in R(ME_0)$, $\nexists\, \sigma \in T^*$ such that $ME[\sigma > ME', mark(ME) = mark(ME')$ and $obs(ME) = obs(ME')$.

Then we introduce some notations to help describing the *K*-diagnosability problem.

Definition 6 *A marking-eventing set (hereafter, ME-set) $\mathcal{ME}$ is a finite set of marking-eventings that are reached from the initial marking-eventing ME_0, by executing the sequences of the same observation and ending with an observable transition,*

$$\mathcal{ME} = \{ME\colon \text{for a given string } s \in L, \exists\, \sigma \in T^* \text{ such that } \sigma^{|\sigma|} \in T_o, P_{\Sigma_o}[\varphi(\sigma)] = s, ME_0[\sigma > ME\}$$

Obviously, for any two marking-eventings $ME, ME' \in \mathcal{ME}$, we have $obs(ME) = obs(ME')$; while the converse may not be true, since a marking-eventing records only the number but no order of event occurrences.

Definition 7 *For a given ME-set $\mathcal{ME}$ and an observable event e, define λ-function of $\mathcal{ME}$ under e as :*

$$\lambda(\mathcal{ME}, e) = \{ME'\colon \exists\, ME \in \mathcal{ME} \text{ and } \sigma \in T^* \text{ such that } P_{\Sigma_o}(\varphi(\sigma)) = \varphi(\sigma^{|\sigma|}) = e, ME[\sigma > ME'\}$$

Here $\lambda(\mathcal{ME}, e)$ is the ME-set reachable from $\mathcal{ME}$ by execution of a sequence of unobservable events followed by event e. We extend this definition to the set of observable events Σ_o. Then we define the Λ-function of $\mathcal{ME}$ as :

$$\Lambda(\mathcal{ME}) = \cup_{e \in \Sigma_o}\{\lambda(\mathcal{ME}, e)\}$$

We denote $\mathcal{ME} \prec \mathcal{ME}'$ if $\mathcal{ME}' \in \Lambda(\mathcal{ME})$.

Definition 8 *For a given ME-set $\mathcal{ME}$ and a fault f, we say that $\mathcal{ME}$ is*
– *normal, if* $\forall\, ME \in \mathcal{ME}, fault(ME) = \vec{0}$;
– f-*uncertain, if* $\exists\, ME_1, ME_2 \in \mathcal{ME}$ *and* $ME_1 \neq ME_2$, *such that* $fault(ME_1) \neq \vec{0}, fault(ME_2) = \vec{0}$;
– f-*certain, if* $\forall\, ME \in \mathcal{ME}, fault(ME) \neq \vec{0}$.

Let $tag(\mathcal{ME})$ denote the tag of ME-set $\mathcal{ME}$.

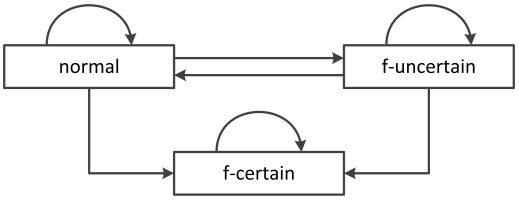

FIGURE 2 – Transformation relations between fault tags of ME-sets

For an ME-set $\mathcal{ME}' \in \Lambda(\mathcal{ME})$, $tag(\mathcal{ME}')$ may be different from $tag(\mathcal{ME})$. The transformation relations between them are shown in Figure 2, where an arrow indicates an observable event. If $\mathcal{ME}$ is normal (resp. f-uncertain), $\mathcal{ME}'$ could be normal, f-uncertain or f-certain. If $\mathcal{ME}$ is f-certain, $\mathcal{ME}'$ is definitely f-certain as the faults are assumed to be permanent. A similar idea is presented by the fault propagation function of diagnoser automata in [8].
In order to discuss the *K*-diagnosability, we introduce the ME-set tree. An ME-set tree is a tree structure, in which

1. the root node is the initial ME-set $\mathcal{ME}_0 = \{ME_0\}$.
2. for any given node $\mathcal{ME}$, the set of its child nodes is $\Lambda(\mathcal{ME})$.

Definition 9 *Two ME-sets $\mathcal{ME}, \mathcal{ME}'$ are said to be equivalent, denoted by $\mathcal{ME} \sim \mathcal{ME}'$, if $\forall\ ME \in \mathcal{ME}, \exists\ ME' \in \mathcal{ME}'$ such that $mark(ME) = mark(ME')$ and [1] $sgn[fault(ME)] = sgn[fault(ME')]$; moreover, $\forall\ ME' \in \mathcal{ME}', \exists\ ME \in \mathcal{ME}$ such that $mark(ME') = mark(ME)$ and $sgn[fault(ME')] = sgn[fault(ME)]$.*

Definition 10 *Define $\mathcal{ME} \circledcirc \mathcal{ME}'$ if*

1. *$\mathcal{ME} \sim \mathcal{ME}'$;*
2. *$\exists \mathcal{ME}_i, i \in 1, 2, ...n, \mathcal{ME}_i$ is f-uncertain, such that $\mathcal{ME} \prec \mathcal{ME}_1 \prec \mathcal{ME}_2 \prec ... \prec \mathcal{ME}_n \prec \mathcal{ME}'$;*
3. *$\forall ME \in \mathcal{ME}, \exists ME' \in \mathcal{ME}'$ and $\sigma \in T^*$ such that $mark(ME) = mark(ME')$ and $ME[\sigma > ME'$.*

Definition 11 *Define $\mathcal{C}(\mathcal{ME}) = \{\mathcal{ME}' \mid \mathcal{ME}'$ is f-uncertain, $\exists \mathcal{ME}_i, i \in 1, 2, ...n, \mathcal{ME}_i$ is f-uncertain, such that $\mathcal{ME} \prec \mathcal{ME}_1 \prec ... \prec \mathcal{ME}_n \prec \mathcal{ME}'\}$*

Definition 12 *In the generation of the ME-set tree, define $delay(\mathcal{ME}_0) = 0$. For any node $\mathcal{ME}' \in \Lambda(\mathcal{ME})$ and there does not exists $\mathcal{ME}''$ in the tree such that $\mathcal{ME}' \circledcirc \mathcal{ME}''$, we define $delay(\mathcal{ME}')$ as shown in Table 1, where "N", "U", "F" denote normal, f-uncertain and f-certain ME-set, respectively.*

$\mathcal{ME}$	$\mathcal{ME}'$	$delay(\mathcal{ME}')$	$\mathcal{ME}$	$\mathcal{ME}'$	$delay(\mathcal{ME}')$
N	N	0	U	N	0
N	U	1	U	U	$delay(\mathcal{ME}) + 1$
N	F	1	U	F	$delay(\mathcal{ME}) + 1$
			F	F	$delay(\mathcal{ME}) + 1$

TABLE 1 – Definition of delay function

For a newly generated node $\mathcal{ME}'$ and an existing node $\mathcal{ME}$ such that $delay(\mathcal{ME}) = d, \mathcal{ME} \circledcirc \mathcal{ME}'$, $delay(\mathcal{ME}) = delay(\mathcal{ME}') = max[d, delay(\mathcal{ME}')]$, and for any $\mathcal{ME}'' \in \mathcal{C}(\mathcal{ME})$, $delay(\mathcal{ME}'')$ should be updated according to Table 1.

4.2. Algorithm

We solve *K*-diagnosability in two steps : for each executable sequence $s \in L$,

1. find the first occurrence of the fault (if there exists one) ;
2. continue investigating all the possible observations after the first occurrence of the fault for at most *K* steps, to determine if there exist other sequences with the same observation that make it distinguishable from that which contains no fault.

We propose a recursive algorithm (Algorithm 1) for testing *K*-diagnosability of a labeled PN. This algorithm computes, step by step with Λ-function, to build the ME-set tree. In the computation, for any node $\mathcal{ME}'$,

1. if $\mathcal{ME}'$ is f-certain, it is not necessary to consider its child nodes, as all of its child nodes carry f-certain tag, and a fault must be detected by looking at this node.
2. if $\mathcal{ME}'$ is normal and there exists an investigated $\mathcal{ME}''$ such that $\mathcal{ME}' \sim \mathcal{ME}''$, then we stop investigating this branch, since $\mathcal{ME}'$ has the same branches as $\mathcal{ME}''$, which has already been considered. (Lines 4 - 7 in Algorithm 1)
3. if $\mathcal{ME}'$ is f-uncertain, the cases are more complicated, (Lines 8 - 17 in Algorithm 1)

1. Define sign function $sgn : \mathbb{N}^1 \to 0, 1, sgn(x) = 0$ if $x = \vec{0}$, otherwise $sgn(x) = 1$.

(a) if $\mathtt{delay}(\mathcal{ME}') = K$, it means that until $\mathcal{ME}'$ there are already K continuous f-uncertain nodes, i.e. the fault cannot be detected after K observable events, thus G is not K-diagnosable. (Line 9 in Algorithm 1)

(b) if $\mathtt{delay}(\mathcal{ME}') < K$,

i. if there exists an investigated $\mathcal{ME}''$ such that $\mathcal{ME}' \sim \mathcal{ME}''$,

A. if $\mathcal{ME}' \circledcirc \mathcal{ME}''$, it means that there exists an F_i-indeterminate cycle [8], therefore G is not diagnosable.

B. else update the $\mathtt{delay}$ value of nodes in $\mathcal{C}(\mathcal{ME}'')$, and check if there exists $\mathcal{ME}''' \in \mathcal{C}(\mathcal{ME}'')$ such that $\mathtt{delay}(\mathcal{ME}''') \geq K$. If so, G is not diagnosable.

ii. else continue investigating this branch.

Algorithm 1: Algorithm for testing K-diagnosability of labeled PN G

```
   input : T, Pre, Post, M_0, C_e, K
   output: K-diagnosability of G
1  solve (ME, tag, delay, e)
2      ME' ⟵ λ(ME, e);
3      switch tag(ME') do
4          case normal
5              if ∄ ME'' ∈ M_vst such that ME' ~ ME'' then
6                  M_vst = M_vst ∪ {ME'};
7                  foreach t ∈ T do solve(ME', normal, delay, φ(t));
8          case f-uncertain
9              if delay(ME') = K then return G is not K-diagnosable;
10             if ∃ ME'' ∈ M_vst such that ME' ~ ME'' then
11                 if ME'' ⊚ ME' then return G is not K-diagnosable;
12                 else
13                     foreach ME''' ∈ C(ME'') do
14                         if delay(ME''') ≥ K then return G is not K-diagnosable;
15             else
16                 M_vst = M_vst ∪ {ME'};
17                 foreach t ∈ T do solve(ME', f-uncertain, delay(ME'), φ(t));
18 begin
19     M_vst ⟵ {ME_0};
20     foreach t ∈ T do solve(ME_0, normal, 0, φ(t));
21     return G is K-diagnosable;
22 end
```

4.3. Example

Analyze the $1, 2, 3, 4$-diagnosability of PN G in Figure 1 respectively. According to the solution process illustrated by Figure 3, we conclude that G is not $1, 2, 3$-diagnosable, while it is 4-diagnosable.

5. Conclusion

The K-diagnosability is stated by our mathematical representation of labeled PNs. A recursive algorithm is proposed to check K-diagnosability of a system modeled as labeled PN on the fly. Compared with previous works, neither a marking graph nor a diagnoser are first constructed in our algorithm. Moreover, it allows us to perform the investigation for each fault Σ_{f_i} under a given K_i, therefore the diagnosability can be analyzed in a finer manner.

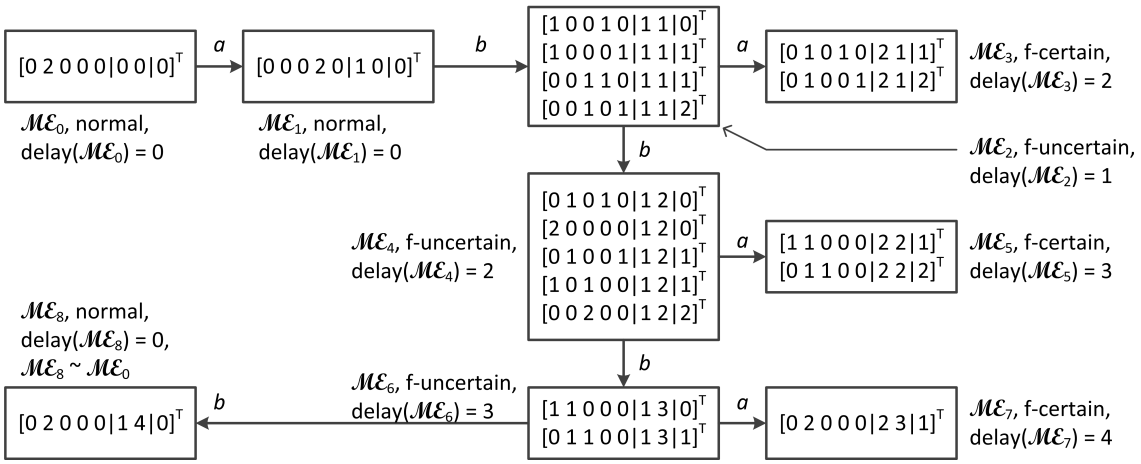

FIGURE 3 – Solution process of $1, 2, 3, 4$-diagnosability for PN G

References

1. F. Basile, P. Chiacchio, and G. De Tommasi. Diagnosability of labeled Petri nets via integer linear programming. *Discret. Event Syst.*, 10, 2010.
2. C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
3. E. Dallal and S. Lafortune. On most permissive observers in dynamic sensor optimization problems for discrete event systems. In *48th Annu. Allerton Conf. Commun., Control, Comput., 2010*, pages 318 – 324, sep 2010.
4. M. Ghazel. *Surveillance des Systèmes à Evénements Discrets à l'aide des Réseaux de Petri Temporels*. PhD thesis, LAGIS - Ecole Centrale de Lille, nov 2005.
5. M. Ghazel, A. Toguyéni, and P. Yim. State observer for DES under partial observation with time Petri nets. *Discret. Event Dyn. Syst.*, 19(2) :137 – 165, 2009.
6. S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial algorithm for testing diagnosability of discrete-event systems. *IEEE Trans. Autom. Control*, 46(8) :1318 – 1321, aug 2001.
7. T. Murata. Petri nets : Properties, analysis and applications. *Proc. IEEE*, 77(4) :541 – 580, apr 1989.
8. M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis. Diagnosability of discrete-event systems. *IEEE Trans. Autom. Control*, 40(9) :1555 – 1575, sep 1995.
9. S. Tripakis. Fault diagnosis for timed automata. In Werner Damm and Ernst Olderog, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 205 – 221. Springer Berlin / Heidelberg, 2002.
10. T. Ushio, I. Onishi, and K. Okuda. Fault detection based on Petri net models with faulty behaviors. In *IEEE Int. Conf. Syst., Man, Cybern., 1998*, volume 1, pages 113 – 118, oct 1998.
11. Y. Wen, M. Jeng, L. Jeng, and F. Pei-Shu. An intelligent technique based on Petri nets for diagnosability enhancement of discrete event systems. In Bogdan Gabrys, Robert Howlett, and Lakhmi Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, pages 879 – 887. Springer Berlin / Heidelberg, 2006.
12. T.-S. Yoo and S. Lafortune. Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Trans. Autom. Control*, 47(9) :1491 – 1495, sep 2002.