Enforcement of diagnosability in labeled Petri nets via optimal sensor selection

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

[a] *College of Electronic and Information Engineering, Hebei University, China (ranning87@hotmail.com)*

[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 deal with the problem of enforcing diagnosability to labeled Petri nets appropriately adding new sensors. We show that, solving an integer linear programming problem, it is possible to select a solution that is optimal with respect to a given objective function (e.g., the cost of sensors). The solution is based on two notions, already introduced by the authors in previous works, namely *basis marking* and *Unfolded Verifier*. This allows to solve the considered problem in a more efficient way with respect to other approaches in the literature. Finally, we propose an algorithm to compute the smallest value of K such that the Petri net system is K-diagnosable under the new labeling function, which implies that faults can be detected in at most K observations after their occurrence.

Published as:
Ning Ran, Alessandro Giua, Carla Seatzu, "Enforcement of diagnosability in labeled Petri nets via optimal sensor selection," *IEEE Transactions on Automatic Control*, Vol. 64, No. 7, pp. 2997-3004, July 2019.
DOI: 10.1109/TAC.2018.2874020.

1 Introduction

A discrete event system is said to be diagnosable if, once a fault has occurred, its occurrence can be detected based on a finite number of observations obtained via appropriate sensors placed in the system. The problem of establishing if a given system is diagnosable has been extensively studied in the past few decades, following the framework of Lafortune [7], both using automata and Petri nets [2,4,8,10,12,13]. In the case of Petri nets, transitions are typically partitioned into observable and unobservable transitions. To the first ones, sensors are attached and produce an observable event whenever they fire. To the second ones, no sensor is attached, so their firing is silent with respect to an observer. Clearly, diagnosability is strictly related to the way sensors are attached to transitions. In this paper we address the following problem: given a system that is not diagnosable under a given configuration of sensors, we want to reconfigure the sensors placement to make the system diagnosable. In particular, we look for a solution that minimizes a given performance index, which typically consists in the cost associated with the new sensors.

To the best of our knowledge, this problem has only been considered by Cabasino *et al.* [7] for bounded/unbounded labeled Petri nets. The method in [7] is based on the notion of *Verifier Net* [4] and on a series of relabeling rules to eliminate the paths in the Unfolded reachability/coverability graph of the Verifier Net that lead to a violation of the conditions for diagnosability. The application of such an approach is limited in practical situations by the fact that the number of reachable markings may increase exponentially with the size of the net (structure and number of tokens in the initial marking).

The goal of this paper is that of looking for a practically efficient solution that can cope with the state explosion problem. In particular, we propose a solution that takes advantage of a special automaton, called *Unfolded Verifier*, which is the unfolded version of another automaton called *Verifier*, recently presented by some authors of this paper in [11] to make codiagnosability analysis of bounded nets. Such an automaton benefits of the notion of *basis marking* [5] and thus reduces (in some cases significantly, depending on the net structure, on the initial marking, and on the labeling function) the set of markings that should be enumerated, with consequent advantages in terms of computational complexity. Similarly to [7], necessary and sufficient conditions for diagnosability can be derived looking at some paths in the Unfolded Verifier. Then, paths in the Unfolded Verifier that prevent diagnosability could be eliminated, appropriate relabeling some of the transitions. A labeling function that is optimal according to a given criterion can be selected solving an integer linear programming problem.

In this paper we also deal with a problem that was not considered in [7] related to the the notion of K-diagnosability [11], which guarantees that faults can be detected in at most K observations after their occurrence. More precisely, we show how to select among the set of optimal solutions of the considered sensor allocation problem, one that leads to the smallest value of K such that the system is K-diagnosable. A similar problem has been addressed by Basile *et al.* in [3]. However, a fair comparison in terms of computational complexity between the two approaches could not be done. Indeed in [3] authors deal with a slightly different notion of K-diagnosability and also with a different problem statement: no a priori partition in observable and unobservable transitions is given there. The solution they proposed (that also applies to unobservable-induced subnets that are not acyclic) is based on integer linear programming, rather that graph analysis.

2 Background on Labeled Petri Nets

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 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 $\mathbb{N} = 0, 1, 2, ...$: $m(p)$ is equal to the number of tokens in place p. (N, m_0) is a PN system with an initial marking m_0.

A transition t is enabled at m if $\forall p \in {}^\bullet t, m(p) \geq W(p,t)$ and is written as $m[t\rangle$. Firing t yields to m' such that $\forall p \in P, m'(p) = m(p) + [N](p,t)$, which is written as $m[t\rangle m'$. $m[\sigma\rangle$ is used to denote that the transition sequence $\sigma = t_1 t_2 ... t_k$ is enabled at 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 $R(N, m)$. We indicate as $|\sigma|$ the length of the sequence σ. 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 use λ to indicate the empty transition sequence in T^*. We write $t \in \sigma$ to denote that a transition t is contained in σ, $T' \cap \sigma \neq \emptyset$ to indicate 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.

A PN is said to be bounded if there exists a positive constant k such that $\forall p \in P$, $\forall m \in R(N, m_0)$, $m(p) \leq k$.

Given a PN system (N, m_0), $t \in T$ 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 the T'-induced subnet of N as 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 PN with no directed circuits is said to be acyclic.

A labeled PN system (LPNS) 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 in A^*.

We use T_u (T_o) to denote the set of transitions whose labels are ε (symbols in A). T_u and T_o are called the set of *unobservable* and *observable* transitions, respectively. Two or more transitions sharing the same label are called *indistinguishable.* $[N]_u$ (or $[N]_o$) is used to denote the restriction of the incidence matrix $[N]$ to T_u (or T_o).

The labeling function is extended to strings $\mathcal{L} : T^* \to A^*$:

(1) $\mathcal{L}(\lambda) = \varepsilon$, where λ is the empty transition sequence;

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

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

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

$\mathcal{L}^{-1}(w)$ denotes the set of all transition sequences consistent with $w \in A^*$, 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 $\mathcal{L}(L(N, m_0))$.

Given a transition sequence $\sigma \in T^*$, we denote $P_u(\sigma)$ (or $P_o(\sigma)$) the projection of σ over T_u (or T_o). Let $K \subseteq T^*$ be a language, we use K/σ to denote the post-language of K after σ, i.e., $K/\sigma = \{\sigma' \in T^* \mid \sigma\sigma' \in K\}$.

3 Problem Statement

Consider an LPNS $(N, m_0, \mathcal{L}_{init})$ with an "initial" labeling function $\mathcal{L}_{init} : T \to A_{init} \cup \{\varepsilon\}$ that assigns to each transition in T either a symbol from a given alphabet A_{init} or the empty string ε. 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. 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 set of regular transitions T_{reg} is partitioned into $T_{r,o} \cup T_{r,uo}$, where $T_{r,o}$ (resp., $T_{r,uo}$) is the set of regular transitions to which is possible (resp., not possible) to associate a sensor. We use $[N]_{r,uo}$ to denote the restriction of the incidence matrix to $T_{r,uo}$. We also assume that $A_{init} \cap T \neq \emptyset$, i.e., none of the original labels can be the name of a transition.

Following [7], we say that $(N, m_0, \mathcal{L})$ having no deadlock after any fault is not diagnosable wrt T_f^i if given any $k \in \mathbb{N}$ there exist two transition sequences $\sigma_1, \sigma_2 \in L(N, m_0)$ satisfying the conditions: (1) $\mathcal{L}(\sigma_1) = \mathcal{L}(\sigma_2)$; (2) $\sigma_1 \cap T_f^i \neq \emptyset$; (3) $\sigma_2 \cap T_f^i = \emptyset$; (4) σ_1 is of "arbitrary length" after fault transition $t_f \in T_f^i$, i.e., there exists at least one decomposition $\sigma_1 = \sigma_1' t_f \sigma_1''$ with $|\sigma_1''| > k$.

Let us now consider the special labeling function $\mathcal{L}_{total}$ where each transition in $T_o \cup T_{r,o}$ is assigned a unique symbol that distinguishes it from all other transitions. Without loss of generality, we take these unique symbols to be the names of the transitions, i.e., $\forall t \in T_o \cup T_{r,o}$, $\mathcal{L}_{total}(t) = t$. In this case, $A_{total} = T_o \cup T_{r,o}$; all transitions except those in $T_{r,uo} \cup T_f$ have a unique label. We make the following assumptions.

(A1) The PN system is bounded.

(A2) The PN system does not enter a deadlock after the occurrence of any fault.

(A3) The $(T_{r,uo} \cup T_f)$-induced subnets is acyclic.

(A4) The PN system is diagnosable under $\mathcal{L}_{total}$.

(A5) There is a single fault class.

As clarified in the following section, Assumption (A1) is necessary to implement the proposed approach based on the notion of basis marking [5]. Note that such an assumption was made in [7] when introducing the problem statement. However, in [7] this was done for the sake of simplicity in the presentation of the results but was not a requirement. Assumption (A2) is a weakened version of the typical "liveness" assumption made in most of the works on diagnosability of discrete event systems. Assumption (A3), which does not appear in [7], is fundamental to apply the proposed approach based on the notion of basis marking [5] [1] . Assumption (A4) ensures that it is possible to diagnose the occurrence of fault transitions when all transitions in $T_o \cup T_{r,o}$ are unambiguously observable. Finally, Assumption (A5) is made for simplicity of exposition. A discussion on how to relax it is proposed in Section 5.2.

Assume that $(N, m_0, \mathcal{L}_{init})$ is not diagnosable. Our objective is to find a new labeling function $\mathcal{L}_{new} : T \to A_{new}$ such that $(N, m_0, \mathcal{L}_{new})$ is diagnosable, where $A_{new} = A_{init} \cup A_{total}$. Furthermore, we also would like to choose the labeling function $\mathcal{L}_{new}$ in a manner that minimizes an objective function related to the cost of attaching sensors to the transitions of the PN system.

As in [7], we assume that a transition $t \in T_o \cup T_{r,o}$ is relabeled based on the following rules:

(R1) For a transition $t \in T_{r,o}$, we either set $\mathcal{L}_{new}(t) = t$ or leave it unchanged as $\mathcal{L}_{new}(t) = \mathcal{L}_{init}(t) = \varepsilon$.

(R2) For a transition $t \in T_o$ such that there exists $t' \in T_o$ with $t \neq t'$ and $\mathcal{L}_{init}(t) = \mathcal{L}_{init}(t')$, we either set $\mathcal{L}_{new}(t) = t$ or leave it unchanged as $\mathcal{L}_{new}(t) = \mathcal{L}_{init}(t)$.

In either case, $\mathcal{L}_{new}$ may assign a new unique label to transition t, namely, t itself.

In [7] it has been proved that when rules (R1) and (R2) are applied incrementally for relabeling, at each

[1] More precisely, the necessity of Assumption A3 follows from Theorem 3.8 in [6]. The acyclicity of the unobservable subnet is indeed instrumental to ensure that a basis marking description provides necessary and sufficient conditions to determine if a marking (or firing sequence) is consistent with a given observation.

step, two transition sequences that are distinguishable under the initial labeling function L_{init} will always remain distinguishable.

Let $\mathcal{L}_{new}$ be a new labeling function obtained from L_{init} by repeated application of rules (R1) and (R2). We denote $T_{new} \subseteq T_o \cup T_{r,o}$ the set of transitions whose labels are changed by $\mathcal{L}_{new}$, i.e., $T_{new} = \{t \in T_o \cup T_{r,o} : \mathcal{L}_{new}(t) = t\}$. By assumption (A4), we know that there exists a choice of T_{new} for which the system is diagnosable, namely, $T_{new} = T_o \cup T_{r,o}$. In [7] it has been demonstrated that given $(N, m_0, \mathcal{L}_{init})$ that is not diagnosable, it is always possible to find at least one transition to relabel according to rules (R1) and (R2).

Finally, since in general multiple solutions exist to the above problem, as in [7], we look for a solution that minimizes a given performance index, e.g., $J(\mathcal{L}_{new}) = \sum_{t \in T_{new}} c_t$, where c_t is the cost of attaching a sensor to transition t that produces a uniquely identifiable label.

4 Unfolded Verifier

4.1 Preliminary definitions

Definition 1 *[5] Given a marking m and an observable transition t, the set of explanations of t at m is denoted by $\Sigma(m,t) = \{\sigma \in T_u^* \mid m[\sigma\rangle m', m'[t\rangle\}$, and the set of e-vectors is indicated as $Y(m,t) = \pi(\Sigma(m,t))$.*

Definition 2 *[5] Given a marking m and an observable transition t, the set of minimal explanations of t at m is denoted by $\Sigma_{min}(m,t) = \{\sigma \in \Sigma(m,t) \mid \nexists \sigma' \in \Sigma(m,t) : \pi(\sigma') \lneq \pi(\sigma)\}$, and the set of minimal e-vectors is indicated as $Y_{min}(m,t) = \pi(\Sigma_{min}(m,t))$.*

Definition 3 *[5] Let $(N, m_0, \mathcal{L})$ be an LPNS 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 indicated as*

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

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

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

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

Definition 4 *[5] Let $(N, m_0, \mathcal{L})$ be an LPNS, $w \in L^*$ be an observation and $\hat{\mathcal{J}}(w)$ be a set of pairs. The set of basis markings of w is indicated as*

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

and $M_b = \bigcup\limits_{w \in L^*} M_b(w)$.

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) [5]. 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.

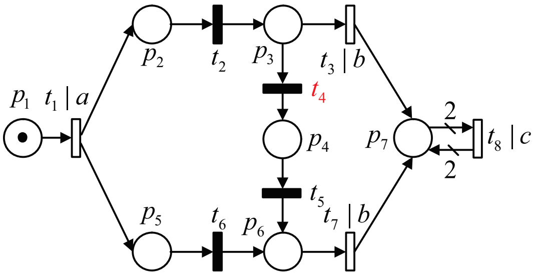


Fig. 1. An LPNS $(N, m_0, \mathcal{L}_{init})$.

Definition 5 *An extended basis marking (EBM) is a basis marking computed assuming that all transitions in $T_f \cup T_{r,o}$ 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 $T_{r,uo}$. In the following, we denote $Y_{min}^{r,uo}(m,t)$ the set of minimal e-vectors restricted to $T_{r,uo}$. The set $Y_{min}^{r,uo}(m,t)$ can be computed using Algorithm 4.4 in [5].

Example 1 *Consider the LPNS $(N, m_0, \mathcal{L}_{init})$ in Fig. 1, where $T_o = \{t_1, t_3, t_7, t_8\}$, $T_u = \{t_2, t_4, t_5, t_6\}$, $T_f = \{t_4\}$, $T_{r,o} = \{t_5\}$, $T_{r,uo} = \{t_2, t_6\}$ and $m_0 = [1\ 0\ 0\ 0\ 0\ 0\ 0]^T$. The labeling function is defined as follows: $\mathcal{L}_{init}(t_1) = a$, $\mathcal{L}_{init}(t_3) = \mathcal{L}_{init}(t_7) = b$ and $\mathcal{L}_{init}(t_8) = c$. The set of EBMs is $m_0 = [1\ 0\ 0\ 0\ 0\ 0\ 0]^T$, $m_1 = [0\ 1\ 0\ 0\ 1\ 0\ 0]^T$, $m_2 = [0\ 0\ 0\ 0\ 1\ 0\ 1]^T$, $m_3 = [0\ 0\ 0\ 1\ 1\ 0\ 0]^T$, $m_4 = [0\ 1\ 0\ 0\ 0\ 0\ 1]^T$, $m_5 = [0\ 0\ 0\ 0\ 1\ 1\ 0]^T$, $m_6 = [0\ 0\ 0\ 1\ 0\ 0\ 1]^T$, $m_7 = [0\ 0\ 0\ 0\ 0\ 1\ 1]^T$, $m_8 = [0\ 0\ 0\ 0\ 0\ 0\ 2]^T$.*

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 $T \setminus T_f$.

We now define the following two graphs inspired from [11].

Definition 6 *Let $(N, m_0, \mathcal{L})$ be an LPNS, M_e the set of EBMs, and $(N', m_0, \mathcal{L}')$ the nonfailure subnet of $(N, m_0, \mathcal{L})$. • 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; $E \subseteq (T_o \times A) \cup T_f \cup T_{r,o}$ 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 $e = t \in T_f \cup T_{r,o}$, if and only if $\exists y \in Y_{min}^{r,uo}(m,t)$ and $m' = m + [N]_{r,uo} \cdot y + [N](\cdot, t)$.*

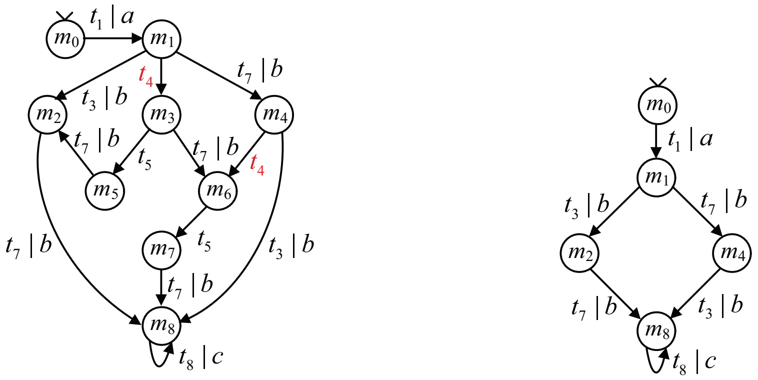
• The nonfailure EBRG, denoted by $G_e^n = (M^n, E^n, \Delta^n, m_0)$, is the EBRG of $(N', m_0, \mathcal{L}')$ constructed under the assumption that the set of observable transitions is equal to $T_o \cup T_{r,o}$, and all transitions in $T_{r,uo}$ are unobservable.

The EBRG G_e is computed using Algorithm 1 in [11] but assuming that the set of observable transitions is equal to $T_o \cup T_{r,o} \cup T_f$, and restricting minimal explanations to the set $T_{r,uo}$. The nonfailure EBRG G_e^n can also be computed using Algorithm 1 in [11] assuming that the set of observable transitions is equal to $T_o \cup T_{r,o}$, and restricting minimal explanations to the set $T_{r,uo}$.

Example 2 *The EBRG G_e and the nonfailure EBRG G_e^n of the LPNS in Example 1 are reported in Fig. 2a and Fig. 2b, respectively.*

4.2 Unfolded Verifier

An *Unfolded Verifier* (UV) $U = (M^U, E^U, \Delta^U, m_0^U)$ is a (non-deterministic) finite state automaton constructed using the following algorithm.



(a) (b)

Fig. 2. a) EBRG G_e of the LPNS in Fig. 1.b) Nonfailure EBRG G_e^n of the LPNS in Fig. 1.

```
Algorithm 1: [Construction of the Unfolded Verifier]

Input: $G_e = (M_e, E, \Delta, m_0)$ and $G_e^n = (M^n, E^n, \Delta^n, m_0)$.

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

1. Let $m_0^U = (m_0, \mathrm{N}; m_0)$ be the root node.
2. While nodes with no tag exist, do
   2.1. select a node $(m_1, l; m_2)$ with no tag,
   2.2. if $(m_1, l; m_2)$ is identical to a node on the path from the root node to $(m_1, l; m_2)$, then tag the
        node $(m_1, l; m_2)$ "duplicate" and goto step 2.
   2.3. $\forall t_1 \in T_o \cup T_f \cup T_{r,o}$ and $\forall t_2 \in T_o \cup T_{r,o}$, do
        • add a node $(m_1', l; m_2')$ and an arc $(t_1, t_2)$ from $(m_1, l; m_2)$ to $(m_1', l; m_2')$ if
            – $(m_1, t_1, m_1') \in \Delta$, $(m_2, t_2, m_2') \in \Delta^n$, $t_1, t_2 \in T_o$, and $\mathcal{L}(t_1) = \mathcal{L}(t_2)$;
        • add a node $(m_1', \mathrm{F}; m_2)$ and an arc $(t_1, \lambda)$ from $(m_1, l; m_2)$ to $(m_1', \mathrm{F}; m_2)$ if
            – $t_1 \in T_f$ and $(m_1, t_1, m_1') \in \Delta$;
        • add a node $(m_1', l; m_2)$ and an arc $(t_1, \lambda)$ from $(m_1, l; m_2)$ to $(m_1', l; m_2)$ if
            – $t_1 \in T_{r,o}$ and $(m_1, t_1, m_1') \in \Delta$.
        • add a node $(m_1, l; m_2')$ and an arc $(\lambda, t_2)$ from $(m_1, l; m_2)$ to $(m_1, l; m_2')$ if
            – $t_2 \in T_{r,o}$ and $(m_2, t_2, m_2') \in \Delta^n$.
   2.4. tag the node $(m_1, l; m_2)$ "old".
```

A state $(m_1, l; m_2)$ in the UV is called an l-state. Furthermore, if it is tagged "duplicate" it is called a *duplicate l-state.*

Proposition 1 *An LPNS $(N, m_0, \mathcal{L})$ is diagnosable iff its UV U has no duplicate F-states.*

Proof. *The statement easily follows from a result proved in [11] and the fact that the UV is the "unfolded version" of the Verifier in [11] in the case of a single observation site that observes all the events in the alphabet A. In particular, an l-cycle of the Verifier is a cycle where each node is an l-state. In [11] it has been proved that an LPNS is diagnosable iff its Verifier has no F-cycles. Now, the only difference between the algorithm to construct the UV and the algorithm to construct the Verifier consists in the way repeated nodes are handled: the first algorithm does not fuse repeated nodes, and tag a node "duplicate" if it is identical to another node on the path from the root node to the considered one; on the contrary, the second algorithm fuses repeated nodes. Therefore, the necessary an sufficient condition for diagnosability based on F-cycles in the Verifier can be easily rephrased as follows in terms of duplicate nodes in the UV:*

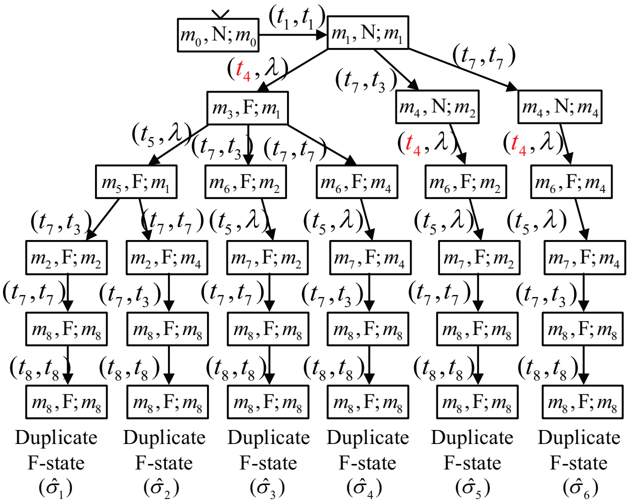


Fig. 3. The elementary F-paths for the LPNS of Example 1.

the LPNS $(N, m_0, \mathcal{L})$ is diagnosable iff its UV U has no duplicate F-states.

Given an automaton G, we write $m \xrightarrow[G]{\sigma} m'$ to denote that m' is reached in G from m with a sequence σ.

Definition 7 *Let $(N, m_0, \mathcal{L})$ be an LPNS and U be its UV. A sequence $\hat{\sigma} = (\gamma_{i_1}, \gamma_{j_1})(\gamma_{i_2}, \gamma_{j_2}) \ldots (\gamma_{i_k}, \gamma_{j_k})$ in U is called an elementary F-path wrt $\mathcal{L}$ if the following holds:*

(1) $m_0 \xrightarrow[G_e]{\gamma_{i_1} \cdots \gamma_{i_q}} m \xrightarrow[G_e]{\gamma_{i_{q+1}} \cdots \gamma_{i_k}} m$*;*
(2) $m_0 \xrightarrow[G_e^n]{\gamma_{j_1} \cdots \gamma_{j_q}} m' \xrightarrow[G_e^n]{\gamma_{j_{q+1}} \cdots \gamma_{j_k}} m'$*;*
(3) $\mathcal{L}(\gamma_{i_1} \ldots \gamma_{i_q}) = \mathcal{L}(\gamma_{j_1} \ldots \gamma_{j_q})$ *and* $\mathcal{L}(\gamma_{i_{q+1}} \ldots \gamma_{i_k}) = \mathcal{L}(\gamma_{j_{q+1}} \ldots \gamma_{j_k})$*;*
(4) $T_f \cap (\gamma_{i_1} \ldots \gamma_{i_q}) \neq \emptyset$*;*
(5) no prefix of $\hat{\sigma}$ satisfies items (1) – (4).

In other words, a sequence $\hat{\sigma}$ in U is called an elementary F-path wrt $\mathcal{L}$ if the sequence $\hat{\sigma}$ starts at the root node of U and ends in a duplicate F-state.

Proposition 2 *A LPNS $(N, m_0, \mathcal{L})$ is diagnosable iff its UV has no elementary F-paths wrt $\mathcal{L}$.*

Proof. *Follows from Proposition 1 and Definition 7.*

Example 3 *Let us consider again Example 1. We first construct the UV U, which is not reported here for the sake of brevity. The elementary F-paths in U are drawn in Fig. 3. Obviously, the state $(m_8, F; m_8)$ is a duplicate F-state, and there are 6 elementary F-paths wrt $\mathcal{L}_{init}$: $\hat{\sigma}_1$ – $\hat{\sigma}_6$. Hence this net is not diagnosable given L_{init}.*

5 Optimal Sensor Selection

We now show how the approach in [7] based on Integer Linear Programming (ILP), can be efficiently used starting from the UV.

5.1 Relabeling of elementary F-paths

The transitions of the UV are pairs (γ_i, γ_j) where:

(1) γ_i either corresponds to a transition in the EBRG G_e or to λ;

(2) γ_j either corresponds to a transition in the nonfailure EBRG G_e^n or to λ.

Based on rules (R1) and (R2) described in Section 3, we propose the following *relabeling options* for the transitions that compose an elementary F-path.

(LO1) (γ_i, γ_j) where $\gamma_i = t_i$ and $\gamma_j = t_j$ with $i \neq j$. According to Algorithm 1, we know that $t_i, t_j \in T_o$ with $\mathcal{L}_{init}(t_i) = \mathcal{L}_{init}(t_j)$. In such a case, we can either assign a new label to t_i, $\mathcal{L}_{new}(t_i) = t_i$, or to t_j, $\mathcal{L}_{new}(t_j) = t_j$.

(LO2) (γ_i, λ) where $\gamma_i = t_i \in T_{r,o}$. In such a case, we can make the unobservable transition t_i observable by assigning to it $\mathcal{L}_{new}(t_i) = t_i$.

(LO3) (λ, γ_j) where $\gamma_j = t_j \in T_{r,o}$. In such a case, we can make the unobservable transition t_j observable by assigning to it $\mathcal{L}_{new}(t_j) = t_j$.

(LO4) (γ_i, γ_j) where $\gamma_i = t_i$ and $\gamma_j = t_j$ with $i = j$. In such a case it is irrelevant to relabel t_i since we are synchronizing a transition with itself when building the UV.

(LO5) (γ_i, λ) where $\gamma_i \in T_f$. In such a case we do noting since we cannot make the fault transition observable.

It should be noted that the above relabeling options for the pair (γ_i, γ_j) are identical with the first five options in [7]. On the contrary, if the relabeling is based on the UV, rather the unfolded reachability/coverability graph of the Verifier Net as in [7], the last two options in [7] (namely, (LO6) and (LO7)) are no more necessary since they handle the case where transition γ_i (t_i) or γ_j (t_j) belongs to the set $T_{r,uo}$.

Now, as in [7], we drop (LO4) and (LO5) and based on the remaining three options, we define the following two rules for the construction of the new labeling function $\mathcal{L}_{new}$.

(R3) For each elementary F-path, relabel at least one transition $t \in T_o \cup T_{r,o}$ in the path from the root to the leaf based on one of the relabeling options (LO1), (LO2) and (LO3).

(R4) For a given elementary F-path, transition $t_i \in T_{r,o}$ should not be chosen in (R3) if it only appears in the path in *consecutive pairs* of the form: $(t_i, \lambda)(\lambda, t_i)$ or $(\lambda, t_i)(t_i, \lambda)$.

As explained in detail in [7], rule (R3) is necessary since if none of the transitions contained in the path are relabeled, then the same path will still arise after relabeling. Rule (R4) can be explained as follows. If t_i only appears in an elementary F-path in consecutive pairs $(t_i, \lambda)(\lambda, t_i)$ or $(\lambda, t_i)(t_i, \lambda)$, and is relabeled based on (LO2) or (LO3) and (R1), then the same elementary F-path may still arise in the new UV and the PN system remains non-diagnosable under $\mathcal{L}_{new}$. For more details, refer to Lemma 4.4 in [7].

The following proposition shows that we can always find at least one transition in each elementary F-path to relabel according to rule (R3).

Proposition 3 *In each elementary F-path in the UV, there must exist at least one transition that can be relabeled according to the relabeling options (LO1) and (LO2) or (LO3) and associated rules (R1), (R2), (R4).*

Proof. *By contradiction, assume that there exists an elementary F-path in which all transition are in the form of (LO4), (LO5) or consecutive pairs. In fact, we need not to consider the case of (LO5) since a failure cannot be relabeled. For the other two cases, we observe that even each transition are relabeled, the elementary F-path would still exist. Consequently, the PN is still not diagnosable under $\mathcal{L}_{total}$, which is a contradiction of assumption (A4). Hence, the result holds.*

The following two propositions, still inspired by [7], ensure that if we relabel an elementary F-path based on rules (R1) – (R4) the elementary F-path is no more feasible, and no new elementary F-path is created.

Proposition 4 *Let $\hat{\sigma} = (\gamma_{i_1}, \gamma_{j_1})(\gamma_{i_2}, \gamma_{j_2}) \dots (\gamma_{i_k}, \gamma_{j_k})$ be an elementary F-path of length k that is relabeled according to the relabeling options (LO1) and (LO2) or (LO3) and associated rules (R1) – (R4). Let $\sigma = \gamma_{i_1}\gamma_{i_2} \dots \gamma_{i_k}$ and $\sigma' = \gamma_{j_1}\gamma_{j_2} \dots \gamma_{j_k}$. It holds that: $\mathcal{L}_{new}(\sigma) \neq \mathcal{L}_{new}(\sigma')$.*

Proof. *Let us consider the case of (LO1) and let (t_i, t_j) be the pair of transitions of interest where $t_i \in \sigma$ and $t_j \in \sigma'$. Since we either assign a new label to t_i, $\mathcal{L}_{new}(t_i) = t_i$, or to t_j, $\mathcal{L}_{new}(t_j) = t_j$, the observable projection of either σ or σ' wrt $\mathcal{L}_{new}$ is changed. Thus $\mathcal{L}_{new}(\sigma) \neq \mathcal{L}_{new}(\sigma')$.*

Let us consider the case of (LO2) and (LO3). We can either relabel t_i (LO2) or t_j (LO3). However, by Rule (R4) it cannot occur that t_i is relabeled according to option (LO2) if tansition t_i only appears in the path in consecutive pairs of the form $(t_i, \lambda)(\lambda, t_i)$ or $(\lambda, t_i)(t_i, \lambda)$. Thus $\mathcal{L}_{new}(\sigma) \neq \mathcal{L}_{new}(\sigma')$. The same argument can be repeated for the case (LO3).

Proposition 5 *If each elementary F-path wrt $\mathcal{L}_{init}$ is relabeled according to the relabeling options (LO1) and (LO2) or (LO3) and associated rules (R1) – (R4), then in the LPNS $(N, m_0, \mathcal{L}_{new})$ there are no more elementary F-paths wrt $\mathcal{L}_{new}$, i.e., the relabeling creates no new elementary F-path.*

Proof. *By contradiction, assume that there is a "new" elementary F-path $\hat{\sigma} = (\gamma_{i_1}, \gamma_{j_1})(\gamma_{i_2}, \gamma_{j_2}) \dots (\gamma_{i_k}, \gamma_{j_k})$ of length k wrt $\mathcal{L}_{new}$. Let $\sigma = \gamma_{i_1}\gamma_{i_2} \dots \gamma_{i_k}$ and $\sigma' = \gamma_{j_1}\gamma_{j_2} \dots \gamma_{j_k}$. Obviously, there exist two transition sequences $\bar{\sigma}, \bar{\sigma}' \in T^*$ such that: (i) $\bar{\sigma}$ is arbitrarily long after the occurrence of the fault; (ii) $\bar{\sigma}'$ contains no fault; and (iii) $\mathcal{L}_{new}(\bar{\sigma}) = \mathcal{L}_{new}(\bar{\sigma}')$. Hence, it is $\mathcal{L}_{init}(\bar{\sigma}) = \mathcal{L}_{init}(\bar{\sigma}')$, and $\bar{\sigma}$ and $\bar{\sigma}'$ could form an elementary F-path wrt $\mathcal{L}_{init}$.*

By rule (R3), we have relabeled at least one transition in each elementary F-path. According to relabeling options (LO1) – (LO3) and rule (R4), it holds that $\mathcal{L}_{new}(\bar{\sigma}) \neq \mathcal{L}_{new}(\bar{\sigma}')$. This leads to a contradiction.

Theorem 1 *Let $(N, m_0, \mathcal{L}_{init})$ be a non-diagnosable PN system satisfying assumptions (A1) – (A5), and U be its UV. Let $\mathcal{L}_{new}$ be a labeling function obtained according to rules (R1) – (R4) with relabeling options (LO1) – (LO3) for elementary F-paths of U. Then $(N, m_0, \mathcal{L}_{new})$ is diagnosable.*

Proof. *The proof is straightforward from Propositions 2, 4 and 5. In fact, by Proposition 4 we know that each elementary F-path is disabled according to rules (R1) – (R4) with relabeling options (LO1) – (LO3). From Proposition 5 we can state that the relabeling procedure does not create any new elementary F-path. Hence, once we relabeled each elementary F-path in U, there are no more elementary F-paths. By Proposition 2, this implies that the PN system $(N, m_0, \mathcal{L}_{new})$ is diagnosable.*

5.2 Optimal relabeling using linear integer programming

In this subsection we recall the approach based on linear integer programming proposed in [7] that may also be applied (in a slightly simplified form) when the elementary paths are computed in the UV. Define a set of binary variables $v_t \in \{0, 1\}$ for each $t \in T_o \cup T_{r,o}$, where $v_t = 1$ means that the transition t has been relabeled under $\mathcal{L}_{new}$. For each elementary F-path in the UV, build an inequality of the form:

$$v_t + v_t' + v_t'' + ... \geq 1 \tag{1}$$

for some transitions $t, t', t'', ... \in T_o \cup T_{r,o}$, where the left hand side of (1) is obtained by the following procedure almost coincident with the one in [7]:

(1) Examine each transition pair (γ_i, γ_j) in the elementary F-path.

(2) For each transition pair (γ_i, γ_j), add terms to left hand side of (1) according to rules C1 – C5:

- C1. If $(\gamma_i, \gamma_j) = (t_i, t_j)$ with $i \neq j$, then add $+v_i + v_j$.
- C2. If $(\gamma_i, \gamma_j) = (t_i, \lambda)$ with $t_i \in T_{r,o}$
  · If the previous transition of the path is (λ, t_i), where $+v_i$ was added at that step, then add $-v_i$;

  - If the previous transition of the path is (λ, t_i), where $-v_i$ was added at that step, then add $+v_i$;
  - If the previous transition of the path is not (λ, t_i), then add $+v_i$.
- C3. If $(\gamma_i, \gamma_j) = (\lambda, t_j)$ with $t_j \in T_{r,o}$
  - If the previous transition of the path is (t_j, λ), where $+v_i$ was added at that step, then add $-v_j$;
  - If the previous transition of the path is (t_j, λ), where $-v_i$ was added at that step, then add $+v_j$;
  - If the previous transition of the path is not (t_j, λ), then add $+v_j$.
- C4. If $(\gamma_i, \gamma_j) = (t_i, t_j)$ with $i = j$, then do nothing.
- C5. If $\gamma_i \in T_f$ and $\gamma_j = \lambda$, then do nothing.

Rules C1 – C5 correspond to relabeling options (LO1) – (LO5). As stated in Section 5.1, the transitions in $T_{r,uo}$ would not arise in the elementary F-paths, thus the last two rules in [7] (C6 and C7) do not appear in this paper.

Let $\mathcal{R}$ be the set of all linear inequalities so obtained for all elementary F-paths of the UV.

Theorem 2 *Let $(N, m_0, \mathcal{L}_{init})$ be a non-diagnosable PN system, and U be its UV. There exists a new labeling function $\mathcal{L}_{new}$, which is obtained according to rules (R1) – (R4) with relabeling options (LO1) – (LO3) for elementary F-paths of U, such that $(N, m_0, \mathcal{L}_{new})$ is diagnosable iff the set $\mathcal{R}$ of linear inequalities has a solution.*

Proof. *(Only if) The set $\mathcal{R}$ of linear inequalities is obtained according to rules (R1) – (R4) with relabeling options (LO1) – (LO3). For each elementary F-path, we relabel either a transition $t \in T_{r,o}$ according to rule (R1) with (LO2)/(LO3), or a transition $t \in T_o$ according to rule (R2) with (LO1). In more detail, rules (R1) and (R2) are imposed by adding $+v_i$ (see C2) or $+v_j$ (see C3) or $+v_i + v_j$ (see C1) in the left hand side of the constraint of the form (1). Rule (R3) is imposed by "≥ 1" in the constraint. In particular, rule (R4) is imposed by (C2) and (C3): if there exists a consecutive pair of the form $(t_i, \lambda)(\lambda, t_i)$ or $(\lambda, t_i)(t_i, \lambda)$, we add $+v_i - v_i$ in the left hand side of the constraint. Hence, if there exists a labeling function $\mathcal{L}_{new}$, obtained according to rules (R1) – (R4) with (LO1) – (LO3), such that $(N, m_0, \mathcal{L}_{new})$ is diagnosable, then the set $\mathcal{R}$ of linear inequalities has a solution.*

(If) Straightforward from Theorem 1.

The new labeling function $\mathcal{L}_{new}$ that minimizes the total cost can be obtained solving the following ILP problem [7]:

$$\min \sum_{t \in T_o \cup T_{r,o}} c_t v_t, \qquad \text{s.t. } \mathcal{R} \tag{2}$$

where c_t and v_t are the cost and the binary variable associated with transition $t \in T_{new}$, respectively.

Example 4 *Consider the LPNS in Example 1. Associate a unitary cost with each transition in $T_o \cup T_{r,o}$ such that the ILP problem (2) reduces to minimize the number of sensors that need to be attached to make the system diagnosable. The elementary F-paths can be identified from Fig. 3 and for each of them we write a linear inequality according to rules C1 – C5. Therefore, the set $\mathcal{R}$ is defined by the six constraints: $v_{t_5} + v_{t_7} + v_{t_3} \geq 1$, $v_{t_5} + v_{t_7} + v_{t_3} \geq 1$, $v_{t_7} + v_{t_3} + v_{t_5} \geq 1$, $v_{t_5} + v_{t_7} + v_{t_3} \geq 1$, $v_{t_7} + v_{t_3} + v_{t_5} \geq 1$, $v_{t_5} + v_{t_7} + v_{t_3} \geq 1$, which in this special case, are coincident.*

We use the tool LINGO to solve the ILP problem (2). We obtain that an optimal solution is the relabeling of transition t_5 which is consistent with the fact that, if $\mathcal{L}_{new}(t_5) = t_5$, the PN system becomes diagnosable. Furthermore, such a solution is obviously optimal since at least one transition should be relabeled to guarantee diagnosability. Note that other optimal solutions exist, namely: $\mathcal{L}'_{new}(t_3) = t_3$ or $\mathcal{L}''_{new}(t_7) = t_7$.

We finally remark that, in the case of r fault classes we first construct an UV U_i for each fault class T_f^i, where all fault transitions in $T_f \setminus T_f^i$ should be considered as regular unobservable transitions that cannot be relabeled. Then we compute all elementary F-paths of U_i and build the set $\mathcal{R}_i$ of linear inequalities according to rules C1 – C5. Finally, we solve the ILP problem (2) with all sets of linear inequalities $\mathcal{R}_i$ for $i = 1, 2, ..., r$.

6 Comparison With the Approach in [7]

We first remark that the two approaches provide the same optimal solutions. Indeed, in both cases, the set of relabeling functions that lead to diagnosability, is exhaustively described by the set of constraints of the ILP problem to be solved (see Theorem 2 in this paper and Theorem 5.2 in [7]). What changes in the two cases, is the way such a set is obtained and described, namely how many paths should be considered and how many constraints should be written.

If we consider again the PN system in Example 1 and compute its pruned Reachability Tree, we realize that 616 paths should be considered (that can be visualized at the web page in [1]). All of them lead to the same constraint, namely $v_{t_3} + v_{t_7} + v_{t_5} \geq 1$, which coincides with the constraints obtained using the proposed approach, so the results of the optimization are the same in the two cases: we should relabel one of the following transitions: t_3, t_5, or t_7.

As a conclusion, our approach is definitely more efficient because the same set of constraints is obtained looking at only 6 paths rather than 616 as in [7]. This is a consequence of the fact that the notion of basis marking avoids exhaustive enumeration of the state space. Therefore, the structure of the UV is in general much simpler than the structure of the pruned RT and, as a consequence, the number of paths that need to be disabled is much smaller in the former case.

Note that in the above example it was immediate to remove redundancy among constraints since all of them were identical. In other cases (that are not reported here for sake of brevity) and in particular, in the presence of several integer variables, this is not the case. Therefore, using the approach in [7], not only we look at a larger number of paths, but we also deal with an ILP with a larger number of constraints.

We conclude this section with a brief discussion on the complexity of the proposed approach. We first observe that 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. Now, let x be the number of nodes of G_e, i.e., $x = |M_e|$. According to Algorithm 1, the maximum number of nodes in an elementary F-path is $2x^2 + 1$, and the maximum number of output arcs at each node is $(|T| + 1)^2 - 1$. Therefore, the maximum number of elementary F-paths is $((|T| + 1)^2 - 1)^{2x^2}$. Since we need to write an inequality for each elementary F-path, the complexity of generating an ILP problem is $O(((|T| + 1|)^2 - 1)^{2x^2})$.

It is well known that an ILP problem is NP-complete in the worst case. In our case, the fact that the variables are binary may help to mitigate the computation burden. The number of such variables is at most equal to $|T_o| + |T_{r,o}|$.

7 K-diagnosability

Definition 8 *[11] Consider an LPNS $(N, m_0, \mathcal{L})$ and an integer K. The LPNS $(N, m_0, \mathcal{L})$ is K-diagnosable wrt the i-th fault class T_f^i if there do not exist two transition sequences σ and σ' such that: (1) $T_f^i \cap \sigma \neq \emptyset$, $T_f^i \cap \sigma' = \emptyset$; (2) $\mathcal{L}(\sigma) = \mathcal{L}(\sigma')$; (3) the number of observable events in σ after the first occurrence of a fault transition $t_f \in T_f^i$ is K.*

The LPNS $(N, m_0, \mathcal{L})$ is K-diagnosable if it is K-diagnosable wrt all fault classes.

In simple words, an LPNS is K-diagnosable wrt a given fault class if faults in that class can be detected in at most K observations after the occurrence of the fault. Obviously, a K-diagnosable PN system is also K'-diagnosable if $K' > K$.

Let L_{opt} be the set of optimal relabeling functions corresponding to a given objective function. Tipically, relabeling functions in L_{opt} result in different properties in terms of K-diagnosability, i.e., all of them make the system K-diagnosable, but with a different value of K. We denote as $K_{min}(\mathcal{L}_{new})$ the smallest value of K such that the system relabeled under $\mathcal{L}_{new}$ is K-diagnosable. We want to select a relabeling function $\mathcal{L}^*_{new}$ to which it corresponds the smallest value of K_{min}, namely, we show to compute $K^*_{min} = \min_{\mathcal{L}_{new} \in L_{opt}} K_{min}(\mathcal{L}_{new})$ and $\mathcal{L}^*_{new} = \arg\min_{\mathcal{L}_{new} \in L_{opt}} K_{min}(\mathcal{L}_{new})$.

For sake of simplicity in the explanation, we again assume that there is a single fault class. In the case of r fault classes, the computation should be repeated r times separately.

The following algorithm allows to compute the value of $K_{min}(\mathcal{L}_{new})$ associated with a generic optimal solution $\mathcal{L}_{new}$. The value of K^*_{min} is obtained enumerating all the relabeling functions in L_{opt}, and arbitrarily selecting one to each it corresponds the smallest value of K_{min}.

```
Algorithm 2: [Computation of $K_{min}(\mathcal{L}_{new})$]

Input: Labeling function $\mathcal{L}_{new}$ and UV $U$.

Output: $K_{min}(\mathcal{L}_{new})$.

1. Let $z$ be the number of elementary F-paths of $U$.
2. For each elementary F-path $\hat{\sigma}_h$, $h = 1, 2, ..., z$, do
   2.1. let $K_h = 1$.
   2.2. for each transition pair $(\gamma_i, \gamma_j)$ along the path $\hat{\sigma}_h$, going from the root to the leaf, do
        • if $\gamma_i, \gamma_j \in T_o$, then
            · if $\gamma_i = \gamma_j$ and the output node of $(\gamma_i, \gamma_j)$ is an F-state, then $K_h = K_h + 1$.
            · if $\gamma_i \neq \gamma_j$, then
                  if neither $\gamma_i$ nor $\gamma_j$ are relabeled by $\mathcal{L}_{new}$ and the output node of $(\gamma_i, \gamma_j)$ is an F-state, then $K_h = K_h + 1$.
                  if $\gamma_i$ or $\gamma_j$ is relabeled by $\mathcal{L}_{new}$, then goto Step 2.
        • if $\gamma_i \in T_f$ and $\gamma_j = \lambda$, then do nothing.
        • if $\gamma_i \in T_{r,o}$ and $\gamma_j = \lambda$, then
            · if $(\gamma_i, \gamma_j)$ is tagged "consecutive pairs" and its output node is an F-state, then $K_h = K_h + 1$.
            · if $\gamma_i$ is relabeled by $\mathcal{L}_{new}$ and $(\gamma_i, \gamma_j)$ have no tag, then
                  if the subsequent transition pair in $\hat{\sigma}_h$ is $(\lambda, \gamma_i)$, then tag $(\lambda, \gamma_i)$ "consecutive pairs".
                  else goto Step 2.
        • if $\gamma_j \in T_{r,o}$ and $\gamma_i = \lambda$, then
            · if $(\gamma_i, \gamma_j)$ is tagged "consecutive pairs" and its output node is an F-state, then $K_h = K_h + 1$.
            · if $\gamma_j$ is relabeled by $\mathcal{L}_{new}$ and $(\gamma_i, \gamma_j)$ have no tag, then
                  if the subsequent transition pair in $\hat{\sigma}_h$ is $(\gamma_j, \lambda)$, then tag $(\gamma_j, \lambda)$ "consecutive pairs".
                  else goto Step 2.
3. Let $K_{min}(\mathcal{L}_{new}) = \max_{h=1,2,...,z} K_h$.
```

The basic idea behind Algorithm 2 is the following. Given an elementary F-path $\hat{\sigma}_h$ in U, we compute the smallest number of observations K_h that lead to the detection of the first occurrence of the fault, when the evolution of the system (relabeled with $\mathcal{L}_{new}$) corresponds to path $\hat{\sigma}_h$ in U. All elementary F-paths are examined, and $K_{min}(\mathcal{L}_{new})$ is equal to the largest value of K_h.

In more detail, Algorithm 2 can be explained step by step, as follows. Let us first introduce the notion of head F-state and terminal state of an elementary F-path $\hat{\sigma}_h$.

– We call *head F-state of* $\hat{\sigma}_h$ the first F-state we encounter following the path, starting from the root node.

– We call *terminal state of* $\hat{\sigma}_h$ the first node we encounter following the path, starting from the root node, whose output arc is "prevented" by $\mathcal{L}_{new}$.

The variable K_h is initialized at 1 since at least one observation should occur before detecting a fault. Step 2.2 examines each pair (γ_i, γ_j) in $\hat{\sigma}_h$ from the root to the leaf. In particular, the value of K_h is increased by 1 when encountering a pair (γ_i, γ_j) that satisfies the following conditions:

(1) the terminal state is an F-state; and
(2) the transition pair (γ_i, γ_j) locates between the head F-state and the terminal state of $\hat{\sigma}_h$, and it satisfies one of the following conditions:
    i) the transitions $\gamma_i, \gamma_j \in T_o$.
    ii) the transition $\gamma_i \in T_{r,o} \cap T_{new}$ and it appears in consecutive pairs.
    iii) the transition $\gamma_j \in T_{r,o} \cap T_{new}$ and it appears in consecutive pairs.

Finally, $K_{min}(\mathcal{L}_{new})$ is computed at Step 3 as the maximum value of K_h among all the elementary F-paths $\hat{\sigma}_h$, $h = 1, \ldots, z$.

Example 5 *Consider again the elementary F-paths in Fig. 3. We first consider the optimal solution* $\mathcal{L}_{new}(t_5) = t_5$. *For the elementary F-path* $\hat{\sigma}_1$, *the head F-state is* $(m_3, F; m_1)$, *the terminal state is also* $(m_3, F; m_1)$ *and the number of observable events (under* $\mathcal{L}_{new}$*) between the head F-state and the terminal state is 0. Therefore,* K_1 *is initialized at* 1 *and it is not increased. Analogously, for the other five elementary F-paths* $\hat{\sigma}_2$ *to* $\hat{\sigma}_6$, *it is* $K_2 = 1$, $K_3 = 2$, $K_4 = 2$, $K_5 = 1$ *and* $K_6 = 1$. *Hence,* $K_{min}(\mathcal{L}_{new}) = 2$, *i.e., the relabeled PN system is 2-diagnosable. We apply the same approach to the other three optimal solutions mentioned in Example 4, and find out that* $K_{min}(\mathcal{L}'_{new}) = K_{min}(\mathcal{L}''_{new}) = 2$. *Thus,* $K^*_{min} = 2$.

8 Conclusions and Future Work

This paper proposes a new approach to enforce diagnosability to a non-diagnosable labeled Petri net system by relabeling some transitions and optimizing a given objective function. We construct an automaton, called Unfolded Verifier, which allows to identify all paths that prevent diagnosability. Then, we formulate an integer linear programming problem based on some relabeling rules applied to the elementary F-paths of the Unfolded Verifier. Compared with the approach in [7], the new technique is computationally more efficient since it employs the notion of basis markings to prevent exhaustive enumeration of the set of reachable markings. We finally provide an approach to select, among the set of optimal relabeling functions, one that leads to the smallest value of K such that the PN system is K-diagnosable.

Our future efforts will be devoted to extending the current approach to unbounded PNs and to a decentralized setting. In particular, concerning unbounded PNs, we plan to use the notion of *Basis Coverability Graph* recently introduced in [9]. Finally, we plan to investigate if the ideas in this paper may be adapted to solve the problem of opacity enforcement.

References

[1] `http://www.diee.unica.it/~seatzu/Example_TAC17.pdf`.

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

[3] F. Basile, G. De Tommasi, and C. Sterle. Sensors selection for K-diagnosability of Petri nets via Integer Linear Programming. In *2015 23rd Mediterranean Conference on Control and Automation (MED)*, pages 168–175, June 2015.

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

[5] 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.

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

[7] M.P. Cabasino, S. Lafortune, and C. Seatzu. Optimal sensor selection for ensuring diagnosability in labeled Petri nets. *Automatica*, 49(8):2373 – 2383, 2013.

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

[9] E. Lefaucheux, A. Giua, and C. Seatzu. Basis Coverability Graph for Partially Observable Petri Nets with Application to Diagnosability Analysis. In *39th Int. Conf. on Applications and Theory of Petri Nets and Concurrency*, pages 164–183. Springer, 2018.

[10] A. Ramirez-Trevino, E. Ruiz-Beltran, J. Aramburo-Lizarraga, and E. Lopez-Mellado. Structural Diagnosability of DES and Design of Reduced Petri Net Diagnosers. *IEEE Trans. on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 42(2):416–429, March 2012.

[11] N. Ran, H. Su, A. Giua, and C. Seatzu. Codiagnosability analysis of bounded petri nets. *IEEE Transactions on Automatic Control*, 63(4):1192–1199, April 2018.

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

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