Kalpa Publications in Computing

Volume 4, 2018, Pages 52–64

28th International Workshop on
Principles of Diagnosis (DX'17)

An Experimental Comparison of Three Diagnosis Techniques for Discrete Event Systems*

Abderraouf Boussif[1], Baisi Liu[1], and Mohamed Ghazel[1]

Univ Lille Nord de France, IFSTTAR, COSYS, ESTAS, F-59650 Villeneuve d'Ascq, France
{abderraouf.boussif, baisi.liu, mohamed.ghazel}@ifsttar.fr

Abstract

This paper deals with a benchmark-based experimental comparison of three diagnoser-based approaches for fault diagnosis of discrete event systems modeled by Petri nets: the MBRG/BRD approach, the FMG/FMSG approach and the SSD approach. The experiments are performed on a level crossing benchmark, using the respective software tools integrating the approaches. Different features are shown in terms of state-space building (*exhaustive or partial*), procedure for analyzing diagnosability (*based on complete or on-the-fly built state-space*) and state-space representation (*concrete or symbolic*). Based on the obtained experimental results, a comparative discussion is provided particularly regarding memory and time consumption for analyzing diagnosability of the three techniques.

1 Introduction

Diagnosability analysis and on-line diagnosis are two crucial issues in safety-critical systems. Most of these systems can be modeled as discrete event systems (DESs) [1] at a high level of abstraction. For technical and/or economic reasons, it is generally inconceivable to sense all the behavioral variations in a complex dynamic system. Thereby, very often, monitoring activities in such systems have to be carried out under partial observability. In particular, DES diagnosis is often explained as the task to deduce and identify the occurrence of unobservable fault, based on the followed observed events. Diagnosability refers to the ability to diagnose any fault in a finite delay after its occurrence.

Most pioneering studies on fault diagnosis adopted finite state automata as analysis models [2–5]. Afterwards, fault diagnosis has also been dealt with in the Petri net (PN) framework [6–9]. For details, the reader can refer to the studies listed in [10].

This paper evaluates three diagnoser-based approaches for fault diagnosis of DESs modeled by PNs. The case study using the railway level crossing (LC) benchmark is carried out to compare the features of three approaches, namely (i) the modified basis reachability graph (MBRG)/basis reachability diagnoser (BRD) technique [7], (ii) the fault marking graph (FMG)/fault marking set graph (FMSG) technique [11] and the so-called semi-symbolic diagnoser (SSD) technique [12, 13].

*The authors acknowledge the support of the ELSAT2020 project. ELSAT2020 is co-financed by the European Union with the European Regional development Fund, the French state and the Hauts-de-France Region Council.

The following section briefly introduces some notations pertaining to PNs and fault diagnosis. In Section 3, an overview of the considered approaches is given. In Section 4, a discussion on the main features of these approaches is provided. In Section 5, a series of benchmark-based experiments are performed to assess the approaches. Finally, some concluding remarks are given in Section 6.

2 Preliminaries

2.1 Labeled Petri Net Modeling

A Petri net (PN) is a structure $N = (P, T, Pre, Post)$, where P is a finite set of places; T is a finite set of transitions; Pre and $Post$ are the pre- and post-incidence mappings, respectively. $C = Post - Pre$ is the incidence matrix. A marking is a vector $m \in \mathbb{N}^{|P|}$ that assigns a non-negative integer to each place. We denote by $m(p)$ the marking of a place p. A marked PN (N, m_0) is a PN N with a given initial marking m_0.

A transition t_i is enabled at marking m, denoted by $m[\, t_i >$, if $(\forall p \in P)[m(p) \geq Pre(p, t_i)]$. A transition t_i enabled at m can fire, yielding to a marking $m' = m + C \cdot \vec{t_i}$, where $\vec{t_i} \in \{0, 1\}^{|T|}$ is a vector in which only the entry associated with transition t_i is equal to 1. Then, m' is said to be reachable from m by firing t_i, denoted by $m\, [\, t_i > m'$. Moreover, a sequence of transitions $s = t_1 t_2 \cdots t_k$ brings m to marking m'', denoted by $m\, [\, s > m''$, if $(\exists m_1, m_2, \cdots, m_{k-1})(m\, [\, t_1 > m_1\, [\, t_2 > \cdots\ m_{k-1}\, [\, t_k > m'')$. m'' can be computed by $m'' = m + C \cdot \pi(s)$ and denoted by $m\, [\, s > m''$, where $\pi(s) = \sum_{i=1}^{k} \vec{t_i}$ is called the firing vector relative to s. A marking m is reachable in (N, m_0) iff there exists a firing sequence s such that $m_0\, [s >\ m$. The set of all markings reachable from m_0 defines the reachability set of (N, m_0) and is denoted by $R(N, m_0)$.

A PN (N, m_0) is bounded if $(\exists\, b \in \mathbb{N})(\forall m \in R(N, m_0))(\forall p \in P)(m(p) \leq b)$. A PN is live if, $(\forall m \in R(N, m_0))(\forall t \in T)(\exists s \in T^*)(m\, [\, st >)$.

A labeled Petri net (LPN) is a tuple $N_L = (N, m_0, \Sigma, \varphi)$, where (N, m_0) is a marked PN, Σ is a finite set of labels (or events) and $\varphi \colon T \to \Sigma$ is the transition labeling function. φ is also extended to sequences of transitions, $\varphi \colon T^* \to \Sigma^*$. The language generated by N_L is $\mathcal{L}(N_L) = \{\varphi(s) \in \Sigma^* \mid s \in T^*,\ m_0\, [\, s >\}$. For short, we write $\mathcal{L}$ instead of $\mathcal{L}(N_L)$. In addition, various transitions can be associated with the same label. We denote by T_σ the set of transitions associated with label σ, i.e., $T_\sigma = \{t \in T \mid \varphi(t) = \sigma\}$.

2.2 Diagnosability of LPNs

With respect to the partial observability characterizing transitions, the set of transitions can be partitioned as $T = T_o \uplus T_u$, where T_o is the set of observable transitions and T_u the set of unobservable transitions. T_u is also partitioned into two subsets $T_u = T_f \uplus T_{reg}$, where T_f is the set of faulty transitions and T_{reg} the set of regular (non-faulty) transitions. As we deal with LPNs, the event set Σ can also be partitioned into two disjoint sets, $\Sigma = \Sigma_o \uplus \Sigma_u$, where Σ_o is the set of observable events and Σ_u the set of unobservable events. Fault events denoted by set Σ_f are unobservable, thus $\Sigma_f \subseteq \Sigma_u$. Moreover, Σ_u can be partitioned into two disjoint sets, $\Sigma_u = \Sigma_f \uplus \Sigma_{reg}$, where $\Sigma_{reg} = \Sigma_u \backslash \Sigma_f$ is the set of regular (non-faulty) events. In addition, Σ_f can be further partitioned into various fault classes, i.e., $\Sigma_f = \biguplus_{i=1}^{m} \Sigma_{f_i}$, where $\Sigma_{f_i} (i \in \{1, 2, \ldots, m\})$ denotes the i^{th} class of faults.

Let $P_o \colon \Sigma^* \to \Sigma_o^*$ be the projection which *erases* all unobservable events in an event sequence $u \in \Sigma^*$. Given a live and prefix-closed language $\mathcal{L} \subseteq \Sigma^*$, the inverse projection is defined as $P_{\mathcal{L}}^{-1}(v) = \{u \in \mathcal{L} \mid P_o(u) = v\}$ for $v \in \Sigma_o^*$. For an event sequence $u \in \mathcal{L}$, the post-language of $\mathcal{L}$ upon u denoted by $\mathcal{L}/u = \{v \in \Sigma^* \mid uv \in \mathcal{L}\}$. We denote by $|u|$ the length of event sequence u, and u^i the i^{th} event of

u. Also, for $a \in \Sigma$ and $u \in \Sigma^*$, we write $a \in u$ if $(\exists\, i \in \mathbb{N})(u^i = a)$. By abuse of notation, we note $\Sigma_{f_i} \in u$ to indicate that $(\exists j \in \mathbb{N})(u^j \in \Sigma_{f_i})$.

Without loss of generality, we will consider one single fault class Σ_f in the sequel.

Definition 1. (*Diagnosability of LPNs* [2])
A given LPN N_L is diagnosable with regard to (w.r.t.) fault class Σ_f and projection P_o if:

$$(\exists\, n \in \mathbb{N})\ (\forall u \in \mathcal{L}, u^{|u|} \in \Sigma_f)(\forall v \in \mathcal{L}/u)|P_o(v)| \geq n \Rightarrow [\forall \omega \in P_{\mathcal{L}}^{-1}(P_o(uv)) : \Sigma_f \in \omega] \qquad \diamond$$

In other terms, an LPN is diagnosable, if for every trace u ending with a fault event belonging to Σ_f and for any sufficiently long continuation v of u, each trace ω having the same observation as uv contains a fault event. In the current study, we consider the following assumptions:

- the LPN is deadlock-free and bounded;
- the LPN has no executable cycle of unobservable transitions and the faults are permanent.

2.3 Railway Level Crossing Benchmark

The considered benchmark is designed to mimic a railway level crossing (LC) system, which is an intersection where a railway line intersects with a road or path at the same level. An LC consists of three main subsystems: 1) sensors to detect trains' position relative to the LC along each track; 2) barriers to stop road traffic; and 3) a local control system to activate/deactivate the barriers, flashing lights and sound alarm.

The LC benchmark was developed to analyze various diagnosis issues [14]. The associated LPN model is live and bounded. Moreover, while the size of the model grows linearly, its state-space grows exponentially, which is suitable for evaluating the efficiency of an approach.

The operational logic of a multi-track LC considers the railway traffic on each track: 1) the LC is closed to road traffic when at least one train is in the crossing zone; 2) the LC is reopened to road traffic only if no train is in the crossing zone.

For diagnosis purposes, a series of LC benchmarks with two classes of faults are considered. The benchmark consists of n (the number of tracks) railway traffic blocks, an LC controller block and a barrier block (see Figure 1).

The two fault classes which may occur are denoted by red colored transitions in the LPN model (Figure 1). The first one, named T_{f_1}, related to a train-sensing defect is modeled by unobservable transition $(t_{i,4}, ig)$ and indicates that the train may enter the LC zone before the barriers are lowered. The second failure, named T_{f_2}, modeled by unobservable transition (t_6, bf) indicates a defect of the barriers that results in a premature rising. Either of these two faults can induce incorrect operation of the LC control and possibly train-car collisions.

3 Approaches Considered for PN Diagnosis

Three approaches of diagnosis of PN models are considered in the current study. Namely, the MBRG/BRD technique [7], the FMG/FMSG technique [11] and the SSD technique [12, 13]. The first one is the MBRG/BRD approach developed in [7]. Although it is based on the seminal diagnoser-based approach [2], it develops the concept of basis marking to perform both diagnosability analysis and on-line diagnosis, using a subset of reachable markings. More recently, Liu *et al.* developed the FMG/FMSG technique based on on-the-fly analysis of diagnosability with the simultaneously (partial) building of the diagnoser state-space [11]. In general, it allows performing diagnosability analysis and

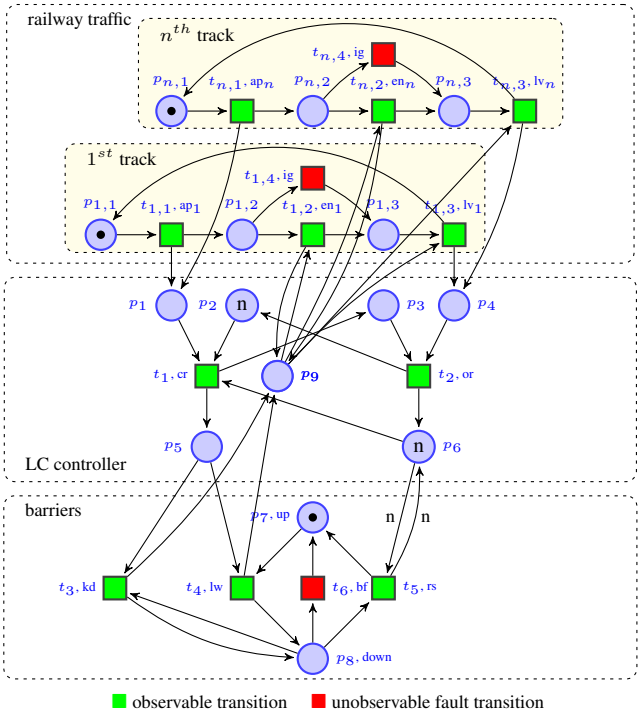

Figure 1: The multi-track level crossing benchmark

on-line diagnosis without building the whole state-space of the diagnoser. Nevertheless, the theoretical complexity is at the same level of the seminal diagnoser-based approach. However, in practice, it shows a good efficiency particularly for non-diagnosable models [15]. In [12, 13], the on-the-fly analysis is adopted while considering a new variant of the diagnoser with a semi-symbolic representation of its state-space using binary decision diagrams (BDDs). The structure of this diagnoser variant shows some interesting features that allow for improving the diagnosability analysis. Moreover, using BDDs to encode the diagnoser nodes brings a considerable gain in terms of memory space and allows for employing efficient operations to handle the diagnoser nodes.

The three techniques show some common features, namely:

- Diagnoser-like structures are built to perform both diagnosability analysis and on-line diagnosis;
- The theoretical complexity is in exponential order of the state-space size of the model;
- They aim to avoid enumerating the whole state-space of the PN to tackle the state explosion problem;
- They have been implemented in dedicated tools, which allow us to make some experimental comparison.

Moreover, some other similarities can be pointed out:

- Both the MBRG/BRD and the FMG/FMSG techniques are based on building intermediate models (MBRG and FMG) as in [2];

- The FMG/FMSG and the SSD approaches analyze diagnosability on the basis of on-the-fly built state-space;
- The MBRG/BRD and the SSD approaches propose compact representation of diagnoser nodes (using basis markings and semi-symbolic representation, respectively).

3.1 MBRG/BRD Approach

The MBRG/BRD approach [7] is a diagnoser-based approach for fault diagnosis of bounded LPNs. It consists in analyzing diagnosability on the basis of two compact models: the so-called MBRG and the BRD. The main idea behind the MBRG/BRD approach is the use of minimal explanations and basis markings [7].

The introduction of minimal explanation aims to only generate necessary markings instead of the whole reachability state-space. Hence, the exhaustive enumeration of the state-space is not required. In order to recall the basic concept of this notion, let us note $T_r = T_o \cup T_f$ and $T_a = T_u \backslash T_f$.

Definition 2. *Given a marking m and a transition $t \in T_r$, we define $\Sigma(m,t) = \{\sigma \in T_a^* | m[\sigma > m', m' \geq Pre(\cdot,t)\}$ as the set of explanations of t at m, and $Y(m,t) = \pi(\Sigma(m,t))$ the corresponding set of e-vectors (or explanation vectors), i.e., the firing vectors associated with the explanations.* ◇

Thus, $\Sigma(m,t)$ is the set of unobservable no-faulty sequences whose firing at m enables t. Among the above sequences, we want to select those whose firing vectors are minimal.

Definition 3. *Given a marking m and a transition $t \in T_r$, we define $\Sigma_{min}(m,t) = \{\sigma \in \Sigma(m,t) | \nexists \sigma_1 \in \Sigma(m,t) : \pi(\sigma_1) \lneqq \pi(\sigma)\}$ as the set of minimal explanations of t at m, and $Y_{min}(m,t) = \pi(\Sigma_{min}(m,t))$ the corresponding set of minimal e-vectors.* ◇

Modified basis reachability graph (MBRG)

Hereafter, we give the definitions of the basis marking and the MBRG as an automaton.

Definition 4. *(Basis marking)*
The initial marking of an LPN is also the initial basis marking $X_b^0 = m_0$. Starting from X_b^0, the reset of basis markings are generated iteratively (until $X_b^{i+1} = X_b^i$) as follow: $X_b^{i+1} = X_b^i \cup \{m' | m \in X_b^i \wedge t \in T_r \wedge \sigma \in \Sigma_{min}(m,t) \wedge m[\sigma t > m'\}$. ◇

The set of basis markings for the MBRG is $Q = X_b^i$.

Definition 5. *(MBRG automaton)*
The MBRG is an automaton $\mathcal{G} = \langle Q, E, \delta_\mathcal{G}, q_0 \rangle$, where $q_0 = x_b^0$, $Q \subset R(N, m_0)$, $E = T_r \times Y_{min}(m,t)$, and $\delta_\mathcal{G} \subset Q \times E \times Q$ denotes the transition relation where $m[\sigma t > m'$, and $\sigma \in \Sigma_{min}(m,t)$ means that $\delta_\mathcal{G}(m,(t,e)) = m'$. ◇

Basis reachability diagnoser (BRD)

The diagnosability analysis using the MBRG/BRD approach is also based on the construction of a graph called BRD, which is a deterministic automaton that, is used in addition whith the MBRG, serves to check the necessary and sufficient condition for diagnosability.

In fact, the BRD is a deterministic graph where each node contains:

- a set of triples (q, v, h), where q is a basis marking, $v \in \{0,1\}$ is a boolean variable indicating the feasible faulty sequences, $h \in \{N, F\}$ indicates if marking q is reachable with a fault transition has occurred or not;
- the diagnosis state of the node represented by $\triangle$.

- the arcs are labeled with observable events from Σ_o.

Diagnosability analysis using the MBRG/BRD technique is based on the necessary and sufficient condition developed in [2]. This conditions is verified using the BRD in conjunction with the MBRG. In particular, one has to check if the BRD contains an uncertain cycle, namely a potential indeterminate cycle, and then using the MBRG to verify if that cycle is indeterminate or not.

3.2 FMG/FMSG Approach

3.2.1 Fault marking graph (FMG)

In order to obtain a finite representation suitable for diagnosability analysis, the FMG is defined to record certain specific markings with their respective fault occurrence information.

Definition 6. *A fault marking (FM) upon a sequence $\sigma \in T^*$ is a vector $x \in \mathbb{N}^{|P|+1}$:*

$$x = \begin{bmatrix} mark(x) \\ tag(x) \end{bmatrix}$$

where $M_0[\,\sigma > mark(x)$, and $tag(x) = 1$ if $(\exists\, j \in \mathbb{N})(\sigma^j \in T_f)$, otherwise $tag(x) = 0$. $\diamond$

For two FMs x and x', we note $x\ [\,\sigma > x'$ iff 1) $mark(x)\ [\,\sigma > mark(x')$, and 2) $tag(x') = 1$ if $(\exists j \in \mathbb{N})(\sigma^j \in T_f)$, otherwise $tag(x') = tag(x)$.

Definition 7. *The fault marking graph (FMG) w.r.t. a given fault class is a 4-tuple $(\mathcal{N}_g, \mathcal{A}_g, delay, x_0)$, where:*

- $x_0 = [M_0^\tau, 0]^\tau$ *is the initial node*[1]*;*
- $\mathcal{N}_g$ *is the set of FMG nodes;*
- $\mathcal{A}_g \subseteq \mathcal{N}_g \times \Sigma_o \times \mathcal{N}_g$ *is the set of directed and labeled arcs.* $(x, e, x') \in \mathcal{A}_g$ *if* $(\exists \sigma \in T_u^* T_o)((x\ [\,\sigma > x') \wedge (P_o(\varphi(\sigma)) = e))$*;*
- $delay : \mathcal{N}_g \to \mathbb{N} \cup \{+\infty\}$ *is the delay function:*

  – $delay(x) = 0$ *if* $tag(x) = 0$*;*

  – *otherwise,*

    * $delay(x) = +\infty$ *if* $(\exists e_1, \cdots, e_k \in \Sigma_o)(\exists x_1, \cdots, x_k,$ $x_{k+1} \in \mathcal{N}_g)[((x_i, e_i, x_{i+1}) \in \mathcal{A}_g) \wedge (x_{k+1} = x_1 = x)]$*;*
    * *otherwise,* $delay(x) = max(\bigcup_{(x_j, e_j, x) \in \mathcal{A}_g} \{delay(x_j)\}) + 1$. $\diamond$

In simple terms, an FMG is a directed non-deterministic graph. Each node indicates an FM and each arc indicates an observable event. Actually, an FMG can be treated as a specific ϵ-reduced observer automaton with tags indicating fault occurrence. Here, $delay$ function is used to determine the maximum number of possible successive fault nodes, which will be used to calculate the $Delay$ value in the FMSG. $delay(x) = +\infty$ implies that x is in a cycle of faulty FMs where the occurrence of a fault can be propagated infinitely.

[1] A^τ denotes the transpose of matrix A.

3.2.2 Fault marking set graph (FMSG)

Based on the FMG, FMSG can be generated for both diagnosability analysis and on-line diagnosis.

A fault marking set (FMS) is a maximal finite set of FMs that are reachable from the initial FMG node by the same observation. Formally, it can be written by: $\{x \in \mathcal{N}_g \mid \exists\ \sigma \in T^*T_o, \sigma \in \mathcal{L}, P_o(\varphi(\sigma)) = s, x_0\ [\ \sigma > x\}$.

Definition 8. *A fault marking set graph (FMSG) is a 5-tuple* $(\mathcal{N}_G, \mathcal{A}_G, Tag, Delay, Y_0)$*, where*

- $Y_0 = \{x_0\}$ *is the initial node;*
- $\mathcal{N}_G$ *is the set of FMS nodes of FMSG;*
- $\mathcal{A}_G \subseteq \mathcal{N}_G \times \Sigma_o \times \mathcal{N}_G$ *is the set of directed and labeled arcs.* $(Y, e, Y') \in \mathcal{A}_G$ *if* $\forall x' \in Y'$, $\exists x \in Y$ *such that* $(x, e, x') \in \mathcal{A}_g$*;*
- tag *is the tagging mapping* $tag\colon \mathcal{N}_G \to \{N, F, U\}$*:*

$$tag(Y) = \begin{cases} N & \text{if } \forall\ x \in Y, tag(x) = 0 \\ F & \text{if } \forall\ x \in Y, tag(x) = 1 \\ U & \text{otherwise} \end{cases}$$

- $Delay\colon \mathcal{N}_G \to \mathbb{N} \cup \{+\infty\}$ *is the delay function:*
  - $Delay(Y) = 0$ *if* $tag(Y) \in \{N, F\}$*, otherwise*
  - $Delay(Y) = max(\bigcup_{x \in Y}\{delay(x)\})$. ◇

An FMS Y is normal (resp. F-certain, F-uncertain) if $tag(Y) = N$ (resp. F, U). $Delay(Y)$ denotes the maximum number of successive F-uncertain FMSs from the first possibly occurred fault until Y. In particular, $Delay(Y) = +\infty$ implies that Y is in an indeterminate cycle and the net is not diagnosable.

The FMSG is a deterministic diagnoser-like graph structure for both diagnosability analysis and on-line diagnosis. For a bounded LPN, both the FMG and the FMSG are finite. In order to perform diagnosability analysis efficiently, one does not necessarily build the whole FMG and FMSG in this approach.

3.3 SSD Approach

The technique is based on the semi-symbolic diagnoser (SSD) computed on the fly, the nodes of which are encoded using *binary decision diagrams* (BDDs) while the transitions linking the nodes are represented *explicitly*. In order to define the SSD more formally, we introduce the following notations:

- Given a subset of transitions $T' \subseteq T$, we define $Enable_{T'}(m) = \{t \in T' \mid m\ [\ t >\}$, as the set of transitions in T' that are enabled at marking m. The extension to a subset of markings $M' \subseteq R(N, m_0)$, is $Enable_{T'}(M') = \bigcup_{m \in M'} Enable_{T'}(m)$.
- Given a subset of markings $M \subseteq R(N, m_0)$ and a transition $t \in T$, we define $Img(M, t) = \{m' \in R(N, m_0) \mid \exists m \in M : m\ [\ t >\ m'\}$ as the set of markings reachable from the markings in M by firing transition t. The generalization to a subset of transitions $T' \subseteq T$ is $Img(M', T') = \bigcup_{t \in T'} Img(M', t)$.
- Given a marking $m \in R(N, m_0)$ and a subset of transition $T' \subseteq T$, we define $Reach_{T'}(m) = \{m\} \cup \{m' \in R(N, m_0) | (\exists s \in T'^*) : m\ [\ s >\ m'\}$ as the set of markings reached by firing a sequence of transitions in T' from marking m. The generalization to a subset of markings $M \subseteq R(N, m_0)$ is $Reach_{T'}(M) = \bigcup_{m \in M} Reach_{T'}(m)$.

3.3.1 The Structure of the diagnoser node

Each node in the SSD is partitioned into two distinct subsets of markings, each of them is encoded using a BDD.

1. *the set of normal markings* (denoted by $\mathcal{M}_N$), which is the subset of markings in the node that are reachable by firing faulty-free sequences.

2. *the set of faulty markings* (denoted by $\mathcal{M}_F$), which is the subset of markings in the node that are reachable by firing faulty sequences.

There may exist some faulty transitions that link some markings in $\mathcal{M}_N$ to some others in $\mathcal{M}_F$ within the same node. The existence of such transitions is also encoded within each node using a boolean variable.Actually, such a node structure can be advantageously explored for rendering diagnosability analysis more efficiently than using the classic structure of diagnosers [7, 16].

To simplify the notation, we use $a.\mathcal{M}_N$ (resp. $a.\mathcal{M}_F$) to indicate the set of normal markings $\mathcal{M}_N$ (resp. set of faulty markings $\mathcal{M}_F$) of a given diagnoser node a.

Definition 9. *(Semi-Symbolic Diagnoser (SSD))*
The SSD associated with an LPN N_L is a directed deterministic graph $\mathcal{D} = \langle \Gamma, \Sigma_o, \delta_{\mathcal{D}}, \Gamma_0 \rangle$, where:

1. Γ *is a finite set of diagnoser nodes;*

2. Σ_o *is a finite set of events associated with a finite set of observable transitions T_o;*

3. Γ_0 *is the initial diagnoser node with:*

   a) $\Gamma_0.\mathcal{M}_N = Reach_{T_{reg}}(m_0)$;

   b) $\Gamma_0.\mathcal{M}_F = Reach_{T_u}(Img(\Gamma_0.\mathcal{M}_N, T_f))$.

4. $\delta_{\mathcal{D}} : \Gamma \times \Sigma_o \rightarrow \Gamma$ *is the transition relation, defined as follows:* $\forall a, a' \in \Gamma$, $\sigma \in \Sigma_o$: $a' = \delta_{\mathcal{D}}(a, \sigma) \Leftrightarrow$
   $a'.\mathcal{M}_N = Reach_{T_{reg}}(Img(a.\mathcal{M}_N, T_\sigma)) \wedge$
   $a'.\mathcal{M}_F = Reach_{T_u}(Img(a'.\mathcal{M}_N, T_f) \cup Img(a.\mathcal{M}_F, T_\sigma))$

Since all the successors of an F-$certain$ node are also F-$certain$, it is unnecessary to build them because they do not bring new information from the diagnosis point of view. Indeed, as regards diagnosability analysis, and given the necessary and sufficient condition of diagnosability established in [2], only the analysis of F-uncertain cycles is necessary.

Table 1: Features of the approaches considered

Features	MBRG/BRD	FMG/FMSG	SSD
Representation	basis markings	fault markings	symbolic markings
The generator	MBRG	FMG	no generator
Double-check	yes	yes	no
On-the-fly analysis	no	yes	yes

Diagnosability Analysis

In the SSD approach, the necessary and sufficient condition for diagnosability is established on the basis of the notion of '*indicating sequence*', which is associated with the F-uncertain cycles.

Definition 10. *($c\ell$-indicating sequence)*
Let $c\ell = a_1, a_2, \ldots, a_n$ be an F-uncertain cycle in $\mathcal{D}$ (the starting node a_1 can be arbitrarily chosen in the cycle), with $\delta_{\mathcal{D}}(a_i, \sigma_i) = a_{(i+1)mod_n}$ for $1 \leq i \leq n$. $c\ell$-indicating sequence $\rho^{c\ell} = \mathcal{S}_1, \mathcal{S}_2, \ldots$, is an infinite sequence of sets of markings, such that:
– $\mathcal{S}_1 = a_1.\mathcal{M}_F$;
– $\forall\, i > 1 :\ \mathcal{S}_i = Reach_{T_u}(Img(\mathcal{S}_{i-1}, T_{\sigma_{(i-1)_{mod_n}}}))$; ◇

In fact, the $c\ell$-indicating sequence tracks the subsets of faulty markings in each node of $c\ell$ without considering the faulty markings generated through the occurrence of some faulty transitions outgoing from the normal set of markings in the traversed nodes (except for $\mathcal{S}_1$ which holds all the faulty states of $a_1.\mathcal{M}_F$, i.e., $\mathcal{S}_1 = a_1.\mathcal{M}_F$).

Actually, the $c\ell$-indicating sequence is introduced with the aim of tracking the actual faulty cycles corresponding to a given F-*uncertain* cycle, if such cycles exist in the original model.

Theorem 1. *For an F-uncertain cycle $c\ell = a_1, a_2, \ldots, a_n$ in $\mathcal{D}$, and $\rho^{c\ell} = \mathcal{S}_1, \mathcal{S}_2, \ldots$ its corresponding $c\ell$-indicating sequence. Then, $c\ell$ is an F-indeterminate cycle if and only if: $\forall i \in \mathbb{N}^* : \mathcal{S}_i \neq \emptyset$.* ◇

Actually, Theorem 1 states that an F-uncertain cycle is an F-indeterminate one if the $c\ell$-indicating sequence does not reach an empty fixed-point.

For the actual verification of diagnosability, *a systematic procedure* is derived directly from Theorem 1 and can be performed as follows:

When an F-uncertain cycle $c\ell$ is found in SSD $\mathcal{D}$, then:

1. generate the successive elements of $c\ell$-indicating sequence $\rho^{c\ell}$ (starting from $\mathcal{S}_1$), and for each element $\mathcal{S}_i$ check the following conditions:
   (a) if $\mathcal{S}_i = \emptyset$, then cycle $c\ell$ is not an F-indeterminate cycle and therefore the procedure is stopped;
   (b) else, if $\mathcal{S}_i \neq \emptyset$ and $\exists k \in \mathbb{N} : i = 1 + kn$ (with $n = |c\ell|$), then:
      i. if $\mathcal{S}_i = \mathcal{S}_{(i-n)}$, then cycle $c\ell$ is an F-indeterminate cycle and stop the procedure;
      ii. else continue.

This procedure is repeated on each F-uncertain cycle generated on the fly in $\mathcal{D}$.

4 Discussion

4.1 Features of Approaches Considered

The main feature of the MBRG/BRD approach is the use of minimal explanations and basis markings, which allow to represent the reachability state-space in a compact manner. However, the MBRG is generally, but not necessarily, smaller than the reachability graph.

The FMG/FMSG approach solves four problems (K- diagnosability, diagnosability, the minimum value K_{min} ensuring K-diagnosability and on-line diagnosis) using the same structures, i.e., the FMG and the FMSG. The idea behind is to solve these diagnostic problems on generating necessary state-space in an incremental way.

The SSD approach, combining the advantages of the former two approaches, uses a compact representation of the diagnoser state-space based on the semi-symbolic representation. It represents the diagnoser nodes symbolically, while the arcs remain in an explicit manner. Moreover, constructing the diagnoser and checking diagnosability are simultaneously performed on the fly. Generally, this avoids building the whole diagnoser. In addition, unlike in other approaches, the structure of the diagnoser in this approach provides some interesting features that allow for checking the necessary and sufficient condition for diagnosability without building any intermediate model. Some main features of these approaches are summarized in Table 1.

4.2 Tools Implementing the Approaches

The MBRG/BRD approach is implemented in a MATLAB Toolbox called PN_Diag [17, 18]. The tool takes PN models in mathematical representation (i.e., matrix form) as input (a MATLAB file). Firstly, it generates the MBRG (on the based of the BRG), and then generates the BRD (on the based of the BRG as well), and finally, checks the diagnosability. In the case of non-diagnosable models, PN_Diag outputs all the event-traces corresponding to F-indeterminate cycles in the BRD.

The FMG/FMSG approach is implemented in OF-PENDA tool [19], which is a software tool developed in C++ programming language. OF-PENDA takes the mathematical model of an LPN as input.The tool is based on the on-the-fly and incremental analysis of K-diagnosability. Therefore, the generated model for the K^{th} step is used for $(K+1)^{th}$ step. If the system is diagnosable, the minimum value K_{min} ensuring K-diagnosability is given, and the online diagnoser is generated.

The SSD approach is implemented in DPN-SOG tool, which is a command-line software tool developed in C++ programming language. DPN-SOG takes as inputs: (*i*) the LPN models in prod format [20], (*ii*) the bound k of the net as an integer and (*iii*) a text file which specifies the sets of observable, non-observable and faulty transitions. Using these ingredients, DPN-SOG builds on the fly the SSD and simultaneously analyzes the diagnosability. When the LPN model is non-diagnosable, DPN-SOG outputs the generated part of the diagnoser as well as a witnessed diagnoser event-trace that violated the diagnosability property (the first encountered event-trace). When the LPN model is stated to be diagnosable, DPN-SOG generates the part of the diagnoser that is sufficient to perform the online diagnosis. Further complementary information that helps the evaluation of the approach can be output, namely, the required CPU time, the size of the diagnoser, the number of used BDD nodes and memory size of the diagnoser (in terms of kilobytes).

5 Experimentations

The three techniques are applied to analyze the diagnosability of the LC benchmark, respectively. The experiments are performed on a 64-bit PC (CPU: Intel Core i5, 2.5 GHz; RAM: 6 GB), and the results are summarized in Table 2. Regarding the obtained results, the following remarks can be underlined:

1. In non-diagnosable cases, one can observe that the SSD and the FMG/FMSG approaches efficiently analyze the diagnosability by only constructing the relevant part of the diagnosers, which reduces the memory/time consumption. Actually, while the PN_Diag tool blocks from $n = 4$, both OF-PENDA and DPN-SOG tools provide the diagnosability verdict in a few seconds even for large values of n. This is due to the on-the-fly analysis which allows for performing the diagnosability analysis based on a partial building of the diagnoser. In other terms, the state-space generation as well as the analysis is stopped as soon as an indeterminate cycle is found. This is an interesting feature especially when we deal with large models.

2. In the case of diagnosable cases, i.e., when the barriers failure (T_{F_2}) is considered, the two approaches implementing an on-the-fly procedure potentially have to construct a larger part of the diagnoser state-space. Consequently, the verification process checks all the F-uncertain cycles of the diagnosers to analyze the diagnosability of the models. In this case, the obtained results clearly reflect the exponential feature of the diagnoser-based approaches. Actually, this is unavoidable when working with diagnoser-based approaches, since it is due to the deterministic nature of the diagnoser.

3. Compared with the other tools, the DPN-SOG tool provides better results on the cases considered. In particular, the DPN-SOG tool successes to analyze the diagnosability of all the models considered, while the other tools fail as early as from 5 tracks. This can be explained through two main points:

   (a) The SSD approach only constructs one graph since diagnosability analysis is performed directly on the diagnoser;

   (b) The systematic procedure for checking the necessary and sufficient condition based on the SSD allows for reducing the verification time;

   (c) Besides the gain in terms of memory, the use of BDDs for representing the diagnoser nodes allows for using efficient techniques to generate the diagnoser nodes and perform the verification.

Table 2: Comparative experimental results

n	Petri net features				BRD			FMSG			SSD			Diag.	T_{F_i}
	$\|P\|$	$\|T\|$	$\|\mathcal{N}\|$	$\|\mathcal{A}\|$	$\|BRD_S\|$	$\|BRD_T\|$	BRD_e (s)	$\|FMSG_S\|$	$\|FMSG_T\|$	$FMSG_e$ (s)	$\|SSD_S\|$	$\|SSD_T\|$	SSD_e (s)		
1	12	10	20	43	21	31	0.1	12	22	$\leq$ 0.1	10	14	$\leq$ 0.1	yes	
2	15	14	142	500	168	454	1.3	25	27	$\leq$ 0.1	9	8	$\leq$ 0.1	no	
3	18	18	832	4085	967	3845	46.0	32	34	$\leq$ 0.1	10	9	$\leq$ 0.1	no	
4	21	22	4314	27142	4869	25740	1477.0	39	41	$\leq$ 0.1	11	10	$\leq$ 0.1	no	T_{F_1}
5	24	26	20556	157551	*	*	o.t.	46	48	$\leq$ 0.1	12	11	$\leq$ 0.1	no	
6	27	30	92070	831384	*	*	o.t.	53	55	$\leq$ 0.1	13	12	$\leq$ 0.1	no	
7	30	34	393336	4086585	*	*	o.t.	60	62	$\leq$ 0.1	14	13	$\leq$ 0.1	no	
1	12	10	20	43	21	31	0.1	17	39	$\leq$ 0.1	10	14	$\leq$ 0.1	yes	
2	15	14	142	500	167	454	1.2	176	661	$\leq$ 0.1	83	205	$\leq$ 0.1	yes	
3	18	18	832	4085	967	3845	38.5	1192	6436	0.6	483	1745	$\leq$ 0.2	yes	
4	21	22	4314	27142	4896	25740	1099.5	8192	58574	47.1	2434	11774	0.7	yes	T_{F_2}
5	24	26	20556	157551	*	*	o.t.	*	*	o.t.	11304	69112	12.4	yes	
6	27	30	92070	831384	*	*	o.t.	*	*	o.t.	56136	414299	236	yes	
7	30	34	393336	4086585	*	*	o.t.	*	*	o.t.	261262	2282890	5822	yes	

*: No result obtained in 4 hours. o.t.: Out of time (more than 4 hours).

- n: the number of tracks;
- $|P|$ and $|T|$: the number of places and transitions in the PN models, respectively;
- $|\mathcal{N}|$ and $|\mathcal{A}|$: the number of nodes and arcs in the reachability graph, respectively;
- $BRD_S|$ and $|BDR_T|$: the numbers of nodes and arcs in the BRD, respectively;
- $|FMSG_S|$ and $|FMSG_T|$: the numbers of nodes and arcs in the FMSG, respectively;
- $|SSD_S|$ and $|SSD_T|$: the numbers of nodes and arcs in the SSD, respectively;
- BRD_e, $FMSG_e$ and SSD_e: the time required to generated the models and checking diagnosability;
- 'Diag.': the diagnosability verdict.

5.1 Combining the Three Approaches

As a conclusion, it may be interesting to develop an approach to incorporate the main features of the three approaches. The minimal explanation in the MBRG/BRD approach can be used to generate the reachability state-space in a compact manner. Then, diagnoser nodes can be composed of only basis markings. In addition, each diagnoser node can be split into two sets of basis markings: the set of normal basis markings and the set of faulty basis markings. Both sets are encoded in a symbolic manner using BDDs, as in the SSD approach. The diagnoser construction and the diagnosability analysis will be simultaneously performed with an on-the-fly depth-first search procedure, as in the FMG/FMSG approach. Such a combination may improve the efficiency of diagnosability analysis and further reduce the time/memory consumption.

6 Conclusion

This paper compares three diagnoser-based techniques for diagnosability analysis of LPNs, namely the MBRG/BRD approach, the FMG/FMSG approach and the SSD approach. A railway level crossing benchmark, which shows both diagnosable and non-diagnosable faults, is chosen as an experimental benchmark. Three software tools associated with the considered approaches are used for the analysis. The approaches are evaluated particularly in terms of the generated state-space of advanced models and the time required for constructing such models and analyzing diagnosability. In the future, we intend to develop an approach that combines the main features of three approaches.

References

[1] C. Cassandras and S. Lafortune, "Introduction to discrete event systems," *Springer*, 2008.

[2] M. Sampath, R. Sengupta, and S. Lafortune, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, 1995.

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

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

[5] S. Hashtrudi Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in discrete-event systems: framework and model reduction," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199–1212, 2003.

[6] T. Ushio, I. Onishi, and K. Okuda, "Fault detection based on Petri net models with faulty behaviors," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 113–118, 1998.

[7] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of bounded Petri nets," in *Proceedings of the 48th IEEE Conference on Decision and Control, held jointly with the 28th Chinese Control Conference*, pp. 1254–1260, 2009.

[8] F. Basile, P. Chiacchio, and G. De Tommasi, "Diagnosability of labeled Petri nets via integer linear programming," in *10th International Workshop on Discrete Event Systems*, pp. 71–77, 2010.

[9] D. Lefebvre and E. Leclercq, "Diagnosability of Petri nets with observation graphs," *Discrete Event Dynamic Systems*, vol. 26, no. 3, pp. 539–559, 2016.

[10] F. Basile, "Overview of fault diagnosis methods based on Petri net models," in *13th European Control Conference*, pp. 2636–2642, 2014.

[11] B. Liu, M. Ghazel, and A. Toguyéni, "On-the-fly and incremental technique for fault diagnosis of discrete event systems modeled by labeled Petri nets," *Asian Journal of Control*, vol. 20, no. 1, pp. 1–13, 2017.

[12] A. Boussif, M. Ghazel, and K. Klai, "Combining enumerative and symbolic techniques for diagnosis of discrete-event systems," in *9th International Workshop on Verification and Evaluation of Computer and Communication Systems*, pp. 23–34, 2015.

[13] A. Boussif, *Contributions to fault diagnosis of discrete-event systems*. PhD thesis, Université Lille Nord de France, University of Lille 1, Villeneuve d'Ascq, France, 2016.

[14] M. Ghazel and B. Liu, "A customizable railway benchmark to deal with fault diagnosis issues in DES," in *13th International Workshop on Discrete Event Systems*, pp. 177–182, IEEE, 2016.

[15] B. Liu, M. Ghazel, and A. Toguyéni, "Model-based diagnosis of multi-track level crossing plants," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 2, pp. 546–556, 2016.

[16] B. Liu, M. Ghazel, and A. Toguyéni, "Toward an efficient approach for diagnosability analysis of DES modeled by labeled Petri nets," in *13th European Control Conference*, pp. 1293–1298, 2014.

[17] M. Pocci, "A toolbox for diagnosability of Petri nets," *http://www.diee.unica.it/giua/TESI/09_Marco.Pocci*, 2011.

[18] M. P. Cabasino, A. Giua, L. Marcias, and C. Seatzu, "A comparison among tools for the diagnosability of discrete event systems," in *8th IEEE International Conference on Automation Science and Engineering*, pp. 218–223, 2012.

[19] B. Liu, M. Ghazel, and A. Toguyéni, "OF-PENDA: A software tool for fault diagnosis of discrete event systems modeled by labeled Petri nets.," in *1st International Workshop Petri Nets for Adaptive Discrete-Event Control Systems*, pp. 20–35, 2014.

[20] K. Varpaaniemi, K. Heljanko, and J. Lilius, "Prod 3.2 an advanced tool for efficient reachability analysis," in *International Conference on Computer Aided Verification*, pp. 472–475, Springer, 1997.