DISCRETE EVENT DIAGNOSIS USING PETRI NETS

Maria Paola Cabasino, Alessandro Giua and Carla Seatzu
Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy
{*cabasino, giua, seatzu*}*@diee.unica.it*

Keywords: Petri nets, Diagnosis, Discrete event systems.

Abstract: This paper serves as a support for the plenary address given by the second author during the conference. In this paper we present an approach to on-line diagnosis of discrete event systems based on labeled Petri nets, that are a particular class of Petri nets where some events are undistinguishable, i.e., events that produce an output signal that is observable, but that is common to other events. Our approach is based on the notion of basis markings and justifications and it can be applied both to bounded and unbounded Petri nets whose unobservable subnet is acyclic. Moreover it is shown that, in the case of bounded Petri nets, the most burdensome part of the procedure may be moved off-line, computing a particular graph that we call *Basis Reachability Graph*.
Finally we present a diagnosis MATLAB toolbox with some examples of application.

1 INTRODUCTION

Failure detection and isolation in industrial systems is a subject that has received a lot of attention in the past few decades. A failure is defined to be any deviation of a system from its normal or intended behavior. Diagnosis is the process of detecting an abnormality in the system behavior and isolating the cause or the source of this abnormality.

Failures are inevitable in today's complex industrial environment and they could arise from several sources such as design errors, equipment malfunctions, operator mistakes, and so on. As technology advances, as we continue to build systems of increasing size and functionality, and as we continue to place increasing demands on the performance of these systems, then so do we increase the complexity of these systems. Consequently (and unfortunately), we enhance the potential for systems to fail, and no matter how safe our designs are, how improved our quality control techniques are, and how better trained the operators are, system failures become unavoidable.

Given the fact that failures are inevitable, the need for effective means of detecting them is quite apparent if we consider their consequences and impacts not just on the systems involved but on the society as a whole. Moreover we note that effective methods of failure diagnosis can not only help avoid the undesirable effects of failures, but can also enhance the operational goals of industries. Improved quality of performance, product integrity and reliability, and reduced cost of equipment maintenance and service are some major benefits that accurate diagnosis schemes can provide, especially for service and product oriented industries such as home and building environment control, office automation, automobile manufacturing, and semiconductor manufacturing. Thus, we see that accurate and timely methods of failure diagnosis can enhance the safety, reliability, availability, quality, and economy of industrial processes.

The need of automated mechanisms for the timely and accurate diagnosis of failures is well understood and appreciated both in industry and in academia. A great deal of research effort has been and is being spent in the design and development of automated diagnostic systems, and a variety of schemes, differing both in their theoretical framework and in their design and implementation philosophy, have been proposed.

In diagnosis approach two different problems can be solved: the problem of diagnosis and the problem of diagnosability.

Solving a problem of diagnosis means that we associate to each observed string of events a diagnosis state, such as "normal" or "faulty" or "uncertain". Solving a problem of diagnosability is equivalent to determine if the system is diagnosable, i.e., to determine if, once a fault has occurred, the system can detect its occurrence in a finite number of steps.

The diagnosis of discrete event systems (DES) is a research area that has received a lot of attention in the last years and has been motivated by the practical need of ensuring the correct and safe functioning of large complex systems. As discussed in the next session the first results have been presented within the framework of automata. More recently, the diagnosis problem has also been addressed using Petri nets (PNs). In fact, the use of Petri nets offers significant advantages because of their twofold representation: graphical and mathematical. Moreover, the intrinsically distributed nature of PNs where the notion of state (i.e., marking) and action (i.e., transition) is local reduces the computational complexity involved in solving a diagnosis problem.

In this paper we summarize our main contributions on diagnosis of DES using PNs (Giua and Seatzu, 2005; Cabasino et al., 2008; Lai et al., 2008; Cabasino et al., 2009). In particular, we focus on arbitrary labeled PNs where the observable events are the labels associated to transitions, while faults are modeled as silent transitions. We assume that there may also be transitions modeling a regular behavior, that are silent as well. Moreover, two or more transitions that may be simultaneously enabled may share the same label, thus they are undistinguishable. Our diagnosis approach is based on the definition of four diagnosis states modeling different degrees of alarm and it applies to all systems whose unobservable subnet is acyclic. Two are the main advantages of our procedure. First, we do not need an exhaustive enumeration of the states in which the system may be: this is due to the introduction of basis markings. Secondly, in the case of bounded net systems we can move off-line the most burdensome part of the procedure building a finite graph called basis reachability graph.

The paper is organized as follows. In Section 2 the state of art of diagnosis for discrete event systems is illustrated. In Section 3 we provide a background on PNs. In Sections 4 and 5 are introduced the definitions of minimal explanations, justifications and basis markings, that are the basic notions of our diagnosis approach. In Section 6 the diagnosis states are defined and a characterization of them in terms of basis markings and j-vectors is given. In Section 7 we show how the most burdensome part of the procedure can be moved offline in the case of bounded PNs. In Section 8 we present the MATLAB toolbox developed by our group for PNs diagnosis and in Section 9 we present some numerical results obtained applying our tool to a parametric model of manufacturing system. In Section 10 we draw the conclusions.

2 LITERATURE REVIEW

In this section we present the state of art of diagnosis of DES using automata and PNs.

2.1 Diagnosis of DES using Automata

In the contest of DES several original theoretical approaches have been proposed using *automata*.

In (Lin, 1994) and (Lin et al., 1993) a state-based DES approach to failure diagnosis is proposed. The problems of off-line and on-line diagnosis are addressed separately and notions of diagnosability in both of these cases are presented. The authors give an algorithm for computing a diagnostic control, i.e., a sequence of test commands for diagnosing system failures. This algorithm is guaranteed to converge if the system satisfies the conditions for on-line diagnosability.

In (Sampath et al., 1995) and (Sampath et al., 1996) the authors propose an approach to failure diagnosis where the system is modeled as a DES in which the failures are treated as unobservable events. The level of detail in a discrete event model appears to be quite adequate for a large class of systems and for a wide variety of failures to be diagnosed. The approach is applicable whenever failures cause a distinct change in the system status but do not necessarily bring the system to a halt. In (Sampath et al., 1995) a definition of diagnosability in the framework of formal languages is provided and necessary and sufficient conditions for diagnosability of systems are established. Also presented in (Sampath et al., 1995) is a systematic approach to solve the problem of diagnosis using diagnosers.

In (Sampath et al., 1998) the authors present an integrated approach to control and diagnosis. More specifically, authors present an approach for the design of diagnosable systems by appropriate design of the system controller and this approach is called active diagnosis. They formulate the active diagnosis problem as a supervisory control problem. The adopted procedure for solving the active diagnosis problem is the following: given the non-diagnosable language generated by the system of interest, they first select an “appropriate” sublanguage of this language as the legal language. Choice of the legal language is a design issue and typically depends on considerations such as acceptable system behavior (which ensures that the system behavior is not restricted more than necessary in order to eventually make it diagnosable) and detection delay for the failures. Once the appropriate legal language is chosen, they then design a controller (diagnostic controller), that achieves a

closed-loop language that is within the legal language and is diagnosable. This controller is designed based on the formal framework and the synthesis techniques that supervisory control theory provides, with the additional constraint of diagnosability.

In (Debouk et al., 2000) is addressed the problem of failure diagnosis in DES with decentralized information. Debouk *et al.* propose a coordinated decentralized architecture consisting of two local sites communicating with a coordinator that is responsible for diagnosing the failures occurring in the system. They extend the notion of diagnosability, originally introduced in (Sampath et al., 1995) for centralized systems, to the proposed coordinated decentralized architecture. In particular, they specify three protocols that realize the proposed architecture and analyze the diagnostic properties of these protocols.

In (Boel and van Schuppen, 2002) the authors address the problem of synthesizing communication protocols and failure diagnosis algorithms for decentralized failure diagnosis of DES with costly communication between diagnosers. The costs on the communication channels may be described in terms of bits and complexity. The costs of communication and computation force the trade-off between the control objective of failure diagnosis and that of minimization of the costs of communication and computation. The results of this paper is an algorithm for decentralized failure diagnosis of DES for the special case of only two diagnosers.

In (Zad et al., 2003) a state-based approach for on-line passive fault diagnosis is presented. In this framework, the system and the diagnoser (the fault detection system) do not have to be initialized at the same time. Furthermore, no information about the state or even the condition (failure status) of the system before the initiation of diagnosis is required. The design of the fault detection system, in the worst case, has exponential complexity. A model reduction scheme with polynomial time complexity is introduced to reduce the computational complexity of the design. Diagnosability of failures is studied, and necessary and sufficient conditions for failure diagnosability are derived.

2.2 Diagnosis of DES using Petri Nets

Among the first pioneer works dealing with PNs, we recall the approach of Prock. In (Prock, 1991) the author proposes an on-line technique for fault detection that is based on monitoring the number of tokens residing into P-invariants: when the number of tokens inside P-invariants changes, then the error is detected.

In (Sreenivas and Jafari, 1993) the authors employ time PNs to model the DES controller and backfiring transitions to determine whether a given state is invalid. Later on, time PNs have been employed in (Ghazel et al., 2005) to propose a monitoring approach for DES with unobservable events and to represent the "a priori" known behavior of the system, and track on-line its state to identify the events that occur.

In (Hadjicostis and Veghese, 1999) the authors use PN models to introduce redundancy into the system and additional P-invariants allow the detection and isolation of faulty markings.

Redundancy into a given PN is used in (Wu and Hadjicostis, 2005) to enable fault detection and identification using algebraic decoding techniques. In this paper Wu and Hadjicostis consider two types of faults: place faults that corrupt the net marking, and transition faults that cause a not correct update of the marking after event occurrence. Although this approach is general, the net marking has to be periodically observable even if unobservable events occur. Analogously, in (Lefebvre and Delherm, 2007) the authors investigate on the determination of the set of places that must be observed for the exact and immediate estimation of faults occurrence.

In (Ruiz-Beltràn et al., 2007) Interpreted PNs are employed to model the system behavior that includes both events and states partially observable. Based on the Interpreted PN model derived from an on-line methodology, a scheme utilizing a solution of a programming problem is proposed to solve the problem of diagnosis.

Note that, all papers in this topic assume that faults are modeled by unobservable transitions. However, while the above mentioned papers assume that the marking of certain places may be observed, a series of papers have been recently presented that are based on the assumption that no place is observable (Basile et al., 2008; Benveniste et al., 2003; Dotoli et al., 2008; Genc and Lafortune, 2007).

In particular, in (Genc and Lafortune, 2007) the authors propose a diagnoser on the basis of a modular approach that performs the diagnosis of faults in each module. Subsequently, the diagnosers recover the monolithic diagnosis information obtained when all the modules are combined into a single module that preserves the behavior of the underlying modular system. A communication system connects the different modules and updates the diagnosis information. Even if the approach does not avoid the state explosion problem, an improvement is obtained when the system can be modeled as a collection of PN modules coupled through common places.

The main advantage of the approaches in (Genc

and Lafortune, 2007) consists in the fact that, if the net is bounded, the diagnoser may be constructed off-line, thus moving off-line the most burdensome part of the procedure. Nevertheless, a characterization of the set of markings consistent with the actual observation is needed. Thus, large memory may be required.

An improvement in this respect has been given in (Benveniste et al., 2003; Basile et al., 2008; Dotoli et al., 2008).

In particular, in (Benveniste et al., 2003) a net unfolding approach for designing an on-line asynchronous diagnoser is used. The state explosion is avoided but the on-line computation can be high due to the on-line building of the PN structures by means of the unfolding.

In (Basile et al., 2008) the diagnoser is built on-line by defining and solving Integer Linear Programming (ILP) problems. Assuming that the fault transitions are not observable, the net marking is computed by the state equation and, if the marking has negative components, an unobservable sequence is occurred. The linear programming solution provides the sequence and detects the fault occurrences. Moreover, an off-line analysis of the PN structure reduces the computational complexity of the ILP problem.

In (Dotoli et al., 2008), in order to avoid the redesign and the redefinition of the diagnoser when the structure of the system changes, the authors propose a diagnoser that works on-line. In particular, it waits for an observable event and an algorithm decides whether the system behavior is normal or may exhibit some possible faults. To this aim, some ILP problems are defined and provide eventually the minimal sequences of unobservable transitions containing the faults that may have occurred. The proposed approach is a general technique since no assumption is imposed on the reachable state set that can be unlimited, and only few properties must be fulfilled by the structure of the PN modeling the system fault behavior.

We also proposed a series of contributions dealing with diagnosis of PNs (Giua and Seatzu, 2005; Cabasino et al., 2008; Lai et al., 2008; Cabasino et al., 2009). Our main results are summarized in the rest of the paper.

Note that none of the above mentioned papers regarding PNs deal with *diagnosability*, namely none of them provide a procedure to determine a priori if a system is *diagnosable*, i.e., if it is possible to reconstruct the occurrence of fault events observing words of finite length.

In fact, whereas this problem has been extensively studied within the framework of automata as discussed above, in the PN framework very few results have been presented.

The first contribution on diagnosability of PNs was given in (Ushio et al., 1998). They extend a necessary and sufficient condition for diagnosability in (Sampath et al., 1995; Sampath et al., 1996) to unbounded PN. They assume that the set of places is partitioned into observable and unobservable places, while all transitions are unobservable in the sense that their occurrences cannot be observed. Starting from the PN they build a diagnoser called *simple* ω *diagnoser* that gives them sufficient conditions for diagnosability of unbounded PNs.

In (Chung, 2005) the authors, in contrast with Ushio's paper, assumes that part of the transitions of the PN modelling is observable and shows as the additional information from observed transitions in general adds diagnosability to the analysed system. Moreover starting from the diagnoser he proposes an automaton called *verifier* that allows a polynomial check mechanism on diagnosability but for finite state automata models.

In (Wen and Jeng, 2005) the authors propose an approach to test diagnosability by checking the structure property of T-invariant of the nets. They use Ushio's diagnoser to prove that their method is correct, however they don't construct a diagnoser for the system to do diagnosis. In (Wen et al., 2005) they also present an algorithm, based on a linear programming problem, of polynomial complexity in the number of nodes for computing a sufficient condition of diagnosability of DES modeled by PN.

3 BACKGROUND

In this section we recall the formalism used in the paper. For more details on PNs we refer to (Murata, 1989).

A *Place/Transition net* (P/T net) is a structure $N = (P, T, Pre, Post)$, where P is a set of m places; T is a set of n transitions; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the *pre–* and *post–* incidence functions that specify the arcs; $C = Post - Pre$ is the incidence matrix.

A *marking* is a vector $M : P \to \mathbb{N}$ that assigns to each place of a P/T net a non–negative integer number of tokens, represented by black dots. We denote $M(p)$ the marking of place p. A P/T *system* or *net system* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0. A transition t is enabled at M iff $M \geq Pre(\cdot, t)$ and may fire yielding the marking $M' = M + C(\cdot, t)$. We write $M[\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_{j_1} \cdots t_{j_k}$ is enabled at M, and we write $M[\sigma\rangle M'$ to denote that the firing of σ yields M'. We also write $t \in \sigma$ to denote that a transition t is contained in σ.

The set of all sequences that are enabled at the initial marking M_0 is denoted $L(N,M_0)$, i.e., $L(N,M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}$.

Given a sequence $\sigma \in T^*$, we call $\pi : T^* \to \mathbb{N}^n$ the function that associates to σ a vector $y \in \mathbb{N}^n$, named the *firing vector* of σ. In particular, $y = \pi(\sigma)$ is such that $y(t) = k$ if the transition t is contained k times in σ.

A marking M is *reachable* in $\langle N,M_0\rangle$ iff there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. The set of all markings reachable from M_0 defines the *reachability set* of $\langle N,M_0\rangle$ and is denoted $R(N,M_0)$.

A PN having no directed circuits is called *acyclic*. A net system $\langle N,M_0\rangle$ is *bounded* if there exists a positive constant k such that, for $M \in R(N,M_0)$, $M(p) \leq k$.

A *labeling function* $\mathcal{L} : T \to L \cup \{\varepsilon\}$ assigns to each transition $t \in T$ either a symbol from a given alphabet L or the empty string ε.

We denote as T_u the set of transitions whose label is ε, i.e., $T_u = \{t \in T \mid \mathcal{L}(t) = \varepsilon\}$. Transitions in T_u are called *unobservable* or *silent*. We denote as T_o the set of transitions labeled with a symbol in L. Transitions in T_o are called *observable* because when they fire their label can be observed. Note that in this paper we assume that the same label $l \in L$ can be associated to more than one transition. In particular, two transitions $t_1, t_2 \in T_o$ are called *undistinguishable* if they share the same label, i.e., $\mathcal{L}(t_1) = \mathcal{L}(t_2)$. The set of transitions sharing the same label l are denoted as T_l.

In the following we denote as C_u (C_o) the restriction of the incidence matrix to T_u (T_o) and denote as n_u and n_o, respectively, the cardinality of the above sets. Moreover, given a sequence $\sigma \in T^*$, $P_u(\sigma)$, resp., $P_o(\sigma)$, denotes the projection of σ over T_u, resp., T_o.

We denote as w the word of events associated to the sequence σ, i.e., $w = P_o(\sigma)$. Note that the length of a sequence σ (denoted $|\sigma|$) is always greater than or equal to the length of the corresponding word w (denoted $|w|$). In fact, if σ contains k' transitions in T_u then $|\sigma| = k' + |w|$.

Definition 3.1 (Cabasino et al., 2009). Let $\langle N,M_0\rangle$ be a labeled net system with labeling function $\mathcal{L} : T \to L \cup \{\varepsilon\}$, where $N = (P,T,Pre,Post)$ and $T = T_o \cup T_u$. Let $w \in L^*$ be an observed word. We define

$$\mathcal{S}(w) = \{\sigma \in L(N,M_0) \mid P_o(\sigma) = w\}$$

the set of firing sequences *consistent* with $w \in L^*$, and

$$\mathcal{C}(w) = \{M \in \mathbb{N}^m \mid \exists \sigma \in T^* : P_o(\sigma) = w \wedge M_0[\sigma\rangle M\}$$

the set of markings *consistent* with $w \in L^*$. ■

In plain words, given an observation w, $\mathcal{S}(w)$ is the set of sequences that may have fired, while $\mathcal{C}(w)$ is the set of markings in which the system may actually be.

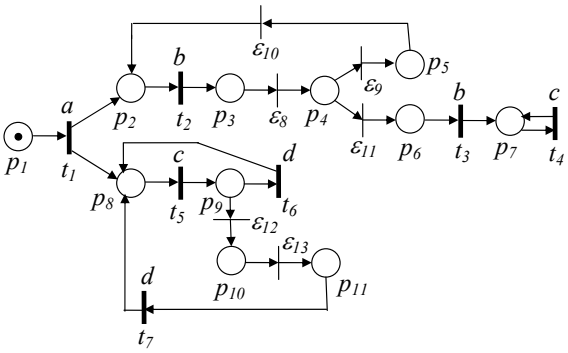

Figure 1: A PN system modeling.

Example 3.2. Let us consider the PN in Figure 1. Let us assume $T_o = \{t_1,t_2,t_3,t_4,t_5,t_6,t_7\}$ and $T_u = \{\varepsilon_8,\varepsilon_9,\varepsilon_{10},\ \varepsilon_{11},\varepsilon_{12},\varepsilon_{13}\}$, where for a better understanding unobservable transitions have been denoted ε_i rather than t_i. The labeling function is defined as follows: $\mathcal{L}(t_1) = a$, $\mathcal{L}(t_2) = \mathcal{L}(t_3) = b$, $\mathcal{L}(t_4) = \mathcal{L}(t_5) = c$, $\mathcal{L}(t_6) = \mathcal{L}(t_7) = d$.

First let us consider $w = ab$. The set of firing sequences that is consistent with w is $\mathcal{S}(w) = \{t_1t_2,\ t_1t_2\varepsilon_8,t_1t_2\varepsilon_8\varepsilon_9,t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10},t_1t_2\varepsilon_8\varepsilon_{11}\}$, and the set of markings consistent with w is $\mathcal{C}(w) = \{[0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T,[0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T,$ $[0\ 0\ 0\ 0\ 1\ 0\ 0\ 1\ 0\ 0\ 0]^T,[0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T,$ $[0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0]^T\}$.

If we consider $w = acd$ the set of firing sequences that are consistent with w is $\mathcal{S}(w) = \{t_1t_5t_6,t_1t_5\varepsilon_{12}\varepsilon_{13}t_7\}$, and the set of markings consistent with w is $\mathcal{C}(w) = \{[0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T\}$. Thus two different firing sequences may have fired (the second one also involving silent transitions), but they both lead to the same marking. ■

4 MINIMAL EXPLANATIONS AND MINIMAL E-VECTORS

In this section we present the notions of minimal explanations and minimal e-vectors for labeled PNs. First we introduce notions of explanations for unlabeled PNs, secondly we define when an explanation is minimal and finally we extend these concepts to labeled PN.

Definition 4.1 (Cabasino et al., 2008). Given a marking M and an observable transition $t \in T_o$, we define

$$\Sigma(M,t) = \{\sigma \in T_u^* \mid M[\sigma\rangle M',\ M' \geq Pre(\cdot,t)\}$$

the set of *explanations* of t at M, and

$$Y(M,t) = \pi(\Sigma(M,t))$$

the *e-vectors* (or *explanation vectors*), i.e., firing vectors associated to the explanations. ■

Thus $\Sigma(M,t)$ is the set of unobservable sequences whose firing at M enables t. Among the above sequences we want to select those whose firing vector is minimal. The firing vector of these sequences are called *minimal e-vectors*.

Definition 4.2 (Cabasino et al., 2008). Given a marking M and a transition $t \in T_o$, we define

$$\Sigma_{\min}(M,t) = \{\sigma \in \Sigma(M,t) \mid \nexists\, \sigma' \in \Sigma(M,t) : \pi(\sigma') \lneqq \pi(\sigma)\}$$

the set of *minimal explanations* of t at M, and we define

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

the corresponding set of *minimal e-vectors*. ■

In (Corona et al., 2004) we proved that, if the unobservable subnet is acyclic and backward conflict-free, then $|Y_{\min}(M,t)| = 1$.

Different approaches can be used to compute $Y_{\min}(M,t)$, e.g., (Boel and Jiroveanu, 2004; Jiroveanu and Boel, 2004). In (Cabasino et al., 2008) we suggested an approach that terminates finding all vectors in $Y_{\min}(M,t)$ if applied to nets whose unobservable subnet is acyclic. It simply requires algebraic manipulations, and is inspired by the procedure proposed in (Martinez and Silva, 1982) for the computation of minimal P-invariants. For the sake of brevity, this algorithm is not reported here.

In the case of labeled PNs what we observe are symbols in L. Thus, it is useful to compute the following sets.

Definition 4.3 (Cabasino et al., 2009). Given a marking M and an observation $l \in L$, we define the set of *minimal explanations of l at M* as

$$\hat{\Sigma}_{\min}(M,l) = \cup_{t \in T_l} \cup_{\sigma \in \Sigma_{\min}(M,t)} (t,\sigma),$$

i.e., the set of pairs (transition labeled l; corresponding minimal explanation), and we define the set of *minimal e-vectors of l at M* as

$$\hat{Y}_{\min}(M,l) = \cup_{t \in T_l} \cup_{e \in Y_{\min}(M,t)} (t,e),$$

i.e., the set of pairs (transition labeled l; corresponding minimal e-vector). ■

Thus, $\hat{\Sigma}_{\min}(M,l)$ is the set of pairs whose first element is the transition labeled l and whose second element is the corresponding minimal explanation $\sigma \in \Sigma_{\min}(M,t)$, namely the corresponding sequence of unobservable transitions whose firing at M enables l and whose firing vector is minimal. Moreover, $\hat{Y}_{\min}(M,l)$ is the set of pairs whose first element is the transition labeled l and whose second element is the firing vector $e \in Y_{\min}(M,t)$ corresponding to the second element in $\hat{\Sigma}_{\min}(M,l)$.

Obviously, $\hat{\Sigma}_{\min}(M,l)$ and $\hat{Y}_{\min}(M,l)$ are a generalization of the sets of minimal explanations and minimal e-vectors introduced for unlabeled PNs with unobservable transitions. Moreover, in the above sets $\hat{\Sigma}_{\min}(M,l)$ and $\hat{Y}_{\min}(M,l)$ different sequences σ and different e-vectors e, respectively, are associated in general to the same $t \in T_l$.

5 BASIS MARKINGS AND J-VECTORS

In this section we introduce the definitions of basis markings and justifications that are the crucial notions of our diagnosis approach.

In particular, given a sequence of observed events $w \in L^*$, a basis marking M_b is a marking reached from M_0 with the firing of the observed word w and of all unobservable transitions whose firing is necessary to enable w. Note that, in general several sequences $\sigma_o \in T_o^*$ may correspond to the same w, i.e., there are several sequences of observable transitions such that $\mathcal{L}(\sigma_o) = w$ that may have actually fired. Moreover, in general, to any of such sequences σ_o a different sequence of unobservable transitions interleaved with it is necessary to make it firable at the initial marking. Thus we need to introduce the following definition of pairs (sequence of transitions in T_o labeled w; corresponding *justification*).

Definition 5.1 (Cabasino et al., 2009). Let $\langle N, M_0 \rangle$ be a net system with labeling function $\mathcal{L} : T \to L \cup \{\varepsilon\}$, where $N = (P,T,Pre,Post)$ and $T = T_o \cup T_u$. Let $w \in L^*$ be a given observation. We define

$$\begin{aligned}\hat{\mathcal{J}}(w) \quad = \{\, & (\sigma_o,\sigma_u),\ \sigma_o \in T_o^*,\ \mathcal{L}(\sigma_o) = w,\ \sigma_u \in T_u^* \mid \\ & [\exists \sigma \in \mathcal{S}(w) : \sigma_o = P_o(\sigma),\ \sigma_u = P_u(\sigma)] \wedge \\ & [\nexists \sigma' \in \mathcal{S}(w) : \sigma_o = P_o(\sigma'),\ \sigma'_u = P_u(\sigma') \wedge \\ & \pi(\sigma'_u) \lneqq \pi(\sigma_u)]\}\end{aligned}$$

the set of pairs (sequence $\sigma_o \in T_o^*$ with $\mathcal{L}(\sigma_o) = w$; corresponding *justification* of w). Moreover, we define

$$\begin{aligned}\hat{Y}_{\min}(M_0,w) = \{(\sigma_o,y),\ & \sigma_o \in T_o^*, \mathcal{L}(\sigma_o) = w, y \in \mathbb{N}^{n_u} \mid \\ & \exists (\sigma_o,\sigma_u) \in \hat{\mathcal{J}}(w) : \pi(\sigma_u) = y\}\end{aligned}$$

the set of pairs (sequence $\sigma_o \in T_o^*$ with $\mathcal{L}(\sigma_o) = w$; corresponding *j-vector*). ■

In simple words, $\hat{\mathcal{J}}(w)$ is the set of pairs whose first element is the sequence $\sigma_o \in T_o^*$ labeled w and whose second element is the corresponding sequence of unobservable transitions interleaved with σ_o whose firing enables σ_o and whose firing vector is minimal.

The firing vectors of these sequences are called *j-vectors*.

Definition 5.2 (Cabasino et al., 2009). Let $\langle N, M_0 \rangle$ be a net system with labeling function $\mathcal{L} : T \to L \cup \{\varepsilon\}$, where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Let w be a given observation and $(\sigma_o, \sigma_u) \in \hat{\mathcal{J}}(w)$ be a generic pair (sequence of observable transitions labeled w; corresponding minimal justification). The marking

$$M_b = M_0 + C_u \cdot y + C_o \cdot y', \qquad y = \pi(\sigma_u),\ y' = \pi(\sigma_o),$$

i.e., the marking reached firing σ_o interleaved with the minimal justification σ_u, is called *basis marking* and y is called its *j-vector* (or *justification-vector*). ■

Obviously, because in general more than one justification exists for a word w (the set $\hat{\mathcal{J}}(w)$ is generally not a singleton), the basis marking may be not unique as well.

Definition 5.3 (Cabasino et al., 2009). Let $\langle N, M_0 \rangle$ be a net system with labeling function $\mathcal{L} : T \to L \cup \{\varepsilon\}$, where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Let $w \in L^*$ be an observed word. We define

$$\mathcal{M}(w) = \{(M, y) \mid (\exists \sigma \in \mathcal{S}(w) : M_0[\sigma\rangle M) \wedge (\exists (\sigma_o, \sigma_u) \in \hat{\mathcal{J}}(w) : \sigma_o = P_o(\sigma), \sigma_u = P_u(\sigma),\ y = \pi(\sigma_u))\}$$

the set of pairs (basis marking; relative j-vector) that are *consistent* with $w \in L^*$. ■

Note that the set $\mathcal{M}(w)$ does not keep into account the sequences of observable transitions that may have actually fired. It only keeps track of the basis markings that can be reached and of the firing vectors relative to sequences of unobservable transitions that have fired to reach them. Indeed, this is the information really significant when performing diagnosis. The notion of $\mathcal{M}(w)$ is fundamental to provide a recursive way to compute the set of minimal explanations.

Proposition 5.4 (Cabasino et al., 2009). Given a net system $\langle N, M_0 \rangle$ with labeling function $\mathcal{L} : T \to L \cup \{\varepsilon\}$, where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Assume that the unobservable subnet is acyclic. Let $w = w'l$ be a given observation.

The set $\hat{Y}_{\min}(M_0, wl)$ is defined as:

$$\hat{Y}_{\min}(M_0, wl) = \{(\sigma_o, y) \mid \sigma_o = \sigma_0' t \wedge y = y' + e : (\sigma_o', y') \in \hat{Y}_{\min}(M_0, w), (t, e) \in \hat{Y}_{\min}(M_b', l) \text{ and } \mathcal{L}(t) = l\},$$

where $M_b' = M_0 + C_u \cdot y' + C_o \cdot \sigma_o'$.

Example 5.5. Let us consider the PN in Figure 1 previously introduced in Example 3.2.

Let us assume $w = acd$. The set of justifications is $\hat{\mathcal{J}}(w) = \{(t_1 t_5 t_6, \varepsilon), (t_1 t_5 t_7, \varepsilon_{12}\varepsilon_{13})\}$ and the set of j-vectors is $\hat{Y}_{min}(M_0, w) = \{(t_1 t_5 t_6, \vec{0}), (t_1 t_5 t_7, [0\ 0\ 0\ 0\ 1\ 1]^T)\}$. The above j-vectors lead to the same basis marking $M_b = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$ thus $\mathcal{M}(w) = \{(M_b, \vec{0}), (M_b, [0\ 0\ 0\ 0\ 1\ 1]^T)\}$.

Now, let us consider $w = ab$. In this case $\hat{\mathcal{J}}(w) = \{(t_1 t_2, \varepsilon)\}$, $\hat{Y}_{min}(M_0, w) = \{(t_1 t_2, \vec{0})\}$ and the basis marking is the same as in the previous case, namely $M_b = [0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$, thus $\mathcal{M}(w) = \{(M_b, \vec{0})\}$. ■

Under the assumption of acyclicity of the unobservable subnet, the set $\mathcal{M}(w)$ can be easily constructed as follows.

Algorithm 5.6 (Computation of the basis markings and j-vectors).

1. Let $w = \varepsilon$.
2. Let $\mathcal{M}(w) = \{(M_0, \vec{0})\}$.
3. Wait until a new label l is observed.
4. Let $w' = w$ and $w = w'l$.
5. Let $\mathcal{M}(w) = \emptyset$.
6. For all M' such that $(M', y') \in \mathcal{M}(w')$, do
  6.1. for all $t \in T_l$, do
    6.1.1. for all $e \in Y_{\min}(M', t)$, do
    6.1.1.1. let $M = M' + C_u \cdot e + C(\cdot, t)$,
    6.1.1.2. for all y' such that $(M', y') \in \mathcal{M}(w')$, do
      6.1.2.1. let $y = y' + e$,
      6.1.2.2. let $\mathcal{M}(w) = \mathcal{M}(w) \cup \{(M, y)\}$.
7. Goto step 3.

■

In simple words, the above algorithm can be explained as follows. We assume that a certain word w (that is equal to the empty string at the initial step) has been observed. Then, a new observable t fires and we observe its label $\mathcal{L}(t)$ (e.g., l). We consider all basis markings at the observation w' before the firing of t, and we select among them those that may have allowed the firing of at least one transition $t \in T_l$, also taking into account that this may have required the firing of appropriate sequences of unobservable transitions. In particular, we focus on the minimal explanations, and thus on the corresponding minimal e-vectors (step 6.1.1). Finally, we update the set $\mathcal{M}(w)$ including all pairs of new basis markings and j-vectors, taking into account that for each basis marking at w' it may correspond more than one j-vector.

Let us now recall the following result.

Definition 5.7 (Cabasino et al., 2008). Let $\langle N, M_0 \rangle$ be a net system where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Assume that the unobservable subnet is acyclic. Let $w \in T_o^*$ be an observed word. We denote

$$\mathcal{M}_{basis}(w) = \{M \in \mathbb{N}^m \mid \exists y \in \mathbb{N}^{n_u} \text{ and } (M, y) \in \mathcal{M}(w)\}$$

the set of basis markings at w. Moreover, we denote as

$$\mathcal{M}_{basis} = \bigcup_{w \in T_o^*} \mathcal{M}_{basis}(w)$$

the set of all basis markings for any observation w. ■

Note that if the net system is bounded then the set $\mathcal{M}_{basis}$ is *finite* being the set of basis markings a subset of the reachability set.

Theorem 5.8 (Cabasino et al., 2008). Let us consider a net system $\langle N, M_0 \rangle$ whose unobservable subnet is acyclic. For any $w \in L^*$ it holds that

$$\mathcal{C}(w) = \{M \in \mathbb{N}^m \mid M = M_b + C_u \cdot y : y \geq \vec{0} \text{ and } M_b \in \mathcal{M}_{basis}(w)\}.$$

6 DIAGNOSIS USING PETRI NETS

Assume that the set of unobservable transitions is partitioned into two subsets, namely $T_u = T_f \cup T_{reg}$ where T_f includes all fault transitions (modeling anomalous or fault behavior), while T_{reg} includes all transitions relative to unobservable but regular events. The set T_f is further partitioned into r different subsets T_f^i, where $i = 1, \ldots, r$, that model the different fault classes.

The following definition introduces the notion of *diagnoser*.

Definition 6.1 (Cabasino et al., 2009). A *diagnoser* is a function $\Delta : L^* \times \{T_f^1, T_f^2, \ldots, T_f^r\} \to \{0,1,2,3\}$ that associates to each observation $w \in L^*$ and to each fault class T_f^i, $i = 1, \ldots, r$, a *diagnosis state*.

- $\Delta(w, T_f^i) = 0$ if for all $\sigma \in \mathcal{S}(w)$ and for all $t_f \in T_f^i$ it holds $t_f \notin \sigma$.
  In such a case the ith fault cannot have occurred, because none of the firing sequences consistent with the observation contains fault transitions of class i.
- $\Delta(w, T_f^i) = 1$ if:
  (i) there exist $\sigma \in \mathcal{S}(w)$ and $t_f \in T_f^i$ such that $t_f \in \sigma$ but
  (ii) for all $(\sigma_o, \sigma_u) \in \hat{\mathcal{J}}(w)$ and for all $t_f \in T_f^i$ it holds that $t_f \notin \sigma_u$.
  In such a case a fault transition of class i may have occurred but is not contained in any justification of w.
- $\Delta(w, T_f^i) = 2$ if there exist $(\sigma_o, \sigma_u), (\sigma'_o, \sigma'_u) \in \hat{\mathcal{J}}(w)$ such that
  (i) there exists $t_f \in T_f^i$ such that $t_f \in \sigma_u$;
  (ii) for all $t_f \in T_f^i$, $t_f \notin \sigma'_u$.
  In such a case a fault transition of class i is contained in one (but not in all) justification of w.
- $\Delta(w, T_f^i) = 3$ if for all $\sigma \in \mathcal{S}(w)$ there exists $t_f \in T_f^i$ such that $t_f \in \sigma$.
  In such a case the ith fault must have occurred, because all firable sequences consistent with the observation contain at least one fault in T_f^i. ■

Example 6.2. Let us consider the PN in Figure 1 previously introduced in Example 3.2. Let $T_f = \{\varepsilon_{11}, \varepsilon_{12}\}$. Assume that the two fault transitions belong to different fault classes, i.e., $T_f^1 = \{\varepsilon_{11}\}$ and $T_f^2 = \{\varepsilon_{12}\}$.

Let us observe $w = a$. Then $\Delta(w, T_f^1) = \Delta(w, T_f^2) = 0$, being $\hat{\mathcal{J}}(w) = \{(t_1, \varepsilon)\}$ and $\mathcal{S}(w) = \{t_1\}$. In words no fault of both fault classes can have occurred.

Let us observe $w = ab$. Then $\Delta(w, T_f^1) = 1$ and $\Delta(w, T_f^2) = 0$, being $\hat{\mathcal{J}}(w) = \{(t_1 t_2, \varepsilon)\}$ and $\mathcal{S}(w) = \{t_1 t_2, t_1 t_2 \varepsilon_8, t_1 t_2 \varepsilon_8 \varepsilon_9, t_1 t_2 \varepsilon_8 \varepsilon_9 \varepsilon_{10}, t_1 t_2 \varepsilon_8 \varepsilon_{11}\}$. This means that a fault of the second fault class may have occurred (e.g. $t_1 t_2 \varepsilon_8 \varepsilon_{11}$) but it is not contained in any justification of ab, while no fault of the first fault class can have occurred.

Now, let us consider $w = abb$. In this case $\Delta(w, T_f^1) = 2$ and $\Delta(w, T_f^2) = 0$, being $\hat{\mathcal{J}}(w) = \{(t_1 t_2 t_2, \varepsilon_8 \varepsilon_9 \varepsilon_{10}), (t_1 t_2 t_3, \varepsilon_8 \varepsilon_{11})\}$ and $\mathcal{S}(w) = \{t_1 t_2 \varepsilon_8 \varepsilon_9 \varepsilon_{10} t_2, t_1 t_2 \varepsilon_8 \varepsilon_9 \varepsilon_{10} t_2 \varepsilon_8, t_1 t_2 \varepsilon_8 \varepsilon_9 \varepsilon_{10} t_2 \varepsilon_8 \varepsilon_9, t_1 t_2 \varepsilon_8 \varepsilon_9 \varepsilon_{10} t_2 \varepsilon_8 \varepsilon_9 \varepsilon_{10}, t_1 t_2 \varepsilon_8 \varepsilon_9 \varepsilon_{10} t_2 \varepsilon_8 \varepsilon_{11}\}$. This means that no fault of the first fault class can have occurred, while a fault of the second fault class may have occurred since one justification does not contain ε_{11} and one justification contains it.

Finally, let us consider $w = abbccc$. In this case $\Delta(w, T_f^1) = 1$ and $\Delta(w, T_f^2) = 3$. In fact since $\hat{\mathcal{J}}(w) = \{(t_1 t_2 t_3 t_5 t_4 t_4, \varepsilon_8 \varepsilon_{11}), (t_1 t_2 t_3 t_4 t_5 t_4, \varepsilon_8 \varepsilon_{11}), (t_1 t_2 t_3 t_4 t_4 t_5, \varepsilon_8 \varepsilon_{11}), (t_1 t_2 t_3 t_4 t_4 t_4, \varepsilon_8 \varepsilon_{11})\}$ a fault of the first fault class must have occurred, while a fault of the second fault class may have occurred (e.g. $t_1 t_2 \varepsilon_8 \varepsilon_{11} t_3 t_4 t_4 t_5 \varepsilon_{12}$) but it is not contained in any justification of w. ■

The following proposition presents how the diagnosis states can be characterized analyzing basis markings and justifications.

Proposition 6.3 (Cabasino et al., 2009). Consider an observed word $w \in L^*$.

- $\Delta(w, T_f^i) \in \{0,1\}$ iff for all $(M, y) \in \mathcal{M}(w)$ and for all $t_f \in T_f^i$ it holds $y(t_f) = 0$.
- $\Delta(w, T_f^i) = 2$ iff there exist $(M, y) \in \mathcal{M}(w)$ and $(M', y') \in \mathcal{M}(w)$ such that:
  (i) there exists $t_f \in T_f^i$ such that $y(t_f) > 0$,
  (ii) for all $t_f \in T_f^i$, $y'(t_f) = 0$.

- $\Delta(w,T_f^i)=3$ iff for all $(M,y)\in\mathcal{M}(w)$ there exists $t_f\in T_f^i$ such that $y(t_f)>0$.

The following proposition shows how to distinguish between diagnosis states 0 and 1.

Proposition 6.4 (Cabasino et al., 2009). For a PN whose unobservable subnet is acyclic, let $w\in L^*$ be an observed word such that for all $(M,y)\in\mathcal{M}(w)$ it holds $y(t_f)=0\ \forall\ t_f\in T_f^i$. Let us consider the constraint set

$$\mathcal{T}(M) \quad = \quad \begin{cases} M+C_u\cdot z\geq\vec{0}, \\ \sum\limits_{t_f\in T_f^i} z(t_f)>0, \\ z\in\mathbb{N}^{n_u}. \end{cases} \tag{1}$$

- $\Delta(w,T_f^i)=0$ if $\forall\ (M,y)\in\mathcal{M}(w)$ the constraint set (1) is not feasible.
- $\Delta(w,T_f^i)=1$ if $\exists\ (M,y)\in\mathcal{M}(w)$ such that the constraint set (1) is feasible.

On the basis of the above two results, if the unobservable subnet is acyclic, diagnosis may be carried out by simply looking at the set $\mathcal{M}(w)$ for any observed word w and, should the diagnosis state be either 0 or 1, by additionally evaluating whether the corresponding integer constraint set (1) admits a solution.

Example 6.5. Let us consider the PN in Figure 1 where $T_f^1=\{\varepsilon_{11}\}$ and $T_f^2=\{\varepsilon_{12}\}$.

Let $w=ab$. In this case $\mathcal{M}(w)=\{(M_b^1,\vec{0})\}$, where $M_b^1=[0\,1\,0\,0\,0\,0\,0\,1\,0\,0\,0]^T$. Being $\mathcal{T}(M_b^1)$ feasible only for the fault class T_f^1 it holds $\Delta(w,T_f^1)=1$ and $\Delta(w,T_f^2)=0$.

Let $w=abb$. It is $\mathcal{M}(w)=\{(M_b^1,[1\ 1\ 1\ 0\ 0\ 0]^T),(M_b^2,[1\ 0\ 0\ 1\ 0\ 0]^T)\}$, where $M_b^2=[0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0]^T$. It is $\Delta(w,T_f^1)=2$ and $\Delta(w,T_f^2)=0$ being both $\mathcal{T}(M_b^1)$ and $\mathcal{T}(M_b^2)$ not feasible.

Let $w=abbccc$. In this case $\mathcal{M}(w)=\{(M_b^3,[1\ 1\ 1\ 0\ 0\ 0]^T),(M_b^4,[1\ 1\ 1\ 0\ 0\ 0]^T)\}$, where $M_b^3=[0\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 0]^T$ and $M_b^4=[0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0\ 0]^T$. It is $\Delta(w,T_f^1)=3$ and being $\mathcal{T}(M_b^4)$ feasible for the second fault class T_f^2 it holds $\Delta(w,T_f^2)=1$. ■

7 BASIS REACHABILITY GRAPH

Diagnosis approach described in the previous section can be applied both to bounded and unbounded PNs. The proposed approach is an on-line approach that for each new observed event updates the diagnosis state for each fault class computing the set of basis markings and j-vectors. Moreover if for a given fault class is necessary to distinguish between diagnosis states 0 and 1, it is also necessary to solve for each basis marking M_b the constraint set $\mathcal{T}(M_b)$.

In this section we show that if the considered net system is bounded, the most burdensome part of the procedure can be moved off-line defining a graph called *Basis Reachability Graph* (BRG).

Definition 7.1. The BRG is a deterministic graph that has as many nodes as the number of possible basis markings.

To each node is associated a different basis marking M and a row vector with as many entries as the number of fault classes. The entries of this vector may only take binary values: 1 if $\mathcal{T}(M)$ is feasible, 0 otherwise.

Arcs are labeled with observable events in L and e-vectors. More precisely, an arc exists from a node containing the basis marking M to a node containing the basis marking M' if and only if there exists a transition t for which an explanation exists at M and the firing of t and one of its minimal explanations leads to M'. The arc going from M to M' is labeled $(\mathcal{L}(t),e)$, where $e\in Y_{\min}(M,t)$ and $M'=M+C_u\cdot e+C(\cdot,t)$. ■

Note that the number of nodes of the BRG is always finite being the set of basis markings a subset of the set of reachable markings, that is finite being the net bounded. Moreover, the row vector of binary values associated to the nodes of the BRG allows us to distinguish between the diagnosis state 1 or 0.

The main steps for the computation of the BRG in the case of labeled PNs are summarized in the following algorithm.

Algorithm 7.2 (Computation of the BRG).

1. Label the initial node (M_0,x_0) where $\forall i=1,\ldots,r$,
$x_0(T_f^i)=\begin{cases}1 & \text{if } \mathcal{T}(M_0) \text{ is feasible,}\\ 0 & \text{otherwise.}\end{cases}$
Assign no tag to it.
2. While nodes with no tag exist
select a node with no tag and do
2.1. let M be the marking in the node (M,x),
2.2. for all $l\in L$
2.2.1. for all $t: L(t)=l\wedge Y_{\min}(M,t)\neq\emptyset$, do
- for all $e\in Y_{\min}(M,t)$, do
- let $M'=M+C_u\cdot e+C(\cdot,t)$,
- if $\nexists$ a node (M,x) with $M=M'$, do
- add a new node to the graph containing (M',x') where $\forall i=1,\ldots,r$,
$x'(T_f^i)=\begin{cases}1 & \text{if } \mathcal{T}(M') \text{ is feasible,}\\ 0 & \text{otherwise.}\end{cases}$
and arc (l,e) from (M,x) to (M',x')
- else
- add arc (l,e) from (M,x) to (M',x')

if it does not exist yet
2.3. tag the node "old".
3. Remove all tags.

■

The algorithm constructs the BRG starting from the initial node to which it corresponds the initial marking and a binary vector defining which classes of faults may occur at M_0. Now, we consider all the labels $l \in L$ such that there exists a transition t with $L(t) = l$ for which a minimal explanation at M_0 exists. For any of these transitions we compute the marking resulting from firing t at $M_0 + C_u \cdot e$, for any $e \in Y_{\min}(M_0, t)$. If a pair (marking, binary vector) not contained in the previous nodes is obtained, a new node is added to the graph. The arc going from the initial node to the new node is labeled (l, e). The procedure is iterated until all basis markings have been considered. Note that, our approach always requires to enumerate a state space that is a strict subset of the reachability space. However, as in general for diagnosis approaches, the combinatory explosion cannot be avoided.

Example 7.3. Let us consider the PN in Figure 1, where $T_o = \{t_1, t_2, t_3, t_4, t_5, t_6, t_7\}$, $T_u = \{\varepsilon_8, \varepsilon_9, \varepsilon_{10}, \varepsilon_{11}, \varepsilon_{12}, \varepsilon_{13}\}$, $T_f^1 = \{\varepsilon_{11}\}$ and $T_f^2 = \{\varepsilon_{12}\}$. The labeling function is defined as follows: $\mathcal{L}(t_1) = a$, $\mathcal{L}(t_2) = \mathcal{L}(t_3) = b$, $\mathcal{L}(t_4) = \mathcal{L}(t_5) = c$, $\mathcal{L}(t_6) = \mathcal{L}(t_7) = d$.

The BRG is shown in Figure 2. The notation used in in this figure is detailed in Tables 1 and 2. Each node contains a different basis marking and a binary row vector of dimension two, being two the number of fault classes. As an example, the binary vector $[0\ 0]$ is associated to M_0 because $\mathcal{T}(M_0)$ is not feasible for both fault classes. From node M_0 to node M_1 there is one arc labeled a and with the null vector as minimal explanation. The node containing the basis marking M_2 has binary vector $[0\ 1]$, because $\mathcal{T}(M_2)$ is feasible only for T_f^2. Node $(M_2, [0\ 1])$ has two output arcs both labeled with d and both directed to node $(M_1, [0\ 0])$ with two different minimal explanations $\vec{0}$ and e_1, respectively, plus another output arc $(b, \vec{0})$ directed to node $(M_4, [1\ 1])$. ■

The following algorithm summarizes the main steps of the on-line diagnosis carried out by looking at the BRG.

Algorithm 7.4 (Diagnosis using the BRG).

1. Let $w = \varepsilon$.
2. Let $\mathcal{M}(w) = \{(M_0, \vec{0})\}$.
3. Wait until a new observable transition fires.
Let l be the observed event.
4. Let $w' = w$ and $w = w'l$.
5. Let $\mathcal{M}(w) = \emptyset$, **[Computation of $\mathcal{M}(w)$]**
6. For all nodes containing M' : $(M', y') \in \mathcal{M}(w')$, do

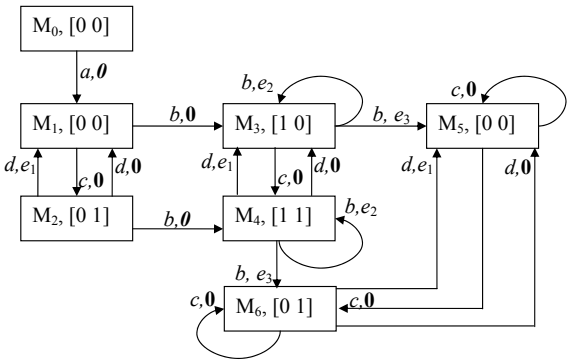

Figure 2: The BRG of the PN in Figure 1.

6.1. for all arcs exiting from the node with M', do
6.1.1. let M be the marking of the output node and e be the minimal e-vector on the edge from M' to M,
6.1.2. for all y' such that $(M', y') \in \mathcal{M}(w')$, do
6.1.2.1. let $y = y' + e$,
6.1.2.2. let $\mathcal{M}(w) = \mathcal{M}(w) \cup \{(M, y)\}$,
7. for all $i = 1, \ldots, r$, do
[Computation of the diagnosis state]
7.1. if $\forall\, (M, y) \in \mathcal{M}(w) \wedge \forall t_f \in T_f^i$ it is $y(t_f) = 0$, do
7.1.1. if $\forall\, (M, y) \in \mathcal{M}(w)$ it holds $x(i) = 0$, where x is the binary vector in node M, do
7.1.1.1. let $\Delta(w, T_f^i) = 0$,
7.1.2. else
7.1.2.1. let $\Delta(w, T_f^i) = 1$,
7.2. if $\exists\, (M, y) \in \mathcal{M}(w)$ and $(M', y') \in \mathcal{M}(w)$ s.t.:
(i) $\exists t_f \in T_f^i$ such that $y(t_f) > 0$,
(ii) $\forall t_f \in T_f^i$, $y'(t_f) = 0$, do
7.2.1. let $\Delta(w, T_f^i) = 2$,
7.3. if $\forall\, (M, y) \in \mathcal{M}(w)\ \exists t_f \in T_f^i : y(t_f) > 0$, do
7.3.1. let $\Delta(w, T_f^i) = 3$.
8. Goto step 3.

■

Steps 1 to 6 of Algorithm 7.4 enables us to compute the set $\mathcal{M}(w)$. When no event is observed, namely $w = \varepsilon$, then $\mathcal{M}(w) = \{(M_0, \vec{0})\}$. Now, assume that a label l is observed. We include in the set $\mathcal{M}(l)$ all couples (M, y) such that an arc labeled l exits from the initial node and ends in a node containing the basis marking M. The corresponding value of y is equal to the e-vector in the arc going from M_0 to M, being $\vec{0}$ the j-vector relative to M_0. In general, if w' is the actual observation, and a new event labeled l fires, we consider all couples $(M', y') \in \mathcal{M}(w')$ and all nodes that can be reached from M' with an arc labeled l. Let M be the basis marking of the generic resulting node. We include in $\mathcal{M}(w) = \mathcal{M}(w't)$ all couples (M, y), where for any M, y is equal to the sum of y' plus the e-vector labeling the arc from M' to M.

Step 7 of Algorithm 7.4 computes the diagnosis

Table 1: The markings of the BRG in Figure 2.

M_0	[1	0	0	0	0	0	0	0	0	0	]^T
M_1	[0	1	0	0	0	0	0	1	0	0	]^T
M_2	[0	1	0	0	0	0	0	0	1	0	]^T
M_3	[0	0	1	0	0	0	0	1	0	0	]^T
M_4	[0	0	1	0	0	0	0	0	1	0	]^T
M_5	[0	0	0	0	0	0	1	1	0	0	]^T
M_6	[0	0	0	0	0	0	1	0	1	0	]^T

Table 2: The e-vectors of the BRG in Figure 2.

	ε_8	ε_9	ε_{10}	ε_{11}	ε_{12}	ε_{13}
e_1	0	0	0	0	1	1
e_2	1	1	1	0	0	0
e_3	1	0	0	1	0	0

state. Let us consider the generic ith fault class. If $\forall(M,y) \in \mathcal{M}(w)$ and $\forall t_f \in T_f^i$ it holds $y(t_f) = 0$, we have to check the ith entry of all the binary row vectors associated to the basis markings M, such that $(M,y) \in \mathcal{M}(w)$. If these entries are all equal to 0, we set $\Delta(w,T_f^i) = 0$, otherwise we set $\Delta(w,T_f^i) = 1$. On the other hand, if there exists at least one pair $(M,y) \in \mathcal{M}(w)$ with $y(t_f) > 0$ for any $t_f \in T_f^i$, and there exists at least one pair $(M',y') \in \mathcal{M}(w)$ with $y(t_f) = 0$ for all $t_f \in T_f^i$, then $\Delta(w,T_f^i) = 2$. Finally, if for all pairs $(M,y) \in \mathcal{M}(w)\ y(t_f) > 0$ for any $t_f \in T_f^i$, then $\Delta(w,T_f^i) = 3$.

The following example shows how to perform diagnosis on-line simply looking at the BRG.

Example 7.5. Let us consider the PN in Figure 1 and its BRG in Figure 2. Let $w = \varepsilon$. By looking at the BRG we establish that $\Delta(\varepsilon,T_f^1) = \Delta(\varepsilon,T_f^2) = 0$ being both entries of the row vector associated to M_0 equal to 0.

Now, let us consider $w = ab$. In such a case $\mathcal{M}(w) = \{(M_3,\vec{0})\}$. It holds $\Delta(ab,T_f^1) = 1$ and $\Delta(ab,T_f^2) = 0$ being the row vector in the node equal to [1 0].

Finally, for $w = abbc$ it holds $\Delta(abbc,T_f^1) = 2$ and $\Delta(abbc,T_f^2) = 1$. In fact $\mathcal{M}(w) = \{(M_4,y_1),(M_5,y_2)\}$, where $y_1 = e_2$, $y_2 = e_2 + e_3$, and the row vectors associated to M_4 and M_5 are respectively [1 1] and [0 0]. ■

8 MATLAB TOOLBOX

Our group at the University of Cagliari has developed a MATLAB toolbox for PNs.

In this section we illustrate how it can be used for the diagnosis of labeled PNs. In particular, we consider the function that given a bounded labeled PN builds the basis reachability graph.

The input of the MATLAB function BRG.m are:

- the structure of the net, i.e., the matrices *Pre* and *Post*;
- the initial marking M_0;
- a cell array F that has as many rows as the number of fault classes, that contains in each row the fault transitions that belong to the corresponding fault class;
- a cell array L that has as many rows as the cardinality of the considered alphabet, that contains in each row the observable transitions having the same label;
- a cell array E that contains in each row a string of characters, each one corresponding to a different label in the considered alphabet. Obviously, the cell array E is ordered according to L.

The output of the MATLAB function BRG.m is a cell array T that univocally identifies the resulting BRG. It has as many rows as the number of nodes of the BRG. A different row is associated to each node and contains the following information:

- an identifier number of the node;
- a matrix whose rows are equal to the transpose of the basis markings associated to the node;
- a matrix with as many rows as the number of basis markings associated to the node and as many columns as the number of fault classes: the jth element in the ith row (corresponding to M_b^i) is equal to $x_i(T_f^j)$ evaluated at M_b^i. Thus, $x_i(T_f^j) = 0$ is $\mathcal{T}(M_b^i)$ is not feasible with respect to T_j^f, 1 otherwise;

- the transitions enabled at node;
- the identifier number of the nodes that are reached firing an enabled transition and the corresponding j-vector.

9 NUMERICAL SIMULATIONS

Let us consider the Petri net in Figure 3 (Lai et al., 2008), where thick transitions represent observable event and thin transitions represent unobservable events. It models a family of manufacturing systems characterized by three parameters: n, m and k.

— n is the number of production lines.

— m is the number of units of the final product that can be simultaneously produced. Each unit of product is composed of n parts.

— k is the number of operations that each part must undergo in each line.

To obtain one unit of final product n orders are sent, one to each line; this is represented by observable event t_s. Each line will produce a part (all parts are identical) and put it in its final buffer. An assembly station will take one part from each buffer (observable event t_e) to produce the final product.

The part in line i ($i = 1,\ldots,n$) undergoes a series of k operations, represented by unobservable events $\varepsilon_{i,1},\varepsilon_{i,2},\cdots,\varepsilon_{i,k}$.

After this series of operations two events are possible: either the part is regularly put in the final buffer of the line, or a fault may occur.

— Putting the part in the final buffer of line 1 corresponds to unobservable event $\varepsilon_{1,k+1}$, while putting the part in the final buffer of line i ($i = 2,\ldots,n$) corresponds to observable event $t_{i,k+1}$.

— There are $n-1$ faults, represented by unobservable events f_i ($i = 1,\ldots,n-1$). Fault f_i moves a part from line i to line $i+1$. Note that on line i ($i = 1,\ldots,n-1$) the fault may only occur when the part has finished processing and is ready to be put in its final buffer; the part goes to the same processing stage in line $i+1$.

In this section we present the results of the computation of the BRG for several numerical simulations. Results obtained for different values of n, k and m are summarized in Tables 3, 4 and 5.

Note that for the sake of simplicity we assumed that all faults belong to the same class.

In these tables we also detail the cardinality of the reachability set R. This is an extremely important parameter to appreciate the advantage of using basis markings. The value of $|R|$ has been computed using a function we developed in MATLAB. For completeness we also reported the time necessary to compute it.

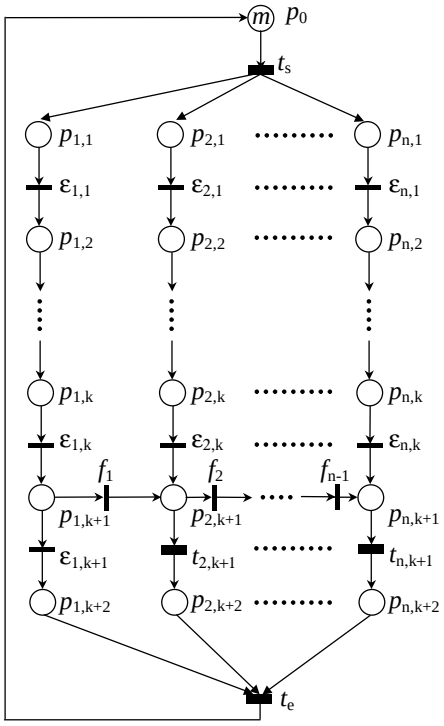

Figure 3: A manufacturing system.

Let us observe that some boxes of the above tables contain the non numerical values o:t: (out of time), that denotes that the corresponding value has not been computed within 6 hours.

All simulations have been run on a PC Athlon 64, 4000+ processor.

— Columns 1 and 2 show the values of n and k.

— Column 3 shows the number of nodes $|R|$ of the reachability graph.

— Column 4 shows the time t_R in seconds we spent to compute the reachability graph.

— Column 4 shows the number of nodes $|BRG|$ of the BRG.

— Column 5 shows the time t_{BRG} in seconds we spent to compute the BRG using the function BRG.m.

Tables 3, 4 and 5 show that the time spent to compute the reachability graph highly increases with the dimension of the net, namely with n and k, and with the number of products m.

On the contrary, the time spent to compute the BRG is always reasonable even for high values of n, k and m.

Tables 3, 4 and 5 also show that the number of nodes of the BRG only depends on n and m, while it is invariant with respect to k. On the other hand, $|R|$ also highly increases with k.

Table 3: Numerical results in the case of $m = 1$.

n	k	$\|R\|$	t_R [sec]	$\|BRG\|$	t_{BRG} [sec]
2	1	15	0.031	5	0.062
2	2	24	0.031	5	0.062
2	3	35	0.047	5	0.062
2	4	48	0.062	5	0.07
2	5	63	0.078	5	0.07
2	6	80	0.094	5	0.07
3	1	80	0.094	17	0.101
3	2	159	0.25	17	0.101
3	3	274	0.672	17	0.109
3	4	431	1.72	17	0.117
3	5	636	3.938	17	0.125
3	6	895	8.328	17	0.132
4	1	495	2.375	69	0.375
4	2	1200	16.969	69	0.43
4	3	2415	77.828	69	0.477
4	4	4320	272.53	69	0.531
4	5	7119	824.69	69	0.594
4	6	11040	2122.4	69	0.664
5	1	3295	155.81	305	4.345
5	2	9691	1615.7	305	4.765
5	3	22707	10288	305	5.25
5	4	*o.t.*	*o.t.*	305	5.75
5	5	*o.t.*	*o.t.*	305	6.897
5	6	*o.t.*	*o.t.*	305	7.894

Table 4: Numerical results in the case of $m = 2$.

n	k	$\|R\|$	t_R [sec]	$\|BRG\|$	t_{BRG} [sec]
2	1	96	0.11	17	0.086
2	2	237	0.469	17	0.094
2	3	496	2.078	17	0.1
3	1	1484	24.204	140	0.78
3	2	5949	486.39	140	0.844
3	3	18311	5320.9	140	0.906
4	1	28203	14006	1433	73.5
4	2	*o.t.*	*o.t.*	1433	76.5
4	3	*o.t.*	*o.t.*	1433	76.5

For the considered Petri net, on the basis of the above simulations, we can conclude that the diagnosis approach here presented is suitable from a computational point of view. In fact, thanks to the basis markings the reachability space can be described in a compact manner.

Table 5: Numerical results in the case of $m = 3$.

n	k	$\|R\|$	t_R [sec]	$\|BRG\|$	t_{BRG} [sec]
2	1	377	1.203	39	0.145
2	2	1293	17.203	39	0.145
3	1	12048	2113.9	553	8.219
3	2	*o.t.*	*o.t.*	553	9.016
4	1	*o.t.*	*o.t.*	9835	4095.06
4	2	*o.t.*	*o.t.*	9835	4095.06

10 CONCLUSIONS AND FUTURE WORK

This paper presents a diagnosis approach for labeled PNs using basis markings. This enables us to avoid an exhaustive enumeration of the reachability set. This approach applies to all bounded and unbounded Petri net systems whose unobservable subnet is acyclic. However, if we consider bounded net systems the most burdensome part of the procedure may be moved off-line computing the Basis Reachability Graph. Finally, we have presented a tool for the diagnosis of labeled bounded PNs and we have shown the simulation results using as diagnosis benchmark a family of manufacturing systems.

We have also studied the problem of diagnosability of bounded and unbounded PNs giving for both cases necessary and sufficient conditions for diagnosability. These results are not reported here, but they have been already submitted to an international conference.

Our future work will be that of studying the diagnosis problem for distributed systems investigating the possibility of extending the approach here presented to this case.

ACKNOWLEDGEMENTS

We thank Marco Pocci, a Master student of Electronic Engineering at the University of Cagliari, for his help in the development of the MATLAB tool for the construction of the BRG for labeled PNs.

REFERENCES

Basile, F., Chiacchio, P., and Tommasi, G. D. (2008). An efficient approach for online diagnosis of discrete event systems. *IEEE Trans. on Automatic Control*. in press.

Benveniste, A., Fabre, E., Haar, S., and Jard, C. (2003). Diagnosis of asynchronous discrete event systems: A

net unfolding approach. *IEEE Trans. on Automatic Control*, 48(5):714–727.

Boel, R. and Jiroveanu, G. (2004). Distributed contextual diagnosis for very large systems. In *Proc. IFAC WODES'04: 7th Work. on Discrete Event Systems*, pages 343–348.

Boel, R. and van Schuppen, J. (2002). Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers. In *Proc. WODES'02: 6th Work. on Discrete Event Systems*, pages 175–181.

Cabasino, M., Giua, A., and Seatzu, C. (2008). Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*. Preliminary accepted.

Cabasino, M., Giua, A., and Seatzu, C. (2009). Diagnosis of discrete event systems using labeled Petri nets. In *Proc. 2nd IFAC Workshop on Dependable Control of Discrete Systems (Bari, Italy)*.

Chung, S. (2005). Diagnosing pn-based models with partial observable transitions. *International Journal of Computer Integrated Manufacturing*, 12 (2):158–169.

Corona, D., Giua, A., and Seatzu, C. (2004). Marking estimation of Petri nets with silent transitions. In *Proc. IEEE 43rd Int. Conf. on Decision and Control (Atlantis, The Bahamas)*.

Debouk, R., Lafortune, S., and Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete-event systems. *Discrete Events Dynamical Systems*, 10(1):33–86.

Dotoli, M., Fanti, M., and Mangini, A. (2008). Fault detection of discrete event systems using Petri nets and integer linear programming. In *Proc. of 17th IFAC World Congress*, Seoul, Korea.

Genc, S. and Lafortune, S. (2007). Distributed diagnosis of place-bordered Petri nets. *IEEE Trans. on Automation Science and Engineering*, 4(2):206–219.

Ghazel, M., Togueni, A., and Bigang, M. (2005). A monitoring approach for discrete events systems based on a time Petri net model. In *Proc. of 16th IFAC World Congress*, Prague, Czech Republic.

Giua, A. and Seatzu, C. (2005). Fault detection for discrete event systems using Petri nets with unobservable transitions. In *Proc. 44th IEEE Conf. on Decision and Control*, pages 6323–6328.

Hadjicostis, C. and Veghese, G. (1999). Monitoring discrete event systems using Petri net embeddings. *Lecture Notes in Computer Science*, 1639:188–207.

Jiroveanu, G. and Boel, R. (2004). Contextual analysis of Petri nets for distributed applications. In *16th Int. Symp. on Mathematical Theory of Networks and Systems (Leuven, Belgium)*.

Lai, S., Nessi, D., Cabasino, M., Giua, A., and Seatzu, C. (2008). A comparison between two diagnostic tools based on automata and Petri nets. In *Proc. IFAC WODES'08: 9th Work. on Discrete Event Systems*, pages 144–149.

Lefebvre, D. and Delherm, C. (2007). Diagnosis of DES with Petri net models. *IEEE Trans. on Automation Science and Engineering*, 4(1):114–118.

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

Lin, F., Markee, J., , and Rado, B. (1993). Design and test of mixed signal circuits: a discrete event approach. In *Proc. 32rd IEEE Conf. on Decision and Control*, pages 246–251.

Martinez, J. and Silva, M. (1982). A simple and fast algorithm to obtain all invariants of a generalized Petri net. In *Informatik-Fachberichte 52: Application and Theory of Petri Nets.*, pages 301–310. Springer-Verlag.

Murata, T. (1989). Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580.

Prock, J. (1991). A new tecnique for fault detection using Petri nets. *Automatica*, 27(2):239–245.

Ruiz-Beltràn, A. R.-T. E., Rivera-Rangel, I., and Lopez-Mellado, E. (2007). Online fault diagnosis of discrete event systems. A Petri net-based approach. *IEEE Trans. on Automation Science and Engineering*, 4(1):31–39.

Sampath, M., Lafortune, S., and Teneketzis, D. (1998). Active diagnosis of discrete-event systems. *IEEE Trans. on Automatic Control*, 43(7):908–929.

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

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1996). Failure diagnosis using discrete-event models. *IEEE Trans. Control Systems Technology*, 4(2):105–124.

Sreenivas, V. and Jafari, M. (1993). Fault detection and monitoring using time Petri nets. *IEEE Trans. Systems, Man and Cybernetics*, 23(4):1155–1162.

Ushio, T., Onishi, L., and Okuda, K. (1998). Fault detection based on Petri net models with faulty behaviors. In *Proc. SMC'98: IEEE Int. Conf. on Systems, Man, and Cybernetics (San Diego, CA, USA)*, pages 113–118.

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

Wen, Y., Li, C., and Jeng, M. (2005). A polynomial algorithm for checking diagnosability of Petri nets. In *Proc. SMC'05: IEEE Int. Conf. on Systems, Man, and Cybernetics*, pages 2542– 2547.

Wu, Y. and Hadjicostis, C. (2005). Algebraic approaches for fault identification in discrete-event systems. *IEEE Trans. Robotics and Automation*, 50(12):2048–2053.

Zad, S. H., Kwong, R., and Wonham, W. (2003). Fault diagnosis in discrete-event systems: framework and model reduction. *IEEE Trans. on Automatic Control*, 48(7):1199–1212.

BRIEF BIOGRAPHY

Alessandro Giua is professor of Automatic Control at the Department of Electrical and Electronic Engineering of the University of Cagliari, Italy. He received the Laurea degree in electric engineering from the University of Cagliari, Italy in 1988, and the M.S. and Ph.D. degrees in computer and systems engineering from Rensselaer Polytechnic Institute, Troy, New York, in 1990 and 1992.

His research interests include discrete event systems, hybrid systems, networked control systems, automated manufacturing, Petri nets, control of mechanical systems, failure diagnosis. He has co-authored two textbooks on Automatic Control (in Italian) and over 150 technical papers.

Dr. Giua is a member of the editorial board of the journals: Discrete Event Dynamic Systems: Theory and Applications; IEEE Trans. on Control Systems Technology; Nonlinear Analysis: Hybrid Systems. He has served in the program committee of over 60 international conferences.

He is chair for Chapter Activities of the Member Activities Board of the IEEE Control Systems Society and chair of the IFAC Technical Committee 1.3 on Discrete Event and Hybrid Systems.