Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems

Maria Paola Cabasino, Alessandro Giua, Marco Pocci, Carla Seatzu
Dip. di Ing. Elettrica ed Elettronica, Università di Cagliari, Italy
Email: {cabasino,giua,marco.pocci,seatzu}@diee.unica.it

Abstract

In this paper an approach to on-line diagnosis of discrete event systems based on labeled Petri nets is presented. The 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 called *Basis Reachability Graph*.

Finally, the effectiveness of the proposed procedure is analyzed applying a MATLAB diagnosis toolbox we developed to a manufacturing example taken from the literature.

Published as: M.P. Cabasino, A. Giua, C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," *Automatica*, Vol. 46, No. 9, pp. 1531-1539, September 2010.

This work has been partially supported by the European Communitys Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498).

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 systems of increasing size and functionality are built, and as increasing demands on the performance of these systems are placed, then the complexity of these systems increases. Consequently (and unfortunately), the potential for systems to fail is enhanced, and no matter how safe the designs are, how improved the quality control techniques are, and how better trained the operators are, system failures become unavoidable (Sampath, 1995).

Given the fact that failures are inevitable, the need for effective means of detecting them is quite apparent if their consequences and impacts are considered not just on the systems involved but on the society as a whole. Moreover, note that effective methods of failure diagnosis can not only help avoiding 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, one can 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.

The diagnosis of discrete event systems (DES) is a research area that has received a lot of attention in the last years. Faults may correspond to any discrete event. As an example, in a telecommunication system, a fault may correspond to a message that is lost or not sent to the appropriate receiver. Similarly, in a transportation system, a fault may be a traffic light that does not switch from red to green according to the given schedule. In a manufacturing system (Viswanadham and Johnson, 1988; Baviehi and Chong, 1994; Lunze and Schroder, 2004; Garcia *et al.*, 2005), it may be the failure of a certain operation, e.g., a wrong assembly, or a part put in a wrong buffer, and so on.

A categorization of faults arises from the manner in which faults are reset after they occur. It can be distinguished between permanent and intermittent faults. A fault is *permanent* if the recovery event occurs only due to a repair/replacement of the fault that is controllable and observable. On the contrary, a fault is *intermittent* if the recovery event can occur either spontaneously or through repair/replacement; it tends to be uncontrollable and unobservable. Example is a loose

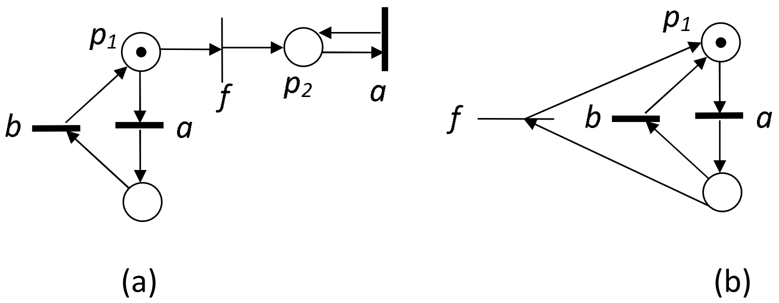

Figure 1: An example of net system with a permanent fault (a) and with an intermittent fault (b).

wire that makes and breaks contact spontaneously. It is important to distinguish between these two types of faults. In fact, intermittent faults may spontaneously recover, making the system oscillate between non-faulty and fault states. On the other hand, in the case of permanent faults, the system cannot spontaneously move from a fault state to a non-fault one (Huang *et al.*, 2003). Examples of permanent and intermittent faults in a given net system are presented in Fig. 1.(a) and Fig. 1.(b) respectively. In particular, the net system during the nominal behavior produces a cyclic sequence of “a” followed by “b”. In Fig. 1.(a) a permanent fault to the sensor that produces “b” is modeled: after the occurrence of the fault f only events a will be produced. On the contrary, Fig. 1.(b) represents an intermittent fault to the sensor “b”: after the occurrence of the fault event f the sensor that produces b may start working again.

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

Solving a problem of diagnosis means that to each observed string of events is associated 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. This paper is focused on the problem of diagnosis; see Cabasino *et al.* (2009*a*) and Cabasino *et al.* (2009*c*) for an extension of the methodology here proposed to the diagnosability problem. However, it is well known that diagnosability is an essential property that must hold if a diagnosis approach is to be applied in real life applications. Thus, the manufacturing example considered in this paper is diagnosable. In particular, it is a parametric example that is diagnosable for any values of the input parameters. This property has been tested using the MATLAB tool in Pocci (2009). Note that if a system contains a non diagnosable fault there exist sequences of unbounded length that lead the system through diagnosis states that are uncertain. This means that when the fault occurs the diagnoser may not be able to detect its firing.

As discussed in the next session the first results on diagnosis of DES 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 PNs 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.

This paper is focused on arbitrary labeled PNs where there is an association between sensors and observable events, while no sensor is available for certain activities — such as faults or other unobservable but regular transitions — due to budget constraints or technology limitations. It is assumed that several different transitions might share the same sensor in order to reduce cost or power consumption. If two transitions are simultaneously enabled and one of them fires it is impossible to distinguish which one has fired, thus they are called *undistinguishable*. The diagnosis approach here presented 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 this procedure. First, it is not necessary 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 the most burdensome part of the procedure, namely building a finite graph called *basis reachability graph* (BRG), can be moved off-line.

Note that the approach here presented, as most of the approaches dealing with diagnosis of discrete event systems (Debouk *et al.*, 2000; Sampath *et al.*, 1995, 1998; Zad *et al.*, 2003*a*), assumes that the faulty behavior is completely known, thus a fault model is available. Such an assumption is applicable to interesting classes of problems: this is the case of many manufacturing systems where the set of possible faults is often predictable and finite in number (Garcia *et al.*, 2005; Baviehi and Chong, 1994; Lunze and Schroder, 2004; Viswanadham and Johnson, 1988). Moreover, the proposed diagnosis approach allows one to deal with both permanent and intermittent faults. However, in the case of intermittent faults, once a fault is detected, even if a recovery event occurs, the diagnosis state associated to the fault is not reset to a non faulty state. The procedure can be easily extended to overcome this limitation. In particular, if the recovery event is observable a simple reset rule on the diagnosis state should be introduced. On the contrary, if the recovery event is not observable a detection procedure on such an event, based on the same features of the fault detection procedure here presented, should be applied.

The paper is organized as follows. In Section 2 the state of art of diagnosis for discrete event systems is illustrated. In Section 3 a background on PNs is provided. In Section 4 are introduced the definitions of minimal explanations, justifications and basis markings, that are the basic notions of the diagnosis approach presented in Section 5. In Section 5 the diagnosis states are defined and a characterization of them in terms of basis markings and j-vectors is given. In Section 6 it is shown how the most burdensome part of the procedure can be moved off-line in the case of bounded PNs. In Section 7 a MATLAB toolbox for PNs diagnosis is presented and in Section 8 some numerical results obtained applying our tool to a manufacturing model taken from the literature are presented. In Section 9 conclusions are drawn.

2 LITERATURE REVIEW

In this section the state of the art of diagnosis of DES using automata and PNs is presented.

2.1 Diagnosis of DES using Automata

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

Lin (1994); Lin *et al.* (1993) propose a state-based DES approach to failure diagnosis. 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.

Sampath *et al.* (1995, 1996) 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. Sampath *et al.* (1995) provide a definition of diagnosability in the framework of formal languages and present necessary and sufficient conditions for diagnosability of systems. Moreover a systematic approach to solve the problem of diagnosis using diagnosers is introduced.

In a related work Sampath *et al.* (1998) 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.

Debouk *et al.* (2000) deal with the problem of failure diagnosis in DES with decentralized information. In particular, they 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.

Boel and van Schuppen (2002) 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.

Zad *et al.* (2003*b*) present a state-based approach for on-line passive fault diagnosis. 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, let us recall the approach of Prock (1991). He 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.

Sreenivas and Jafari (1993) 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 by Ghazel *et al.* (2005) that 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.

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

Wu and Hadjicostis (2005) use redundancy into a given PN to enable fault detection and identification using algebraic decoding techniques. In this paper the authors 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, Lefebvre and Delherm (2007) investigate on the determination of the set of places that must be observed for the exact and immediate estimation of faults occurrence.

Miyagi and Riascos (2010) introduce a methodology, based on the hierarchical and modular integration of PNs, for modeling and analyzing fault-tolerant manufacturing systems that not only optimizes normal productive processes, but also performs detection and treatment of faults.

Ramirez-Treviño (2007) employ Interpreted PNs 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.*, 2009; Benveniste *et al.*, 2003; Dotoli *et al.*, 2008; Genc and Lafortune, 2007).

In particular, Genc and Lafortune (2007) 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.* (2009); Dotoli *et al.* (2008).

In particular, Benveniste *et al.* (2003) use a net unfolding approach for designing an on-line asynchronous diagnoser. 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.

Basile *et al.* (2009) build the diagnoser 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.

Dotoli *et al.* (2008) propose a diagnoser that works on-line in order to avoid the redesign and the redefinition of the diagnoser when the structure of the system changes. In particular, the diagnoser 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. A problem strictly related to diagnosis has been recently studied by Dotoli *et al.* (2010). They address the problem of identifying the model of the unobservable behavior of PN systems in the industrial automation framework. Assuming that the fault-free system structure and dynamics are known, the paper proposes an algorithm that monitors the system on-line, storing the occurred observable event sequence and the corresponding reached states.

A series of contributions dealing with diagnosis of PNs (Cabasino *et al.*, 2010; Lai *et al.*, 2008; Cabasino *et al.*, 2009*b*) have also been proposed by the authors of this paper. In particular, in Cabasino *et al.* (2010); Lai *et al.* (2008) *free-labeled* PNs are considered, while in Cabasino *et al.* (2009*b*), as well as in this paper, the focus is on *labeled* PNs.

Some authors of this paper have also addressed the problem of diagnosability, namely the problem of providing a procedure to verify if it is possible to reconstruct the occurrence of fault events observing words of finite length. In particular, two different approaches for bounded (Cabasino *et al.*, 2009*a*) and unbounded (Cabasino *et al.*, 2009*c*) PNs have been proposed.

Very few other results deal with diagnosability within the framework of PNs.

The first contribution was given by Ushio *et al.* (1998) that extend a necessary and sufficient condition for diagnosability given in Sampath *et al.* (1995, 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.

Chung (2005), 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.

Finally, Wen and Jeng (2005) propose an approach to test diagnosability by checking the structure property of T-invariants 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. Moreover Wen *et al.* (2005) 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 PNs.

3 BACKGROUND

In this section the formalism used in the paper is recalled. For more details on PNs the reader is referred 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 nonnegative integer number of tokens, represented by black dots. The marking of place p is denoted as $M(p)$. 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 $M' = M + C(\cdot, t)$. One writes $M\ [\sigma\rangle$ to denote that the sequence of transitions $\sigma = t_{j_1} \cdots t_{j_k}$ is enabled at M, and $M\ [\sigma\rangle\ M'$ to denote that the firing of σ yields M'. One writes $t \in \sigma$ 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 $\sigma \in T^*$, let $\pi : T^* \rightarrow \mathbb{N}^n$ be the function that associates to σ a vector $y \in \mathbb{N}^n$, called 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 $M \in R(N, M_0)$, $M(p) \leq k$.

The association between sensors and transitions can be captured by a *labeling function* $\mathcal{L} : T \rightarrow L \cup \{\varepsilon\}$ assigns to each transition $t \in T$ either a symbol from a given alphabet L or the empty string ε.

The set of transitions whose label is ε is denoted as T_u, i.e., $T_u = \{t \in T \ \mid \ \mathcal{L}(t) = \varepsilon\}$. Transitions in T_u are called *unobservable* or *silent*. T_o denotes 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 it is assumed 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 let C_u (C_o) be the restriction of the incidence matrix to T_u (T_o) and n_u and n_o, respectively, be 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.

The word w of events associated to sequence σ is $w = P_o(\sigma)$. 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*b*) Let $\langle N, M_0\rangle$ be a labeled net system with labeling function $\mathcal{L} : T \rightarrow L \cup \{\varepsilon\}$, where $N = (P, T, Pre, Post)$ and $T = T_o \cup T_u$. Let $w \in L^*$ be an observed word. Let

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

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

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

be the set of reachable 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.

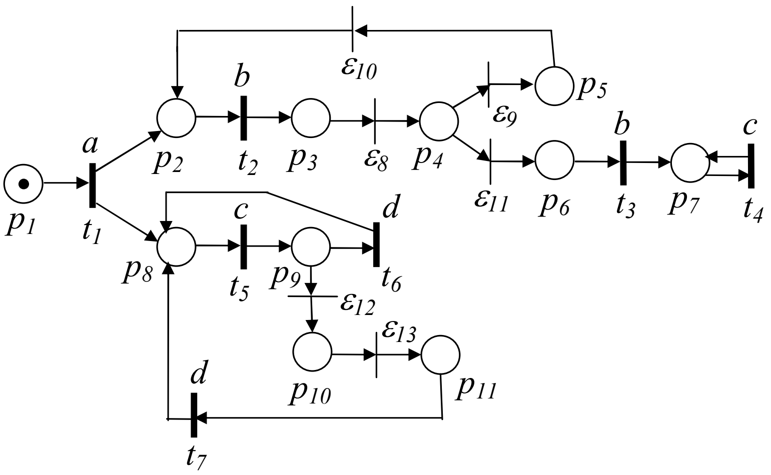

Figure 2: The PN system considered in Sections 3 to 6.

Example 3.2 Consider the PN in Fig. 2. 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 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 $w = acd$ is considered 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. ■

Finally, consider the following definition.

Definition 3.3 *Given a net $N = (P, T, Pre, Post)$, and a subset $T' \subseteq T$ of its transitions, let us define the $T'-$*induced subnet *of N as the new net $N' = (P, T', Pre', Post')$ where $Pre', Post'$ are the restrictions of $Pre, Post$ to T'. The net N' can be thought as obtained from N removing all transitions in $T \setminus T'$. Let us also write $N' \prec_{T'} N$.* ■

4 Characterization of the set of consistent markings

To solve a diagnosis problem, it is essential to be able to compute the set of sequences and markings consistent with a given observation w. In this section a formalism that allows one to characterize these sets without resorting to explicit enumeration is provided. The approach is based on the notions of minimal explanations and basis markings that are introduced in the following two subsections.

4.1 Minimal explanations and minimal e-vectors

In this subsection the notion of minimal explanation for unlabeled PNs is introduced and later it is extended to labeled PNs.

Definition 4.1 Given a marking M and an observable transition $t \in T_o$, let

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

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

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

be 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 select those whose firing vector is minimal. The firing vector of these sequences are called *minimal e-vectors.*

Definition 4.2 Given a marking M and a transition $t \in T_o$, let us define[1]

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

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

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

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

Example 4.3 Consider the PN in Fig. 2 previously introduced in Example 3.2. It holds that $\Sigma(M_0,t_1) = \{\varepsilon\}$. Then $\Sigma(M_0,t_2) = \emptyset$. Finally, let $M = [\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ ]^T$, it holds that $\Sigma(M,t_5) = \{\varepsilon, \varepsilon_8, \varepsilon_8\varepsilon_9, \varepsilon_8\varepsilon_{11}, \varepsilon_8\varepsilon_9\varepsilon_{10}\}$, while $\Sigma_{min}(M,t_5) = \{\varepsilon\}$. It follows that $Y(M,t_5) = \{[0\ 0\ 0\ 0\ 0\ 0]^T,\ [1\ 0\ 0\ 0\ 0\ 0]^T,\ [1\ 1\ 0\ 0\ 0\ 0]^T,\ [1\ 0\ 0\ 1\ 0\ 0]^T,\ [1\ 1\ 1\ 0\ 0\ 0]^T\}$, and $Y_{min}(M,t_5) = \{[0\ 0\ 0\ 0\ 0\ 0]^T\}$.

■

In Corona *et al.* (2004) it was shown 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 this paper it is suggested an approach that terminates finding all vectors in $Y_{\min}(M,t)$ if applied to nets whose T_u-induced subnet is acyclic. It simply

[1] Given two vectors x and y, $x \lneqq y$ denotes that all components of x are less than or equal to all corresponding components of y and there exists at least one component of x that is strictly less than the corresponding component of y.

requires algebraic manipulations, and is inspired by the procedure proposed by Martinez and Silva (1982) for the computation of minimal P-invariants. It can be briefly summarized by the following algorithm.

Algorithm 4.4 [Computation of $Y_{\min}(M,t)$]

```
1. Let $\Gamma := \begin{array}{|c|c|} C_u^T & I_{n_u \times n_u} \\ \hline A & B \end{array}$ where $A := (M - Pre(\cdot,t))^T$,  $B := \vec{0}_{n_u}^{\,T}$.
2. While $A$ has negative entries do,
  2.1. choose an element $A(i^*,j^*) < 0$;
  2.2. let $\mathcal{I}^+ = \{i \mid C_u^T(i,j^*) > 0\}$;
  2.3. for all $i \in \mathcal{I}^+$, add to $[A \mid B]$ a new row
         $[A(i^*,\cdot) + C_u^T(i,\cdot) \mid B(i^*,\cdot) + \vec{e}_i^{\,T}]$
       where $\vec{e}_i$ is the $i$-th canonical basis vector.
  2.4. Remove the row $[A(i^*,\cdot) \mid B(i^*,\cdot)]$ from the table.
3. Remove from $B$ any row that covers other rows.
4. Each row of $B$ is a vector in $Y_{\min}(M,t)$.
```

■

The above algorithm can be explained as follows.

Given a marking M and a transition t, Algorithm 4.4 computes the *minimal e-vectors* , i.e., the firing vectors of unobservable sequences whose firing at M is necessary to enable t.

At Step 1 a row vector is defined, $A = A(1,\cdot)$, that has a number of columns equal to number of places of the net. This vector contains a negative element $A(1,j)$ if place p_j does not enable t at M. In particular, the absolute value $|A(1,j)|$ denotes the number of tokens missing from p_j to enable t at M. Finally, B is initially a null firing vector.

While A has negative entries, one of such entries is chosen and at Step 2.2 it is checked if there exists an unobservable transition whose firing may increase the number of tokens in p_j: if so all possible such firings (of a single transition) computing the markings reached by each of these firings are considered. Vector B, in the right part of the table, denotes the corresponding firing vector. These new markings and the correspondent firing vectors will be the new rows of matrix A, while the previous row is removed.

Note that at Step 2.3 it may be possible that the new row $[A(i^*,\cdot) + C_u^T(i,\cdot) \mid B(i^*,\cdot) + \vec{e}_i^{\,T}]$ is identical to a row already in the table: if such is the case it is not necessary to add it.

The *while* loop is repeated until all markings represented by matrix A have non negative components.

4.2 Basis markings and j-vectors

In this subsection the definitions of basis markings and justifications, that are the crucial notions of the diagnosis approach presented in this paper, are introduced.

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 strictly necessary to enable w. Such a sequence of unobservable transitions is called *justification.* 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 the introduction of the following definition of pairs (sequence of transitions in T_o labeled w; corresponding justification) is needed.

Definition 4.5 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. Let

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

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

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

be 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.*

Example 4.6 Consider the PN in Fig. 2 previously introduced in Example 3.2.

Assume $w = ab$. In this case $\hat{\mathcal{J}}(w) = \{(t_1t_2, \varepsilon)\}$ and $\hat{Y}_{min}(M_0, w) = \{(t_1t_2, \vec{0})\}$. Now, consider $w = acd$. The set $\hat{\mathcal{J}}(w) = \{(t_1t_5t_6, \varepsilon), (t_1t_5t_7, \varepsilon_{12}\varepsilon_{13})\}$ and $\hat{Y}_{min}(M_0, w) = \{(t_1t_5t_6, \vec{0}), (t_1t_5t_7, [0\ 0\ 0\ 0\ 1\ 1]^T)\}$. ■

The main difference among minimal explanations and justifications is that the first ones are functions of a generic marking M and transition t, while justifications are functions of the initial marking M_0 and w. Moreover, as will be claimed in the following Proposition 4.9, in the case of acyclic unobservable subnets, justifications can be computed recursively summing up minimal explanations.

Definition 4.7 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 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 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 4.8 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. Let

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

be 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 4.9 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.

It holds:

$$\begin{aligned}\hat{Y}_{\min}(M_0, w'l) = \{(\sigma_o, y) \mid \;\; & \sigma_o = \sigma'_o t \wedge y = y' + e \; : \\ & (\sigma'_o, y') \in \hat{Y}_{\min}(M_0, w'), \\ & (t, e) \in \hat{Y}_{\min}(M'_b, l) \text{ and } \mathcal{L}(t) = l\},\end{aligned}$$

where $M'_b = M_0 + C_u \cdot y' + C_o \cdot \pi(\sigma'_o)$ and $\hat{Y}_{\min}(M'_b, l)$ is the set of pairs (transition labeled l that may have fired at M'_b, corresponding j-vector) introduced in Definition 4.5.

Proof: Let us prove this result by induction on the length of the observed string w.

(Basis step) For $w = \varepsilon$ the result trivially follows from Definitions 4.5 and 4.7.

(Inductive step) Assume the result is valid for w'. Let us prove it is also true for $w = w'l$ where $l = \mathcal{L}(t)$.

In fact, if there exists a sequence $w = w'l \in L^*$, such that $M_0[\sigma_o\rangle \tilde{M}$ with $\mathcal{L}(\sigma_o) = w$ then there exist sequences σ' and σ'' such that

$$M_0[\sigma'\rangle M'[t\rangle M''[\sigma''\rangle \tilde{M}$$

where $\mathcal{L}(\sigma') = w'$, $\mathcal{L}(t) = l$ and $\sigma'' \in T_u^*$. By induction, there exists $(M'_b, y') \in \mathcal{M}(w')$ such that

$$M_0[\sigma'_a\rangle M'_b[\sigma'_b\rangle M'[t\rangle M''[\sigma''\rangle \tilde{M}$$

where $\mathcal{L}(\sigma'_a) = w'$, $\pi(\sigma'_a) = \pi(\sigma'_o) + y'$ and $\sigma'_b \in T_u^*$. Now there exists at least one minimal explanation[2] $\sigma'_c \in \hat{\Sigma}_{\min}(M'_b, l)$ such that $\pi(\sigma'_c) \leq \pi(\sigma'_b)$ and, since the T_u-induced subnet is acyclic the state equation gives necessary and sufficient conditions for the reachability, thus the marking reached is not dependent by the order of the firing of the unobservable transitions, thus

$$M_0[\sigma'_a\rangle M'_b[\sigma'_c\rangle M'_c[t\rangle M'_d[\sigma'_d\rangle M''[\sigma''\rangle \tilde{M} \tag{1}$$

where $\pi(\sigma'_c) + \pi(\sigma'_d) = \pi(\sigma'_b)$ and $(M'_d, \pi(\sigma'_c)) \in \mathcal{M}(w'l) = \mathcal{M}(w)$. From eq. (1) it holds

$$M'_b = M_0 + C \cdot \pi(\sigma'_a) = M_0 + C_u \cdot y' + C_o \cdot \pi(\sigma'_o),$$

$$M'_d = M'_b + C_u \cdot \pi(\sigma'_c) + C_o \cdot \vec{t} = M_0 + C_u \cdot (y' + \pi(\sigma'_c)) + C_o \cdot (\pi(\sigma'_o) + \vec{t}).$$

Thus $\sigma_o = \sigma'_o t$ and $y = y' + \pi(\sigma'_c)$, where $(\sigma_0, y) \in \hat{Y}_{min}(M_0, w'l)$. □

Example 4.10 Consider the PN in Fig. 2 previously introduced in Example 3.2.

Assume $w = ab$. As shown in Example 4.6 $\hat{\mathcal{J}}(w) = \{(t_1t_2, \varepsilon)\}$, thus the basis marking is $M_b = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$, and $\mathcal{M}(w) = \{(M_b, \vec{0})\}$.

Now, consider $w = acd$. As computed in Example 4.6, the set $\hat{\mathcal{J}}(w) = \{(t_1t_5t_6, \varepsilon), (t_1t_5t_7, \varepsilon_{12}\varepsilon_{13})\}$. All 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 $\mathcal{M}(w) = \{(M_b, \vec{0}), (M_b, [0\ 0\ 0\ 0\ 1\ 1]^T)\}$.

■

By Proposition 4.9, under the assumption of acyclicity of the unobservable subnet, the set $\mathcal{M}(w)$ can be easily constructed as follows.

Algorithm 4.11 [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.
```

■

[2] In fact, being the T_u-induced subnet acyclic, this is always true for all sequences σ'_c enabled at M and such that $\pi(\sigma'_c) = \pi(\sigma'_b)$ (in such case $\pi(\sigma'_d) = \vec{0}$).

In simple words, the above algorithm can be explained as follows. Assume that, after a certain word w' has been observed, a new observable t fires and its label $l = \mathcal{L}(t)$ is observed. Consider all basis markings at the observation w' and 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, let us focus on the minimal explanations, and thus on the corresponding minimal e-vectors (Step 6.1.1). Finally, update the set $\mathcal{M}(w't)$ 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.

Definition 4.12 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. Let

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

be the set of basis markings at w. Moreover, 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.

In the following theorem a result proved for unlabeled PNs (Cabasino *et al.*, 2010) is extended to labeled PNs.

Theorem 4.13 Consider a net system $\langle N, M_0 \rangle$ whose unobservable subnet is acyclic. For any $w \in L^*$ it holds that

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

Proof: This proof has been given for unlabeled PNs in Cabasino *et al.* (2010). Also in the case of labeled PNs a formal proof can be given by induction on the length of observed string w, following the same arguments in the proof of Proposition 4.9. □

The above result shows that the set $\mathcal{C}(w)$ can be characterized in linear algebraic terms given the set $\mathcal{M}_{basis}(w)$, thus not requiring exhaustive enumeration. This is the main advantage of the approach here presented.

5 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.

Usually, fault transitions that belong the same fault class are transitions that represent similar physical faulty behavior.

The following definition introduces the notion of *diagnoser*.

Definition 5.1 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.

■

Note that assuming that certain transitions belong to the same fault class is not a restrictive assumption. On the contrary, it makes the presentation more general. If one is interested in reconstructing the occurrence of a particular transition t_f, with no ambiguity with other transitions, it is sufficient to define a fault class only containing t_f.

Example 5.2 Consider the PN in Fig. 2 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 simple words no fault of both fault classes may 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 first fault class may have occurred (firing the sequence $t_1t_2\varepsilon_8\varepsilon_{11}$) but it is not contained in any justification of *ab*, while no fault of the second fault class can have occurred.

Now, consider $w = abb$. In this case $\Delta(w, T_f^1) = 2$ and $\Delta(w, T_f^2) = 0$, being $\hat{\mathcal{J}}(w) = \{(t_1t_2t_2, \varepsilon_8\varepsilon_9\varepsilon_{10}), (t_1t_2t_3, \varepsilon_8\varepsilon_{11})\}$ and $\mathcal{S}(w) = \{t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2,\ t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2\varepsilon_8, t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2\varepsilon_8\varepsilon_9,$ $t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2\varepsilon_8\varepsilon_9\varepsilon_{10},\ t_1t_2\varepsilon_8\varepsilon_9\varepsilon_{10}t_2\varepsilon_8\varepsilon_{11}, t_1t_2\varepsilon_8\varepsilon_{11}t_3\}$. This means that no fault of the second fault class can have occurred, while a fault of the first fault class may have occurred since one justification does not contain ε_{11} and one justification contains it.

Finally, consider $w = abbccc$. In this case $\Delta(w, T_f^1) = 3$ and $\Delta(w, T_f^2) = 1$. In fact since $\hat{\mathcal{J}}(w) = \{(t_1t_2t_3t_5t_4t_4, \varepsilon_8\varepsilon_{11}), (t_1t_2t_3t_4t_5t_4, \varepsilon_8\varepsilon_{11}),\ (t_1t_2t_3t_4t_4t_5, \varepsilon_8\varepsilon_{11}),\ (t_1t_2t_3t_4t_4t_4, \varepsilon_8\varepsilon_{11})\}$ a fault of the first fault class must have occurred, while a fault of the second fault class may have occurred (e.g. $t_1t_2\varepsilon_8\varepsilon_{11}t_3t_4t_4t_5\varepsilon_{12}$) but it is not contained in any justification of w. ■

The following two results proved in Cabasino *et al.* (2010) for unlabeled PNs still hold in the case of labeled PNs. In particular, the following proposition presents how the diagnosis states can be characterized analyzing basis markings and justifications.

Proposition 5.3 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 $y(t_f) > 0$.

Proof: By Definition 5.1, $\Delta(w, T_f^i) = 0$ iff no fault transition $t_f \in T_f^i$ is contained in any firing sequence that is consistent with w, while $\Delta(w, T_f^i) = 1$ iff no fault $t_f \in T_f^i$ is contained in any justification of w but there exists at leat one sequence that is consistent with w that contains a transition $t_f \in T_f^i$. Therefore, a necessary and sufficient condition to have $\Delta(w, T_f^i) \in \{0, 1\}$ is that for all j-vectors y at w and all $t_f \in T_f^i$ it is $y(t_f) = 0$, thus proving the first item.

Analogously, $\Delta(w, T_f^i) = 2$ if a transition $t_f \in T_f^i$ is contained in at least one (but not in all) justification of w. Thus, to have $\Delta(w, T_f^i) = 2$ it is necessary and sufficient that there exists at least one j-vector y that contains at least one transition $t_f \in T_f^i$ and one j-vector y' that does not contain transitions $t_f \in T_f^i$, thus proving the second item.

Finally, given an observed word w and a fault class T_f^i it holds $\Delta(w, T_f^i) = 3$ if all firable sequences consistent with w contain at least one fault transition $t_f \in T_f^i$. Thus, to have $\Delta(w, T_f^i) = 3$ it is necessary and sufficient that all justifications contain at least one transition $t_f \in T_f^i$. This proves the third item. □

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

Proposition 5.4 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 $y(t_f) = 0 \; \forall \; t_f \in T_f^i$. Consider the constraint set

$$\mathcal{T}(M, T_f^i) \;\; = \;\; \begin{cases} M + C_u \cdot z \geq \vec{0}, \\ \displaystyle\sum_{t_f \in T_f^i} z(t_f) > 0, \\ z \in \mathbb{N}^{n_u}. \end{cases} \tag{2}$$

- $\Delta(w, T_f^i) = 0$ if $\forall \; (M, y) \in \mathcal{M}(w)$ the constraint set (2) is not feasible.
- $\Delta(w, T_f^i) = 1$ if $\exists \; (M, y) \in \mathcal{M}(w)$ such that the constraint set (2) is feasible.

Proof: Let $w \in L^*$ be an observed word such that $\forall (M, y) \in \mathcal{M}(w)$ it is $y(t_f) = 0 \; \forall \; t_f \in T_f^i$. By Definition 5.1 it immediately follows that:

- $\Delta(w, T_f^i) = 0$ if $\forall (M, y) \in \mathcal{M}(w)$ and $\forall t_f \in T_f^i$ there does not exist a sequence $\sigma \in T_u^*$ such that $M[\sigma\rangle$ and $t_f \in \sigma$;
- $\Delta(w, T_f^i) = 1$ if $\exists$ at least one $(M, y) \in \mathcal{M}(w)$ and a sequence $\sigma \in T_u^*$ such that for at least one $t_f \in T_f^i$, $M[\sigma\rangle$ and $t_f \in \sigma$.

Now, as proved in Corona *et al.* (2007) if a generic PN is *acyclic* its state equation gives necessary and sufficient conditions for marking reachability. Therefore, if such a result is applied to the unobservable subnet, that is acyclic by assumption, it can be concluded that the set $\mathcal{T}(M, T_f^i)$ characterizes the reachability set of the unobservable net at marking M. Thus, due to this fact and the above two items, it can be concluded that there exists a sequence containing a transition $t_f \in T_f^i$ firable at M on the unobservable subnet if and only if $\mathcal{T}(M, T_f^i)$ 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 (2) admits a solution.

Example 5.5 Consider the PN in Fig. 2 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 $M_b^1 = [0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$. Being $\mathcal{T}(M_b^1, T_f^i)$ feasible only for $i = 1$ 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 $\mathcal{T}(M_b^1, T_f^2)$ and $\mathcal{T}(M_b^2, T_f^2)$ 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 $\mathcal{T}(M_b^4, T_f^2)$ feasible it holds $\Delta(w, T_f^2) = 1$. ■

The approach described above requires to compute for each observed word w and for each fault class i a diagnosis state $\Delta(w, T_f^i)$. Let us conclude this section with a brief discussion on the definition of diagnosis states $\Delta = 1$ and $\Delta = 2$. Firstly, observe that both the diagnosis states correspond to *uncertain states* even if a higher degree of alarm is associated to $\Delta = 2$ with respect to $\Delta = 1$. Secondly, observe that an advantage in terms of computational complexity can be obtained by splitting the uncertain condition in two diagnosis states, namely $\Delta = 1$ and $\Delta = 2$. In fact, the diagnosis approach is based on the preliminary computation of the set $\mathcal{M}(w)$. If $\Delta = 2$ or $\Delta = 3$ no additional computation is required. On the contrary to distinguish among $\Delta = 0$ and $\Delta = 1$ an integer programming problem should be solved.

6 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 the fault class T_f^i 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 $\mathcal{T}(M_b, T_f^i)$.

In this section it is shown 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 6.1 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 $\mathcal{T}(M, T_f^i)$ 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 6.2 [Computation of the BRG]

```
1. Label the initial node $(M_0, x_0)$ where $\forall i = 1, \ldots, r$,
     $x_0(T_f^i) = \begin{cases} 1 & \text{if } \mathcal{T}(M_0, T_f^i) \text{ is feasible,} \\ 0 & \text{otherwise.} \end{cases}$
   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 $\forall i = 1, \ldots, r$,
                  $x'(T_f^i) = \begin{cases} 1 & \text{if } \mathcal{T}(M', T_f^i) \text{ is feasible,} \\ 0 & \text{otherwise.} \end{cases}$
                  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 fault may occur at M_0. Now, 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 each of these transitions 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 the approach here presented always requires to enumerate a state space that is a subset (usually a strict subset) of the reachability space. However, as in general for diagnosis approaches, the combinatory explosion cannot be avoided.

Example 6.3 Consider again the PN in Fig. 2, 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 Fig. 3. 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 $\mathcal{T}(M_0, T_f^i)$ is not feasible for $i = 1$ and $i = 2$. From node M_0 to node M_1 there is one arc labeled a with the null vector as minimal explanation. The node containing the

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

Table 1: The markings of the BRG in Fig. 3.

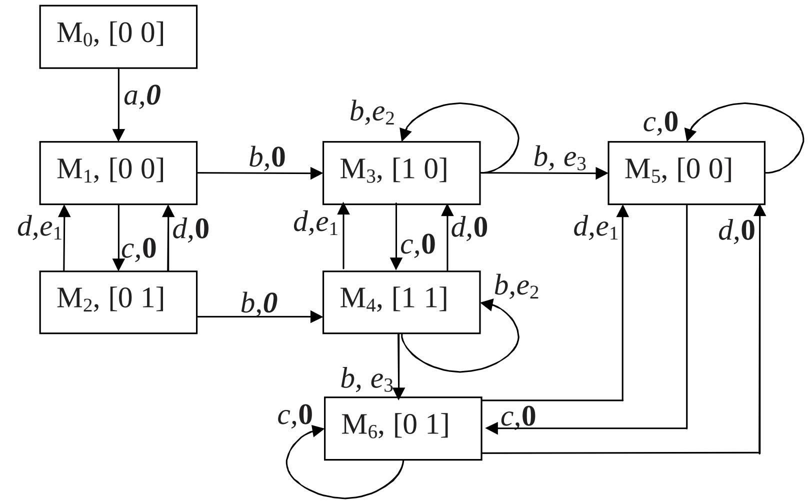

Figure 3: The BRG of the PN in Fig. 2.

basis marking M_2 has binary vector [0 1], because $\mathcal{T}(M_2, T_f^i)$ is feasible only for $i = 2$. 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 6.4 [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
   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 $i = 1, \dots, r$, 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 6.4 enable 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. 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 are included in the set $\mathcal{M}(l)$. 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, one has to 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. Include in $\mathcal{M}(w) = \mathcal{M}(w'l)$ 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 6.4 computes the diagnosis state. Consider the generic ith fault class. If $\forall (M, y) \in \mathcal{M}(w)$ and $\forall t_f \in T_f^i$ it holds $y(t_f) = 0$, the ith entry of all the binary row vectors associated to the basis markings M has to be checked, such that $(M, y) \in \mathcal{M}(w)$. If these entries are all equal to 0, it holds $\Delta(w, T_f^i) = 0$, otherwise it holds $\Delta(w, T_f^i) = 1$. 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 6.5 Consider the PN in Fig. 2 and its BRG in Fig. 3. Let $w = \varepsilon$. By looking at the BRG it holds 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, consider $w = ab$. In such a case $\mathcal{M}(w) = \{(M_3, \vec{0})\}$. It holds $\Delta(ab, T_f^1) = 1$ and

	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

Table 2: The e-vectors of the BRG in Fig. 3.

$\Delta(ab, T_f^2) = 0$ 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 $\mathcal{M}(w) = \{(M_4, y_1), (M_5, y_2), (M_6, y_3)\}$, where $y_1 = e_2$, $y_2 = y_3 = e_3$, and the row vectors associated to M_4 and M_5 are respectively [1 1], [0 0] and [0 1]. ■

Let us conclude this section observing that the BRG is a graph containing all information necessary for the construction of an observer. In the case of bounded PNs a modified version of the BRG is used to build the diagnoser that it is used to study the diagnosability of the system (Cabasino *et al.*, 2009*a*). Note that, if an automaton has a number N of states, in the worst case (that depends on the labeling of events) the cardinality of the set of nodes of its observer is $2^N - 1$ (Cassandras and Lafortune, 2007). On the contrary, the number of nodes of the BRG is equal to the number of basis markings that is at most equal to the number of reachable markings.

7 Matlab toolbox

In this section it is briefly illustrated a MATLAB function (BRG.m) that, given a bounded labeled PN, builds the basis reachability graph. This function, together with other MATLAB functions for diagnosis of labeled PNs, can be downloaded on the web (Pocci, 2009).

The inputs 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: each row contains the indices of the transitions that belong to the corresponding class;
- a cell array L that has as many rows as the cardinality of the considered alphabet: each row contains the indices of the observable transitions having the corresponding label;
- a cell array E that contains in each row a character (or a string of characters) defining a label of the considered alphabet. The cell array E is ordered according to L.

As an example, for the PN in Fig. 2 introduced in the Example 3.2 the cell arrays F, L and E are:

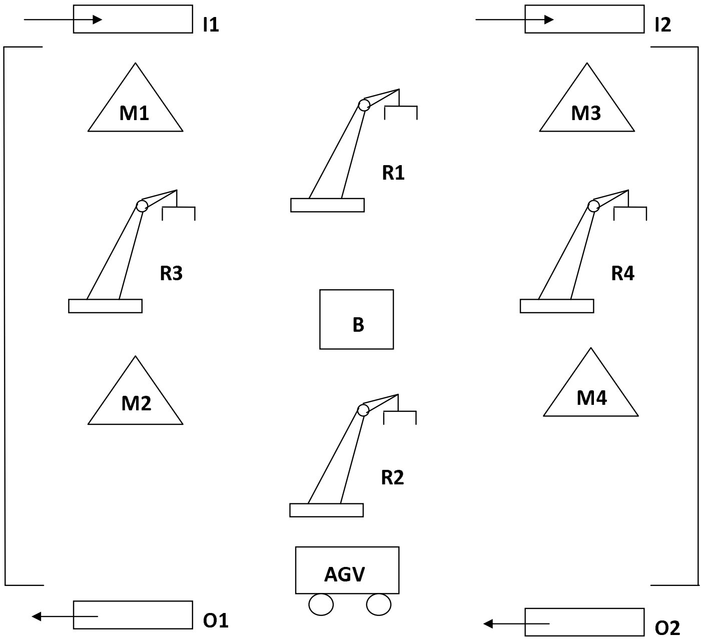

Figure 4: Layout of the manufacturing system.

$$F = \{[11];[12]\}, \quad L = \{[1];[2\ \ 3];[4\ \ 5];[6\ \ 7]\}, \quad E = \{['a'];['b'];['c'];['d']\}.$$

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;
- the transpose of the basis marking M_b^i associated to the node;
- a vector with as many columns as the number of fault classes: the jth element is equal to $x_i(T_f^j)$ evaluated at M_b^i. Thus, $x_i(T_f^j) = 0$ if $\mathcal{T}(M_b^i, T_f^j)$ is not feasible, 1 otherwise;
- the indices of the transitions enabled at the node;
- the identifier number of the nodes that are reached firing an enabled transition and the corresponding j-vector.

8 A manufacturing example

In this section the diagnosis approach proposed in Section 5 is applied to an example modeling a manufacturing system. The considered net is similar to the one described in Zhou and DiCesare (1993). The automated manufacturing system layout is shown in Fig. 4 and the corresponding

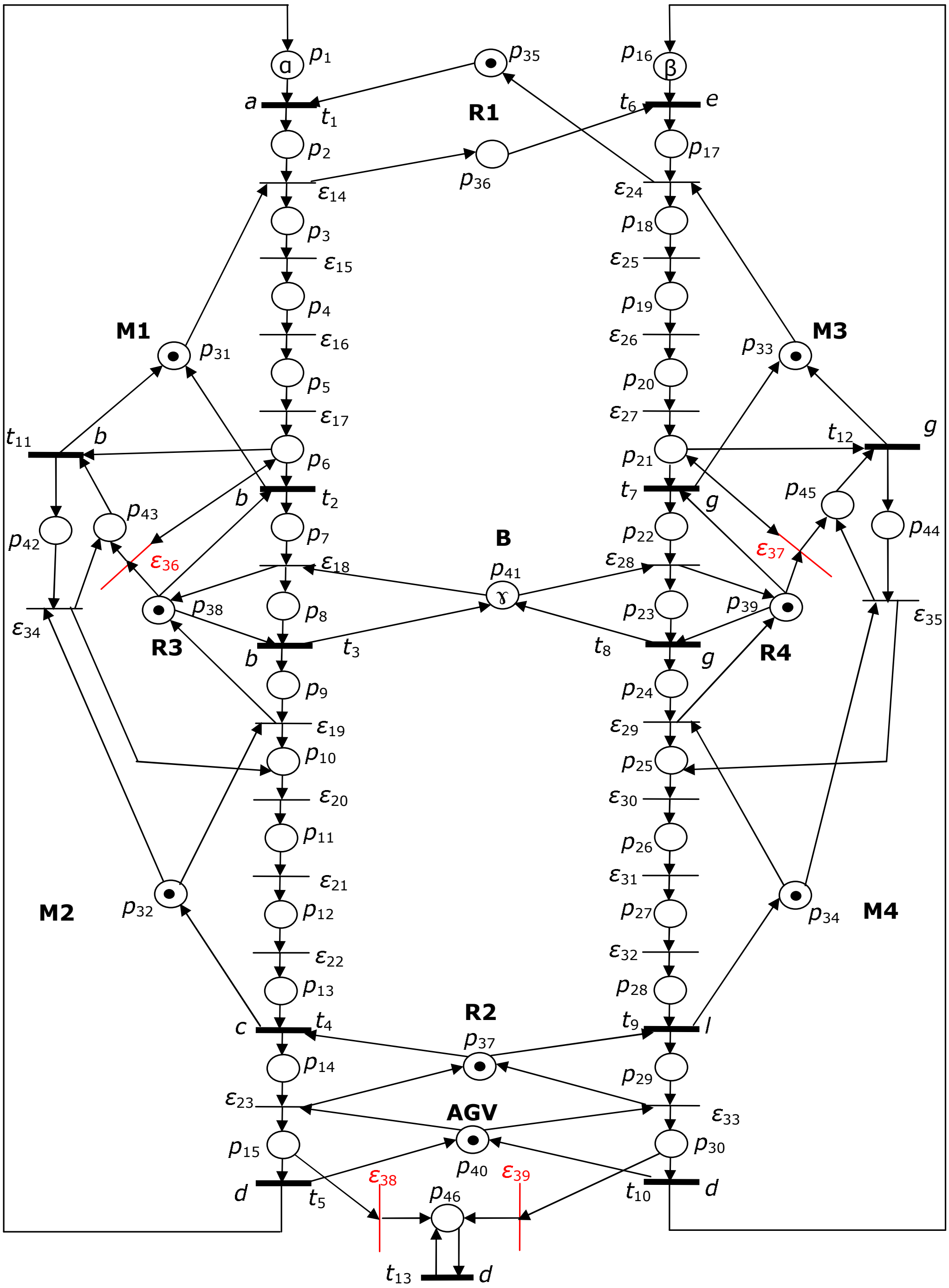

Figure 5: Petri net model of the manufacturing system in Fig. 4.

PN is depicted in Fig. 5, where thick transitions represent observable events and thin transitions represent unobservable events.

The plant consists of four machines (M1 to M4), four robots (R1 to R4), one AGV system (AGV), one buffer of finite capacity (B), two inputs of parts to be processed (I1 and I2) and two outputs for the processed parts (O1 and O2). The two production lines produce two different kinds of final product.

This PN has 46 places and 39 transitions. The marking of place p_{41} (γ) represents the number of free slots of the buffer, while α (the marking of place p_1) and β (the marking of place p_{16}) represent the number of parts of type 1 and 2, respectively. Places from 42 to 46 represent the faulty behavior (in the sense that these places are marked only if a fault has occurred).

The set of observable transitions T_o is composed by transitions from t_1 to t_{13}, the set of unobservable but regular transitions T_{reg} is composed by transitions from ε_{14} to ε_{35} and the set of fault transitions is partitioned into three fault classes: $T_f^1 = \{\varepsilon_{36}\}$, $T_f^2 = \{\varepsilon_{37}\}$ and $T_f^3 = \{\varepsilon_{38}, \varepsilon_{39}\}$. The first fault class models a fault in the robot R3 that moves a part from the output buffer of machine M1 to the input buffer of machine M2, rather than putting it in the buffer B. Analogously, the second fault class models a fault in the robot R4 that moves a part from the output buffer of machine M3 to the input buffer of machine M4, rather than putting it in the buffer B. Finally, the third fault class models a fault in the AGV. In particular, when the AGV is working correctly, a processed part exits the system and a new one is admitted. If a fault in the AGV occurs parts do not exit the production lines, and are not replaced by new input parts. However, in faulty behavior the sensors associated to the AGV may indefinitely produce the same signal they provide when a part regularly exits the production line.

Assume that each robot is equipped of a sensor that observes each time that the robot picks up a part. In particular, for R1 $\mathcal{L}(t_1) = a$ and $\mathcal{L}(t_6) = e$, for R2 $\mathcal{L}(t_4) = c$ and $\mathcal{L}(t_9) = l$, for R3 $\mathcal{L}(t_2) = \mathcal{L}(t_3) = \mathcal{L}(t_{11}) = b$, for R4 $\mathcal{L}(t_7) = \mathcal{L}(t_8) = \mathcal{L}(t_{12}) = g$. Moreover assume it is possible to observe each time that a part is moved by the AGV $\mathcal{L}(t_5) = \mathcal{L}(t_{10}) = \mathcal{L}(t_{13}) = d$.

In this section the results of the computation of the BRG for several initial states are presented. In particular, the cardinality of the number of nodes of the BRG, denoted as $|BRG|$, are summarized in Table 3 for different values of α and β.

The table also shows the cardinality of the reachability set R , i.e., $|R|$. This is an extremely important parameter to appreciate the advantage of using basis markings rather than exhaustively enumerating the set of reachable states, as it typically occurs in the automata based approaches. The value of $|R|$ has been computed using the PN tool TINA (Time PNs Analyzer) (see TINA website: http://homepages.laas.fr/bernard/tina).

— Column 1: $(\alpha+\beta)$ represents the total number of parts to be processed by the two production lines.

— Columns 2 and 3: α and β represent the number of parts to be processed in the first and second production line, respectively.

$\alpha+\beta$	α	β	$\|R\|$	$\|BRG\|$
2	2	0	27	9
2	1	1	1,640	170
2	0	2	1	1
3	3	0	27	9
3	2	1	10,260	604
3	1	2	10,260	604
3	0	3	1	1
4	4	0	27	9
4	3	1	35,098	1,343
4	2	2	62,210	2,128
4	1	3	35,098	1,343
5	5	0	27	9
5	4	1	78,404	2,294
5	3	2	205,761	4,691
5	2	3	205,761	4,691
5	1	4	78,404	2,294
6	6	0	27	9
6	5	1	131,614	3,325
6	4	2	448,306	7,963
6	3	3	655,472	10,250
6	2	4	448,306	7,963
6	1	5	131,614	3,325
7	7	0	27	9
7	6	1	186,808	4,373
7	5	2	741,035	11,503
7	4	3	1,383,391	17,273
7	3	4	1,383,391	17,273
7	2	5	741,035	11,503
7	1	6	186,808	4,373

Table 3: Numerical results in the case of $\gamma = 8$.

— Column 4 shows the number of nodes $|R|$ of the reachability graph.

— Column 5 shows the number of nodes $|BRG|$ of the BRG.

Table 3 shows how the state space of the reachability graph highly increases with the number of pallets circulating in the first and in the second production line (places p_1 and p_{16}). In particular, it increases exponentially. Note that, also in the case of the BRG the number of nodes increases exponentially, but much more slowly with respect to the cardinality of R.

Since robot R1 always starts taking one part from the first production line, all cases in which

α is equal to zero present only one node corresponding to M_0 both in the BRG and in the reachability graph. Moreover all cases in which β is equal to zero present 9 nodes in the BRG and 27 nodes in the reachability graph corresponding respectively to the number of basis markings and consistent markings that can be reached when only one part is introduced in the first production line. Finally, since the first production line is perfectly symmetric to the second one the number of nodes both in the BRG and in the reachability graph does not change exchanging α with β. This is shown in Table 3.

For the considered PN, on the basis of the above simulations, it can be concluded 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 more compact manner.

Finally, remark that, although in this paper the problem of diagnosability is not addressed, the manufacturing system illustrated in this section is diagnosable for any value of α, β and γ. The diagnosability of this system has been tested using the MATLAB tool in Pocci (2009).

9 Conclusions and future work

This paper presents a diagnosis approach for labeled PNs using basis markings. This enables one to avoid an exhaustive enumeration of the reachability set. This approach applies to all bounded and unbounded PN systems whose unobservable subnet is acyclic. However, if bounded net systems are considered the most burdensome part of the procedure may be moved off-line computing the Basis Reachability Graph. Finally, a tool for the diagnosis of labeled bounded PNs has been presented and the simulations results using a net system taken from the manufacturing domain have been shown.

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.

References

Basile, F., P. Chiacchio and G. De Tommasi (2009). An efficient approach for online diagnosis of discrete event systems. *IEEE Trans. on Automatic Control* **54**(4), 748–759.

Baviehi, S. and E.K.P. Chong (1994). Automated fault diagnosis using a discrete event systems. In: *1994 IEEE Int Symposium on Intelligent Control.* Ohio, USA.

Benveniste, A., E. Fabre, S. Haar and C. Jard (2003). Diagnosis of asynchronous discrete event systems: A net unfolding approach. *IEEE Trans. on Automatic Control* **48**(5), 714–727.

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

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

Cabasino, M.P., A. Giua and C. Seatzu (2009*a*). Diagnosability of bounded Petri nets. In: *Proc. 48th IEEE Conf. on Decision and Control.* Shanghai, China.

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

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

Cabasino, M.P., A. Giua, S. Lafortune and C. Seatzu (2009*c*). Diagnosability analysis of unbounded Petri nets. In: *Proc. 48th IEEE Conf. on Decision and Control.* Shanghai, China.

Cassandras, C.G. and S. Lafortune (2007). *Introduction to discrete event systems, Second Edition.* Springer.

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

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

Corona, D., A. Giua and C. Seatzu (2007). Marking estimation of Petri nets with silent transitions. *IEEE Trans. on Automatic Control* **52**(9), 1695–1699.

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

Dotoli, M., M.P. Fanti, A. Mangini and W. Ukovich (2010). Identification of the unobservable behaviour of industrial automation systems by Petri nets. *Control Engineering Practice.* (In press).

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

Garcia, E., A. Correcher, F. Morant, E. Quiles and R. Blasco (2005). Modular fault diagnosis based on discrete event systems. *Discrete Event Dynamic Systems* **15**(3), 237–256.

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

Ghazel, M., A. Togueni and M. Bigang (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.

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

Huang, Z., V. Chandra, S. Jiang and R. Kumar (2003). Modeling discrete event systems with faults using a rules based modeling formalism. *Mathematical and Computer Modelling of Dynamical Systems* **9**(3), 233–254.

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

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

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

Lunze, J. and J. Schroder (2004). Sensor and actuator fault diagnosis of systems with discrete inputs and outputs. *IEEE Transactions on Systems, Man, and CyberneticsPart B: Cybernetics* **34**(3), 1096–1107.

Martinez, J. and M. Silva (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.*. Springer-Verlag. pp. 301–310.

Miyagi, P.E. and L.A.M. Riascos (2010). Modeling and analysis of fault-tolerant systems for machining operations based on Petri nets. *Control Engineering Practice* **14**(4), 397–408.

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

Pocci, Marco (2009). The MATLAB toolbox is available at the following web address: *http://www.diee.unica.it/giua/TESI/09_Marco.Pocci/PN_DIAG.zip.*

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

Ramirez-Treviño, A., E. Ruiz-Beltràn, I. Rivera-Rangel and E. Lopez-Mellado (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. (1995). A Discrete Event Systems Approach to Failure Diagnosis. PhD thesis. The University of Michigan. Ann Arbor, Michigan.

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

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

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

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

Ushio, T., L. Onishi and K. Okuda (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)*. pp. 113–118.

Viswanadham, N. and T. L. Johnson (1988). Fault detection and diagnosis of automated manufacturing systems. In: *Proc. 27th IEEE Conf. on Decision and Control*. Austin, Texas. pp. 2301–2306.

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

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

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

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

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

Zhou, M.C. and F. DiCesare (1993). *Petri net synthesis for discrete event control manufacturing systems*. Kluwer.