Cybernetics and Intelligence

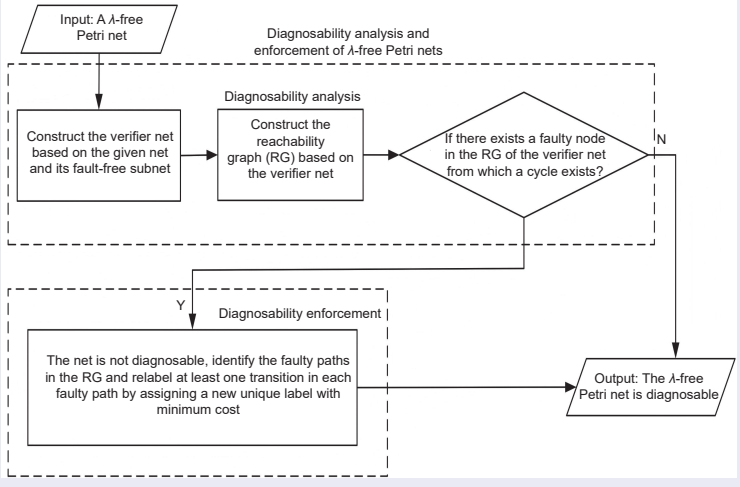
Research Article

Diagnosability analysis and enforcement of λ-free Petri nets

Ning Ran[1], Xinyue Gu[1], Haijun Jia[2], Ruicai Si[3], Zijian Dong[4], Zhou He[5], Jinyuan Hao[6,✉]

Cite this article: Ran N, Gu XY, Jia HJ, et al. *Cybernetics and Intelligence* 2026, **1**(3): 9390021. https://doi.org/10.26599/CAI.2026.9390021

ABSTRACT: This paper proposes an approach for diagnosability analysis and enforcement of λ-free Petri nets. First, a verifier net is constructed based on the system model and its fault-free subnet. Then, the reachability graph of the verifier net is constructed, and a necessary and sufficient condition is established to determine whether the system is diagnosable. If the system is not diagnosable, a diagnosability enforcement strategy is further proposed. Specifically, we present a systematic procedure to construct a new labeling function that ensures the diagnosability of the system while minimizing cost. The construction follows a set of labeling rules, and an integer linear programming model is employed to obtain the optimal labeling scheme.



KEYWORDS: diagnosability analysis; diagnosability enforcement; integer linear programming (ILP); Petri nets (PN)

1 Introduction

In the fault diagnosis of discrete event systems (DESs) [1–3], diagnosability analysis and enforcement are two core issues. Diagnosability determines whether faults can be detected using limited observed information when they occur and is a prerequisite for effective fault diagnosis. Over the past few decades, diagnosability has been extensively studied in both automata and Petri nets.

Sampath et al. [4] first introduce the concept of diagnosability. They propose an approach based on constructing a diagnoser. The necessary and sufficient conditions for diagnosability are also provided by them. Hashtrudi Zad et al. [5] propose an approach for online passive fault diagnosis modeled as finite-state automata based on state estimation. In their approach, a diagnoser is constructed to update the set of possible system states according to the observed output sequence, without requiring synchronized initialization between the plant and the diagnoser. They further introduce a model reduction scheme to alleviate the exponential complexity of diagnoser construction. Furthermore, Basilio and Silva [6] propose an automaton-based diagnoser and present necessary and sufficient conditions for verifying K-diagnosability. This work further enriches automaton-theoretic approaches to diagnosability.

When a system contains multiple interacting components, automata-based models inevitably suffer from the state space combinatorial explosion problem. Petri nets [7, 8] provide a compact representation that can effectively describe the behavior of concurrent systems. Madalinski et al. [9] present a diagnosability analysis approach based on unfolded Petri nets. Instead of constructing a diagnoser or a verifier, their method explores the unfolding structure of the net and checks whether there exist two infinite sequences with identical observations but different fault occurrences. On the other hand, Cabasino et al. [10] construct a verifier net from the original Petri net model. Based on its reachability and coverability graphs, they propose the concepts of diagnosability and K-step diagnosability for bounded and unbounded Petri nets, and present an approach to compute the value of K. They also develop a linear programming method for the diagnosability analysis of unbounded nets. Although the implementation differs, the basic idea of verifying diagnosability via a verifier net is similar to that in this work.

Basile et al. [11] and Dotoli et al. [12] estimate the number of faults during system operation by solving a series of integer linear

[1] College of Electronic and Information Engineering, Hebei University, Baoding 071002, China. [2] Jingneng Xilingol Energy Co., Ltd., Inner Mongolia 011400, China. [3] Jilin Electric Power Research Institute Co., Ltd., Changchun 130021, China. [4] Department of Automation, North China Electric Power University, Baoding 071003, China. [5] School of Electrical and Control Engineering, Shaanxi University of Science and Technology, Xi'an 710021, China. [6] HBU-UCLan School of Media, Communication and Creative Industries, Hebei University, Baoding 071002, China.

✉ Corresponding author. E-mail: hjy@hbu.edu.cn

Received: April 24, 2026; Revised: June 2, 2026; Accepted: June 20, 2026

programming (ILP) problems, and further determine the real-time operating states of systems. Some scholars propose bounded Petri net diagnosability analysis methods based on basis marking [13–15], which reduce redundant state space enumeration. Jiroveanu and Boel [16] introduce the minimum explanation into the fault diagnosis of bounded labeled Petri nets and propose a low complexity fault diagnosability analysis algorithm. Lefebvre [17, 18] investigates the fault diagnosis problem of Petri nets, calculates valid transition sequences and reachable state sets from system observations, and utilizes such information to verify fault diagnosability and design diagnosers. Paiva et al. [19] study the online diagnosis and language diagnosability verification of DESs based on acyclic labeled Petri nets. In recent years, Miao et al. [20] study the computational complexity of diagnosability verification for automata and labeled Petri nets, revealing fundamental complexity limits of diagnosability analysis and providing theoretical insights for more efficient verification methods.

If a system is not diagnosable, its diagnosability can be enforced by modifying the initial labeling function or by using a supervisory control policy. Cabasino et al. [21] and Ran et al. [22] study the diagnosability enforcement problem in labeled Petri nets by modifying the label of transitions. In their approaches, indistinguishable observable transitions are relabeled, which essentially corresponds to replacing sensors, or unobservable transitions are assigned new labels, i.e., by assigning new sensors to these transitions. They formulate the problem as an ILP model to obtain an optimal relabeling function. Hu et al. [23] propose a systematic strategy based on a mask labeling function. Instead of adding or replacing sensors, their method silences certain observable transitions. A simplified verifier is constructed for analysis, and an ILP model is also developed to optimize the mask labeling function. On this basis, Hu et al. [24] address diagnosability verification and enforcement for labeled Petri nets with observation and control delays by integrating a delayed verifier with supervisory control.

Research on diagnosability analysis and enforcement of λ-free Petri nets [25–27] remains limited. To address this gap, this paper proposes an approach for diagnosability analysis and enforcement based on a verifier net and its reachability graph. Furthermore, additional sensors are assigned to some transitions, and an integer linear programming model is developed to obtain an optimal labeling function, thereby making originally nondiagnosable systems diagnosable.

We present the organization of the rest of this paper as follows. Section 2 introduces the basic concepts of Petri nets and λ-free Petri nets. Section 3 presents a diagnosability analysis approach based on a verifier net. Section 4 proposes a diagnosability enforcement approach via optimal sensor selection. Section 5 introduces an example to illustrate the proposed approach. Finally, Section 6 concludes this work and discusses future research directions.

2 Background on Petri nets

This section reviews the basic concepts of Petri nets and λ-free Petri nets. Further details can be found in Refs. [26, 28].

2.1 Petri nets

A Petri net (PN) is a four-tuple $N=(P,T,F,W)$, where P and T are disjoint, finite and nonempty sets. The flow relation is defined as $F\subseteq(P\times T)\cup(T\times P)$. The weight function W assigns a positive weight to an arc (x,y) if $(x,y)\in F$ and $W(x,y)=0$ otherwise, where $x,y\in P\cup T$.

The incidence matrix $[N]$ of N, where $[N](p,t)=W(t,p)-W(p,t)$, is an integer matrix of size $|P|\times|T|$. Let $x\in P\cup T$ be a node in the net N. We use ${}^\bullet x$ to denote the preset of x, which is given by ${}^\bullet x=\{y\in P\cup T\mid(y,x)\in F\}$, and $x^\bullet$ to denote its postset, i.e., $x^\bullet=\{y\in P\cup T\mid(x,y)\in F\}$.

A marking M in a PN is a mapping from the set of places P to the set of non-negative integers $\mathbb{N}=\{0,1,2,\dots\}$, where $M(p)$ represents the number of tokens contained in place p. A PN system is denoted as (N,M_0), with M_0 being the initial marking of the net.

A transition t is enabled at a marking M if $\forall p\in{}^\bullet t, M(p)\geq W(p,t)$ holds, which is denoted by $M[t\rangle$. Firing transition t yields a new marking M' such that $\forall p\in P$, $M'(p)=M(p)+[N](p,t)$, which is denoted by $M[t\rangle M'$. A transition sequence $\sigma=t_1t_2,\dots,t_k$ is enabled at marking M, denoted by $M[\sigma\rangle$. If there exists a transition sequence σ such that $M[\sigma\rangle M'$, then the marking M' is said to be reachable from M. All transition sequences that can be enabled at M_0 constitute the set $L(N,M_0)$. All markings that can be reached from M_0 constitute the reachability set $R(N,M_0)$. Let T_1 be a set of transitions. $t\in\sigma$ is used to denote that the transition t belongs to the sequence σ. Similarly, $T_1\cap\sigma\neq\emptyset$ denotes that σ contains at least one transition from T_1, whereas $T_1\cap\sigma=\emptyset$ denotes that no transition from T_1 occurs in σ.

A PN is bounded if there exists a constant $k>0$ such that $\forall p\in P,\forall M\in R(N,M_0), M(p)\leq k$. A PN is acyclic if it has no directed cycles.

A transition $t\in T$ is live at the initial marking M_0 if, for every $M\in R(N,M_0)$, there exists a marking $M'\in R(N,M)$ such that $M'[t\rangle$. The net is live if every transition $t\in T$ is live under M_0; it is dead at M_0 if no transition $t\in T$ is enabled at M_0; the net is deadlock-free if, for all $M\in R(N,M_0)$, there exists at least one transition $t\in T$ that is enabled at M.

Consider a PN system (N,M_0) and a subset of transitions $T'\subseteq T$. The subnet of N induced by T', denoted as $N'=(P,T',F',W')$, is constructed by removing all transitions in $T\setminus T'$ from the original net N. The set F' of this induced subnet is obtained by restricting the original set F to $(P\times T')\cup(T'\times P)$.

2.2 λ-free Petri nets

A λ-free PN is a four-tuple $G=(N,M_0,\Sigma,\mathcal{L})$, where (N,M_0) is a PN system, Σ is a set of labels, and $\mathcal{L}:T\to\Sigma\cup\{\varepsilon\}$ is a labeling function.

The transition set T is partitioned into two disjoint subsets, i.e., $T=T_o\cup T_f$. T_o denotes the set of observable transitions, and T_f denotes the set of fault transitions whose firing produces no observations. Transitions sharing the same label are called indistinguishable. The set of transitions labeled with e is denoted by T_e.

The labeling function is extended to strings $\mathcal{L}:T^*\to\Sigma^*$. It satisfies the following conditions:

(1) $\mathcal{L}(t)=e$ for some $e\in\Sigma$, if $t\in T_o$;

(2) $\mathcal{L}(t)=\varepsilon$, if $t\in T_f$;

(3) $\mathcal{L}(\sigma t)=\mathcal{L}(\sigma)\mathcal{L}(t)$, for $\sigma\in T^*$ and $t\in T$.

Let $\mathcal{L}^{-1}(\omega)$ denote the set of all transition sequences consistent with $\omega\in\Sigma^*$, i.e., $\mathcal{L}^{-1}(\omega)=\{\sigma\in L(N,M_0)\mid\mathcal{L}(\sigma)=\omega\}$. $\mathcal{L}(L(N,M_0))$ denotes the language of all transition labels using the extended labeling function.

Let $K\subseteq T^*$ be a language. The post-language of K after $\sigma\in T^*$ is denoted by K/σ and defined as $K/\sigma=\{\sigma'\in T^*\mid\sigma\sigma'\in K\}$.

Example 1: A λ-free PN $G=(N,M_0,\Sigma,\mathcal{L})$, as shown in Fig. 1, has an observable transition set $T_o=\{t_2,t_3,t_4\}$ and a fault transition set $T_f=\{t_f\}$. The labeling function satisfies

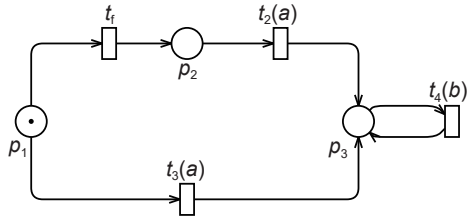


Fig. 1 A λ-free PN.

$\mathcal{L}(t_2)=\mathcal{L}(t_3)=a$ and $\mathcal{L}(t_4)=b$. The set of reachable markings is listed in Table 1.

From this example, we observe that the system has two transition sequences,

$$\sigma_1=t_f t_2(t_4)^k,\quad \sigma_2=t_3(t_4)^k,\quad k\geq 0$$

which reach the same marking $M_2=(0\,0\,1)^{\mathrm{T}}$. Moreover, these two sequences generate identical label sequences, i.e., $\mathcal{L}(\sigma_1)=\mathcal{L}(\sigma_2)=ab^k$. Therefore, based only on the label sequences, it is impossible to distinguish whether the fault transition t_f has occurred, which implies that the system is not diagnosable.

3 Diagnosability analysis of λ-free Petri nets

In this section, we address the diagnosability analysis problem. We first introduce the definition of diagnosability of λ-free PNs. We then construct a verifier net from the net and its fault-free subnet. Finally, we construct its reachability graph and present a necessary and sufficient condition for diagnosability.

3.1 The definition of diagnosability

Given a λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$, we analyze the diagnosability of λ-free PNs under the following assumptions:

(A1) The λ-free PN is bounded.

(A2) The λ-free PN is deadlock-free after firing any fault transition.

Assumption (A1) ensures that the λ-free PN has a finite state space. The diagnosability of unbounded PNs will be addressed in our future research. Assumption (A2) is a weakened version of the conventional "liveness" assumption widely used in DES fault diagnosis work, avoiding the technical issues that must be addressed when the system may deadlock after a fault.

The definition of diagnosability of a λ-free PN, inspired by Ref. [10], is given in Definition 1. The set $\Psi(T_f)$ is defined as the set of all sequences in $L(N,M_0)$ whose last transition belongs to T_f.

Definition 1: A λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$ is diagnosable with respect to the fault set T_f, if the condition (1) holds:

$$\begin{aligned}&\forall\sigma_1\in\Psi(T_f),\ \exists K\in\mathbb{N},\ \forall\sigma_2\in L(N,M_0)/\sigma_1,|\sigma_2|\geq K\\&\Rightarrow\ \forall\sigma_3\in\mathcal{L}^{-1}(\mathcal{L}(\sigma_1\sigma_2)),\ \exists t_f\in T_f:t_f\in\sigma_3\end{aligned}\tag{1}$$

By Definition 1, a λ-free PN is not diagnosable with respect to T_f if and only if there exist two transition sequences σ', $\sigma\in L(N,M_0)$ satisfying all of the following conditions:

(1) $\mathcal{L}(\sigma')=\mathcal{L}(\sigma)$;

(2) $\sigma\cap T_f\neq\emptyset$;

(3) $\sigma'\cap T_f=\emptyset$;

(4) after a fault transition fires, the length of σ can be arbitrary.

Table 1 The reachable markings of Fig. 1

M	Marking
M_0	$(1\,0\,0)^{\mathrm{T}}$
M_1	$(0\,1\,0)^{\mathrm{T}}$
M_2	$(0\,0\,1)^{\mathrm{T}}$

3.2 The verifier net

Let $G'=(N',M_0,\Sigma',\mathcal{L}')$ denote the subnet of $G=(N,M_0,\Sigma,\mathcal{L})$ induced by $T'=T\setminus T_f$. In other words, G' represents the fault-free subnet of G. We denote by $L(N',M_0)$ the language generated by G'.

Example 2: Reconsider the λ-free PN shown in Fig. 1. The corresponding fault-free subnet is shown in Fig. 2. It is obtained by removing the fault transition t_f and the arcs connected to it from $G=(N,M_0,\Sigma,\mathcal{L})$.

The verifier net [10] is synchronized on observable transitions to simulate the behavior of the system under different states. The definition of a verifier net is given as follows.

Definition 2 ([10]): Given a λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$ and its fault-free subnet $G'=(N',M_0,\Sigma',\mathcal{L}')$, the verifier net $V=(\widetilde{N},\widetilde{M}_0,\widetilde{\Sigma},\widetilde{\mathcal{L}})$ is constructed by the parallel composition of G and G', where

$$\begin{cases}\widetilde{N}=(\widetilde{P},\widetilde{T},\widetilde{Pre},\widetilde{Post})\\ \widetilde{P}=P'\cup P\\ \widetilde{T}=\widetilde{T}_o\cup(\{\varepsilon\}\times T_f)\\ \widetilde{T}_o=\{(t',t)\mid t'\in T_o,\ t\in T_o,\ \mathcal{L}(t')=\mathcal{L}(t)\}\\ \widetilde{\Sigma}=\Sigma'\times\Sigma\\ \widetilde{\mathcal{L}}:\widetilde{T}\to\widetilde{\Sigma}\end{cases}$$

The construction of a verifier net is given in Algorithm 1.

The verifier net is constructed according to Algorithm 1 by synchronizing the original PN with its fault-free subnet. Observable transitions are synchronized according to identical labels, while fault transitions are introduced independently and preserve their original flow relations. In simple words, each transition in the verifier net is represented by a pair. For observable transitions, each pair consists of two synchronized transitions, e.g., (t'_o,t_o). Each fault transition is represented by a pair consisting of ε and a fault transition, e.g., (ε,t_f). Since the label a is associated with two transitions t_2 and t_3, the verifier net contains four transitions labeled (a,a). For clarity, a double-headed arrow is used in the graph if a place p has a self-loop with a transition t, e.g., the arc between place p_3 and transition pair (t'_4,t_4).

Example 3: Consider the λ-free PN shown in Fig. 1 and its corresponding fault-free subnet in Fig. 2. The verifier net is shown in Fig. 3.

3.3 The reachability analysis of the verifier net

The reachability graph (RG) [29, 30] is a graphical structure used to represent the state change of systems, which is widely applied in

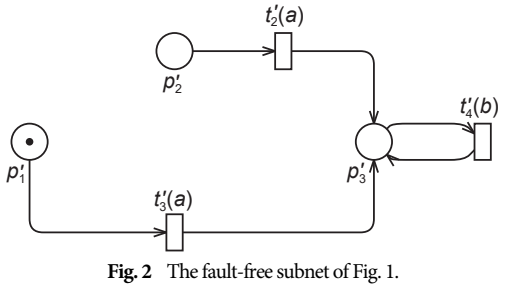


Fig. 2 The fault-free subnet of Fig. 1.

Algorithm 1 Construction of a verifier net

Input: A λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$

Output: A verifier net $V=(\widetilde{N},\widetilde{M}_0,\widetilde{\Sigma},\widetilde{\mathcal{L}})$

1: Let $G'=(N',M_0,\Sigma',\mathcal{L}')$ be the fault-free subnet of $G=(N,M_0,\Sigma,\mathcal{L})$.

2: Let $\widetilde{P}=P'\cup P$.

3: Let $\widetilde{M}_0=(M'_0,\ M_0)^{\mathrm{T}}$.

4: For all transitions $t_{\mathrm{f}}\in T_{\mathrm{f}}$:

5: • Add a transition $t\in\widetilde{T}$ denoted as $(\varepsilon,t_{\mathrm{f}})$;

6: • For all $p\in P'$, let $\widetilde{Pre}(p,t)=\widetilde{Post}(p,t)=0$;

7: • For all $p\in P$, let $\widetilde{Pre}(p,t)=Pre(p,t_{\mathrm{f}})$ and $\widetilde{Post}(p,t)=Post(p,t_{\mathrm{f}})$.

8: For all labels $e\in\Sigma$:

9: For any pair $t'_{\mathrm{o}}\in T_{\mathrm{o}}$, $t_{\mathrm{o}}\in T_{\mathrm{o}}$ with $\mathcal{L}(t'_{\mathrm{o}})=\mathcal{L}(t_{\mathrm{o}})=e$:

10: • Add a transition $t\in\widetilde{T}$ denoted as $(t'_{\mathrm{o}},t_{\mathrm{o}})$;

11: • For all $p\in P'$, let $\widetilde{Pre}(p,t)=Pre'(p,t'_{\mathrm{o}})$ and $\widetilde{Post}(p,t)=Post'(p,t'_{\mathrm{o}})$;

12: • For all $p\in P$, let $\widetilde{Pre}(p,t)=Pre(p,t_{\mathrm{o}})$ and $\widetilde{Post}(p,t)=Post(p,t_{\mathrm{o}})$.

13: • Label transition t with (e,e).

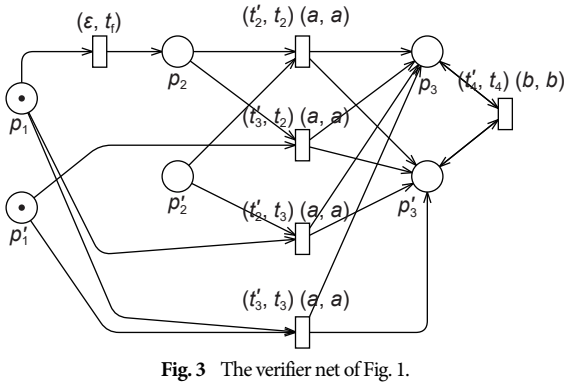


Fig. 3 The verifier net of Fig. 1.

system modeling and analysis. For PNs, it describes the relationship between markings and transition firings.

Algorithm 2 presents the approach of constructing an RG.

Definition 3: A transition sequence is said to be repetitive if there exists a marking from which it can be executed infinitely often.

By Definition 3, a repetitive sequence corresponds to a cycle in the RG of the verifier net.

Proposition 1: Given a λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$, and its verifier net $V=(\widetilde{N},\widetilde{M}_0,\widetilde{\Sigma},\widetilde{\mathcal{L}})$, if a transition sequence $\widetilde{\sigma}\in\widetilde{T}^*$ is repetitive in V, then there exist two repetitive transition sequences σ' in G' and σ in G, and both transition sequences σ' and σ have the same label sequence.

Proof: This result follows from the construction of the verifier net. In fact, the two sequences have the same label sequence by construction of the verifier net. Moreover, the existence of a sequence $\widetilde{\sigma}\in L(\widetilde{N},\widetilde{M}_0)$ implies that $\sigma'\in L(N',M_0)$ and $\sigma\in L(N,M_0)$. The transition sequences σ' and σ are repetitive, respectively, in G' and in G, given that $\widetilde{\sigma}$ is repetitive in the verifier net. ▪

Definition 4: Given a λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$ and its verifier net $V=(\widetilde{N},\widetilde{M}_0,\widetilde{\Sigma},\widetilde{\mathcal{L}})$, the set of faulty nodes in the RG of V, denoted by $\mathcal{F}(V)$, consists of all nodes that are reachable by a path containing at least one fault transition.

Algorithm 2 Construction of the RG of the verifier net

Input: A λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$ and its fault-free subnet $G'=(N',M_0,\Sigma',\mathcal{L}')$

Output: The RG

1: Construct the verifier net $V=(\widetilde{N},\widetilde{M}_0,\widetilde{\Sigma},\widetilde{\mathcal{L}})$ by Algorithm 1.

2: Let $\widetilde{M}_0$ be the initial marking with no label.

3: **while** there are nodes with no label **do**

4: Select a marking $\widetilde{M}$ with no label.

5: **for** each transition (t',t) enabled at $\widetilde{M}$ **do**

6: Let $\widetilde{M}'=\widetilde{M}+\widetilde{N}(\cdot,(t',t))$.

7: **if** $\widetilde{M}'$ does not exist in the RG **then**

8: Add $\widetilde{M}'$ to the RG.

9: Add an arc (t',t) from $\widetilde{M}$ to $\widetilde{M}'$.

10: **end if**

11: **end for**

12: Label the node $\widetilde{M}$ as old.

13: **end while**

14: Remove all labels from nodes.

Proposition 2: A λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$ is diagnosable if and only if there does not exist a cycle associated with a repetitive sequence in the RG of its verifier net that is reachable from a node in $\mathcal{F}(V)$.

Proof: (Only if) By contradiction, assume that there exists a cycle associated with a repetitive sequence in the RG of the verifier net, which is reachable from a node in $\mathcal{F}(V)$. According to Definitions 3 and 4, this implies an infinite transition sequence containing a fault in the verifier net. According to Proposition 1, there exist two transition sequences σ' in G' and σ in G, such that $\mathcal{L}(\sigma')=\mathcal{L}(\sigma)$, where σ contains at least one fault transition while σ' is fault-free. Moreover, σ can be extended to arbitrary length after the fault occurs. This violates the definition of diagnosability given in Definition 1, so the system is not diagnosable.

(If) By contradiction, assume that the system is not diagnosable. Then there exist two sequences σ' in G' and σ in G, where σ contains a fault and σ' does not, and σ can be extended infinitely. By construction of the verifier net, there exists an infinite sequence in the verifier net, which corresponds to a cycle associated with a repetitive sequence reachable from $\mathcal{F}(V)$. This contradicts the Assumption (A2). Therefore, no such repetitive sequence exists, and the system is diagnosable. ▪

Example 4: The RG constructed by Algorithm 2 is shown in Fig. 4. The set of reachable markings is listed in Table 2, and the set of faulty nodes is given by $\mathcal{F}(V)=\{\widetilde{M}_1,\widetilde{M}_2\}$. It can be observed that there exists a cycle that is reached after the occurrence of a fault transition from the initial node, namely,

$$\widetilde{M}_0\xrightarrow{(\varepsilon,t_{\mathrm{f}})}\widetilde{M}_1\xrightarrow{(t'_3,t_2)}\widetilde{M}_2\xrightarrow{(t'_4,t_4)}\widetilde{M}_2$$

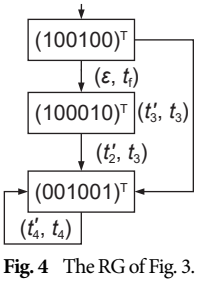


Fig. 4 The RG of Fig. 3.

Table 2 The reachable markings of Fig. 4

M	Marking
M_0	$(1\,0\,0\,1\,0\,0)^{\mathrm{T}}$
M_1	$(1\,0\,0\,0\,1\,0)^{\mathrm{T}}$
M_2	$(0\,0\,1\,0\,0\,1)^{\mathrm{T}}$

Thus the system is not diagnosable according to Proposition 2.

4 Diagnosability enforcement of λ-free Petri nets

In this section, we address diagnosability enforcement problem of λ-free PNs via optimal sensor selection. First, the diagnosability violations captured by the verifier net guide us to select the transitions that need to be relabeled. Then, we present a set of labeling rules that should be satisfied. Finally, an ILP model is employed to obtain the optimal labeling scheme with minimum cost.

4.1 Relabeling of faulty paths

Definition 5: Given a λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$ and the RG of its verifier net $V=(\widetilde{N},\widetilde{M}_0,\widetilde{\Sigma},\widetilde{\mathcal{L}})$, a sequence $\widetilde{\sigma}=(\sigma'_p\sigma'_s,\sigma_p\sigma_s)$ in the RG of V is called a faulty path with respect to $\mathcal{L}$ if the following conditions hold:

(1) $M_0[\sigma'_p\rangle M'[\sigma'_s\rangle M'$;

(2) $M_0[\sigma_p\rangle M[\sigma_s\rangle M$;

(3) $\mathcal{L}(\sigma'_p)=\mathcal{L}(\sigma_p),\quad \mathcal{L}(\sigma'_s)=\mathcal{L}(\sigma_s)$;

(4) $t_{\mathrm{f}}\in\sigma_p\sigma_s$;

(5) no prefix of $\widetilde{\sigma}$ satisfies conditions (1)–(4).

Proposition 3: A λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$ is diagnosable if and only if the RG of its verifier net $V=(\widetilde{N},\widetilde{M}_0,\widetilde{\Sigma},\widetilde{\mathcal{L}})$ has no faulty paths with respect to $\mathcal{L}$.

Proof: Straightforward from Proposition 2 and Definition 5. ▪

According to Proposition 3, a system is diagnosable if and only if there are no faulty paths in the RG of V. Therefore, if faulty paths exist, we need to examine them and determine which transitions should be "prevented" to eliminate these paths and remove the corresponding violations of diagnosability.

Consider the special labeling function $\mathcal{L}_{\text{total}}$, where each transition in $T\setminus T_{\mathrm{f}}$ is assigned a unique label that distinguishes it from all other transitions. The corresponding label set is Σ_{total}, i.e., $\Sigma_{\text{total}}=T_{\mathrm{o}}$. Without loss of generality, these unique labels are taken to be the names of the transitions, i.e., $\mathcal{L}_{\text{total}}(t)=t$ for all $t\in T\setminus T_{\mathrm{f}}$. Hence, all transitions except those in T_{f} have unique labels.

We add a new assumption:

(A3) The λ-free PN is diagnosable under the labeling function $\mathcal{L}_{\text{total}}$.

Assumption (A3) ensures that the diagnosability enforcement problem admits at least one feasible solution.

Assume that a λ-free PN system $G=(N,M_0,\Sigma,\mathcal{L})$ is not diagnosable. We aim to determine a new labeling function $\mathcal{L}_N$, such that $G_N=(N,M_0,\Sigma_N,\mathcal{L}_N)$ is diagnosable. The new labeling function is $\mathcal{L}_N: T\to\Sigma_N\cup\{\varepsilon\}$, where $\Sigma_N=\Sigma\cup\Sigma_{\text{total}}$. Furthermore, $\mathcal{L}_N$ is required to minimize a suitable objective function that captures the cost of assigning sensors to the transitions of the net.

As in Ref. [21], we relabel a transition $t\in T_{\mathrm{o}}$ according to the following rule:

(R1) For $t\in T_{\mathrm{o}}$ such that there exists $t_k\in T_{\mathrm{o}}$ with $t\neq t_k$ and $\mathcal{L}(t)=\mathcal{L}(t_k)$, we either set $\mathcal{L}_N(t)=t$ or leave it unchanged as $\mathcal{L}_N(t)=\mathcal{L}(t)$.

In Ref. [21], it has been proved that when rule (R1) is applied incrementally for relabeling, at each step, two transition sequences σ_1 and σ_2 that are distinguishable under the initial labeling function $\mathcal{L}$ will never become indistinguishable, i.e.,

$$\mathcal{L}(\sigma_1)\neq\mathcal{L}(\sigma_2)\Rightarrow\mathcal{L}_N(\sigma_1)\neq\mathcal{L}_N(\sigma_2)$$

Given a choice of $\mathcal{L}_N$, we let $T_N\subseteq T_{\mathrm{o}}$ be the transitions relabeled by $\mathcal{L}_N$, i.e.,

$$T_N=\{t\in T_{\mathrm{o}}\mid\mathcal{L}_N(t)\neq\mathcal{L}(t)\}$$

From Assumption (A3), there exists a suitable selection of T_N such that the system is diagnosable.

Note that the transition pairs in the RG of V are the form of (x',x), where

(1) x' corresponds to a transition in the fault-free system, or to ε if no transition is enabled in that system;

(2) x corresponds to a transition in the original system.

For the transition pairs that compose a faulty path, we propose the following relabeling options according to rule (R1).

(O1) (x',x) where $x'=t_i\in T_{\mathrm{o}}$ and $x=t_j\in T_{\mathrm{o}}$ with $i\neq j$. According to Algorithm 3, we know that $\mathcal{L}(t_i)=\mathcal{L}(t_j)$. Then we either set $\mathcal{L}_N(t_i)=t_i$ and $\Sigma_N=\Sigma\cup\{t_i\}$, or $\mathcal{L}_N(t_j)=t_j$ and $\Sigma_N=\Sigma\cup\{t_j\}$;

(O2) (x',x) where $x'=t_i\in T_{\mathrm{o}}$ and $x=t_j\in T_{\mathrm{o}}$ with $i=j$. In this case, relabeling a transition is irrelevant, because we are synchronizing a transition with itself when constructing the verifier net;

(O3) (ε,x) where $x\in T_{\mathrm{f}}$. In this case, we do nothing, as it is impossible to make the fault transition observable.

Since no relabeling is possible under relabeling options (O2) and (O3), these options are dropped. In view of the relabeling option (O1) and the system, we add an additional rule as follows:

(R2) For each faulty path, relabel at least one transition $t\in T_{\mathrm{o}}$ in the path from the initial marking to the end based on the relabeling option (O1).

Rule (R2) is necessary since if no transition in the faulty path is relabeled, the path still exists under the new labeling function $\mathcal{L}_N$.

Proposition 4: For every faulty path in the RG of the verifier net $V=(\widetilde{N},\widetilde{M}_0,\widetilde{\Sigma},\widetilde{\mathcal{L}})$, there must exist at least one transition that can be relabeled according to the relabeling option (O1) and associated rule (R1)

Proof: By contradiction, assume that there exists a faulty path in which all transition pairs are in the form of relabeling options (O2), (O3). In fact, we do not need to consider the relabeling option (O3) since a fault transition cannot be relabeled. For the relabeling option (O2), even if all transitions are relabeled, the faulty path still exists. Consequently, the λ-free PN system remains nondiagnosable under $\mathcal{L}_{\text{total}}$, which contradicts Assumption (A3). Hence, the result holds. ▪

Proposition 4 shows that there always exists at least one transition in each faulty path that can be relabeled.

Proposition 5: Let $\widetilde{\sigma}=(\sigma'_p\sigma'_s,\sigma_p\sigma_s)$ be a faulty path that is relabeled according to the relabeling option (O1) and associated rules (R1) and (R2). It holds that $\mathcal{L}_N(\sigma'_p\sigma'_s)\neq\mathcal{L}_N(\sigma_p\sigma_s)$.

Proof: Consider the relabeling option (O1) and let (t_i,t_j) be the transition pair of interest where $t_i\in\sigma'_p\sigma'_s$ and $t_j\in\sigma_p\sigma_s$. Since we either assign a new label to t_i, $\mathcal{L}_N(t_i)=t_i$, or to t_j, $\mathcal{L}_N(t_j)=t_j$ the label sequence $\mathcal{L}_N(\sigma'_p\sigma'_s)$ or $\mathcal{L}_N(t_{\sigma\,p}\sigma_s)$ is changed. Thus, $\mathcal{L}_N(\sigma'_p\sigma'_s)\neq\mathcal{L}_N(\sigma_p\sigma_s)$. ▪

By Proposition 5, if a faulty path is relabeled according to rules (R1) and (R2), it is no longer feasible.

Proposition 6: If each faulty path with respect to $\mathcal{L}$ is relabeled

according to the relabeling option (O1) and associated rules (R1) and (R2), then there are no more faulty paths in the λ-free PN with respect to $\mathcal{L}_N$, i.e., the relabeling does not create any new faulty paths.

Proof: By contradiction, assume that there exists a new faulty path $\widetilde{\sigma} = (\sigma'_p\sigma'_s, \sigma_p\sigma_s)$ with respect to $\mathcal{L}_N$. Let σ' and σ denote the relabeled segments of $\sigma'_p\sigma'_s$ and $\sigma_p\sigma_s$, respectively, under $\mathcal{L}_N$. These paths satisfy the following conditions:

(1) σ is arbitrarily long after the occurrence of the fault;

(2) σ' contains no fault;

(3) $\mathcal{L}_N(\sigma) = \mathcal{L}_N(\sigma')$.

Then, it follows that $\mathcal{L}(\sigma) = \mathcal{L}(\sigma')$, which implies that σ' and σ form a faulty path with respect to $\mathcal{L}$. Moreover, the non-relabeled segments of $\sigma'_p\sigma'_s$ and $\sigma_p\sigma_s$ remain unchanged under $\mathcal{L}$. However, by rule (R2), at least one transition must be relabeled in each faulty path. By the relabeling option (O1), it holds that $\mathcal{L}_N(\sigma') \neq \mathcal{L}_N(\sigma)$. This leads to a contradiction.

By Proposition 6, if a faulty path is relabeled according to rules (R1) and (R2), no new faulty paths are created.

Proposition 7: Let $G = (N, M_0, \Sigma, \mathcal{L})$ be a nondiagnosable λ-free PN system satisfying Assumptions (A1)–(A3). Let $\mathcal{L}_N$ be the new labeling function obtained according to relabeling option (O1) and associated rules (R1) and (R2) for all faulty paths of the RG of the verifier net. Then, $G_N = (N, M_0, \Sigma_N, \mathcal{L}_N)$ is diagnosable.

Proof: It is straightforward from Propositions 3, 5, and 6. Specifically, each faulty path is not feasible under the relabeling option (O1) and associated rules (R1) and (R2) according to Proposition 5. Moreover, by Proposition 6, the relabeling procedure does not create any new faulty paths. Therefore, after all faulty paths in the RG are eliminated via relabeling, no faulty paths remain. Finally, by Proposition 3, the system $G_N = (N, M_0, \Sigma_N, \mathcal{L}_N)$ is diagnosable. ▪

Example 5: Let us reconsider the RG shown in Fig. 4. A faulty path is illustrated in Fig. 5. The new labeling function $\mathcal{L}_N$, obtained according to the relabeling option (O1) and associated rules (R1) and (R2), is $\mathcal{L}_N(t_3) = t_3$ or $\mathcal{L}_N(t_2) = t_2$. Under $\mathcal{L}_N$, no faulty paths exist. Therefore, the relabeled λ-free PN system is diagnosable.

4.2 Optimal relabeling using ILP

In general, the diagnosability enforcement problem admits multiple solutions. We aim to determine a solution that minimizes a given performance index, e.g.,

$$J(\mathcal{L}_N) = \sum_{t \in T_N} c_t$$

where c_t denotes the cost of equipping transition t with a sensor that generates a unique label. If $c_t = 1$ for all $t \in T_N$, then $J(\mathcal{L}_N)$ reduces to the number of transitions relabeled from $\mathcal{L}$ to $\mathcal{L}_N$.

We aim to determine which transitions should be relabeled under $\mathcal{L}_N$. For each $t \in T_o$, define a binary variable $v_t \in \{0,1\}$,

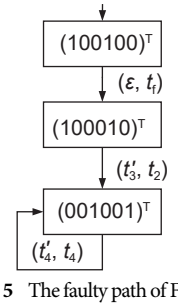


Fig. 5 The faulty path of Fig. 4.

where $v_t = 1$ indicates that transition t is relabeled, and $v_t = 0$ otherwise.

For each faulty path, we construct an inequality of the form

$$v_t + v_{t'} + v_{t''} + \cdots \geq 1 \tag{2}$$

for some transitions $t, t', t'', \ldots \in T_o$. The left-hand side of Eq. (2) is obtained by Algorithm 3.

Algorithm 3 introduces the procedure for obtaining the left-hand side of an inequality for a faulty path. There may exist multiple faulty paths in the system. Therefore, Algorithm 3 should be applied to each faulty path individually.

Let $\mathcal{R}$ be the set of all linear inequalities obtained from every faulty path in the RG of V.

Proposition 8: Let $G = (N, M_0, \Sigma, \mathcal{L})$ be a nondiagnosable λ-free PN system satisfying Assumptions (A1)–(A3). Let $\mathcal{L}_N$ be the new labeling function obtained according to the relabeling option (O1) and associated rules (R1) and (R2) for faulty paths of the RG of the verifier net $V = (\widetilde{N}, \widetilde{M}_0, \widetilde{\Sigma}, \widetilde{\mathcal{L}})$. $G = (N, M_0, \Sigma_N, \mathcal{L}_N)$ is diagnosable if and only if the set $\mathcal{R}$ of linear inequalities has a feasible solution.

Proof: (Only if) According to the relabeling option (O1) and associated rules (R1) and (R2), the set of linear inequalities $\mathcal{R}$ is constructed. For each faulty path, we relabel at least one transition $t \in T_o$ according to rule (R1) with (O1). More specifically, rule (R1) is enforced by adding $+v_i + v_j$ to the left-hand side of Eq. (2). Rule (R2) is enforced by the constraint ≥ 1, which guarantees that each faulty path is not feasible. Hence, if there exists a labeling function $\mathcal{L}_N$ obtained by the relabeling option (O1) and associated rules (R1) and (R2) such that $G_N = (N, M_0, \Sigma_N, \mathcal{L}_N)$ is diagnosable, then the set $\mathcal{R}$ of linear inequalities admits a feasible solution.

(If) The result follows straightforwardly from Proposition 7. ▪

In practice, transition relabeling corresponds to sensor deployment or sensor reconfiguration, which is typically associated with hardware, communication, and maintenance costs. To minimize the economic burden, it is desirable to reduce the number of relabeled transitions while ensuring the system is diagnosable. This can be achieved by solving the ILP problem as Eq. (3).

$$\begin{cases} \min & \displaystyle\sum_{t \in T_N} c_t v_t \\ \text{s.t.} & \mathcal{R} \end{cases} \tag{3}$$

Algorithm 3 Construction of the left-hand side of an inequality for faulty path relabeling

Input: A faulty path in a λ-free PN system $G = (N, M_0, \Sigma, \mathcal{L})$

Output: The left-hand side of Eq. (2)

1: Initialize the left-hand side of the inequality to 0.

2: Examine each transition pair (x', x) in the faulty path.

3: **for** each transition pair (x', x) **do**

4: **if** $x' = t_i \in T_o$ and $x = t_j \in T_o$ with $i \neq j$ **then**

5: Add $+v_i + v_j$ to the left-hand side.

6: **else if** $x' = t_i \in T_o$ and $x = t_j \in T_o$ with $i = j$ **then**

7: Do nothing.

8: **else if** $x' = \varepsilon$ and $x \in T_f$ **then**

9: Do nothing.

10: **end if**

11: **end for**

Example 6: Consider the faulty path in Fig. 5, and examine the transition pairs. According to Algorithm 3, the transition pair (t'_3, t_2) adds $+v_{t_3}+v_{t_2}$ to the left-hand side of Eq. (2), while the transition pairs (t'_4, t_4) and (ε, t_f) are not processed. Thus, the linear inequality can be written as Eq. (4):

$$v_{t_3}+v_{t_2}\geq 1 \tag{4}$$

Assume that $c_t=1$ for all transitions $t\in T_o$. By solving the ILP problem in Eq. (3), the optimal solution is obtained by relabeling transition t_2. If we relabel

$$\mathcal{L}_N(t_2)=t_2$$

the system becomes diagnosable.

Note that there exists another optimal solution, i.e.,

$$\mathcal{L}_N(t_3)=t_3$$

5 An example

In this section, an example is presented to illustrate the proposed approach in Sections 3 and 4. Consider the λ-free PN $G=(N,M_0,\Sigma,\mathcal{L})$ shown in Fig. 6. Its observable transition set is $T_o=\{t_1,t_2,t_3,t_5,t_6\}$, and the set of fault transitions is $T_f=\{t_f\}$. The labeling function $\mathcal{L}$ is defined as $\mathcal{L}(t_1)=a$, $\mathcal{L}(t_2)=c$, $\mathcal{L}(t_3)=\mathcal{L}(t_5)=d$, and $\mathcal{L}(t_6)=b$.

According to Fig. 6, the system contains a single fault transition t_f. Therefore, the fault-free subnet can be obtained by removing the transition t_f and the arcs connected to it. The resulting subnet is shown in Fig. 7.

By performing the synchronous product of the PNs shown in Figs. 6 and 7 using Algorithm 1, the corresponding verifier net is obtained, as illustrated in Fig. 8. Transitions t_3 and t_5 share the same label d and are therefore synchronized during the construction of the verifier net. Consequently, four transition pairs are generated, namely (t'_3,t_3), (t'_3,t_5), (t'_5,t_3), and (t'_5,t_5). Since the transition t_f and the arcs connected to it have been removed, the fault-related behavior is eliminated in the fault-free subnet. Accordingly, no synchronization involving t_f occurs during the construction of the verifier net.

According to Algorithm 2, the RG is shown in Fig. 9. The set of reachable markings is listed in Table 3, and the set of faulty nodes is given by $\mathcal{F}(V)=\{\widetilde{M}_2,\widetilde{M}_3\}$. It is observed that there exists a

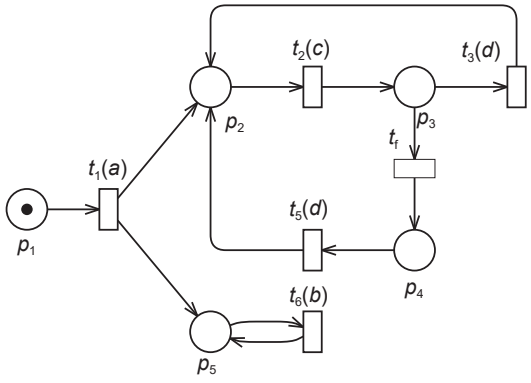


Fig. 6 A λ-free PN.

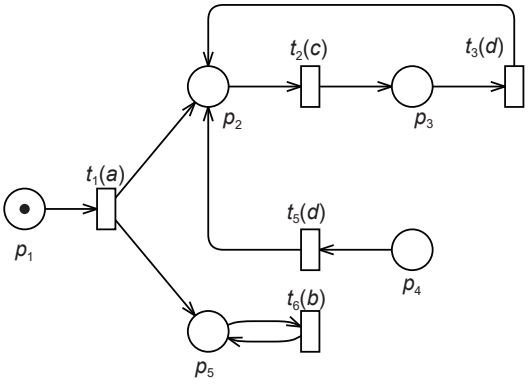


Fig. 7 The fault-free subnet of Fig. 6.

cycle that is entered after the occurrence of a fault transition starting from the initial node, namely,

$$\widetilde{M}_0 \xrightarrow{(t'_1,t_1)} \widetilde{M}_1 \xrightarrow{(t'_2,t_2)} \widetilde{M}_2 \xrightarrow{(\varepsilon,t_f)} \widetilde{M}_3 \xrightarrow{(t'_3,t_5)} \widetilde{M}_2$$

This forms a faulty path according to Definition 5. For clarity, it is shown in Fig. 10. By Proposition 3, the system is not diagnosable.

To obtain the left-hand side of Eq. (2), we apply Algorithm 3 to the faulty path. According to the relabeling option (O1), the transition pair (t'_3,t_5) adds $+v_{t_3}+v_{t_5}$ to the left-hand side of Eq. (2). According to relabeling options (O2) and (O3), the transition pairs (t'_1,t_1), (t'_2,t_2), and (ε,t_f) are not processed. Thus, the linear inequality corresponding to the transition pairs (x',x) of this faulty path can be written as Eq. (5)

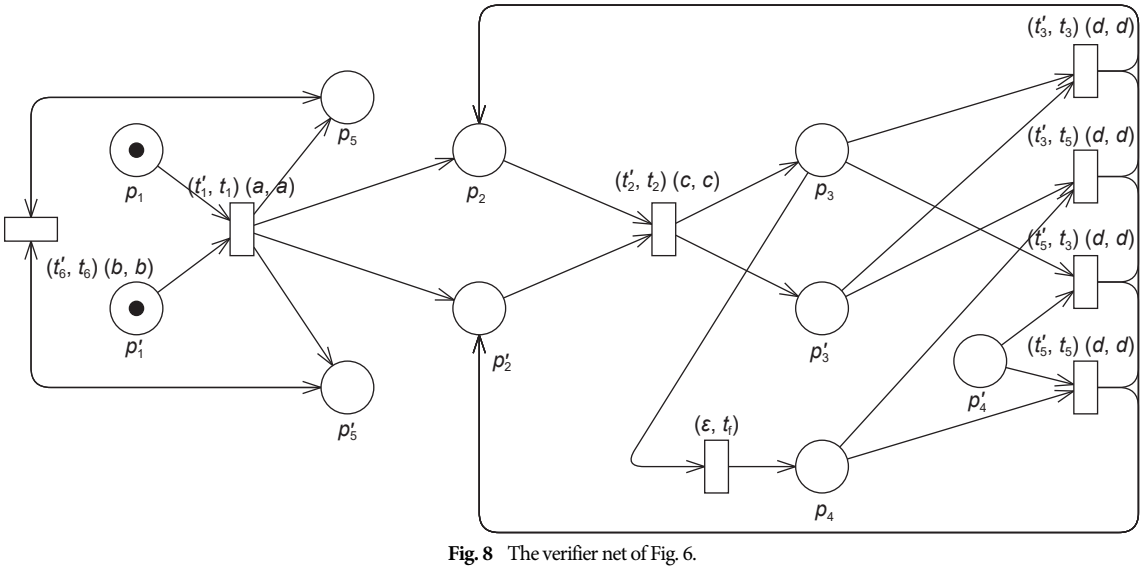


Fig. 8 The verifier net of Fig. 6.

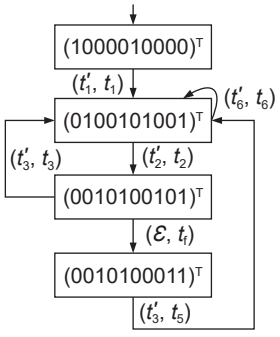


Fig. 9 The RG of Fig. 8.

Table 3 The reachable markings of Fig. 9

M	Marking
M_0	$(1\,0\,0\,0\,0\,1\,0\,0\,0\,0)^{\mathrm{T}}$
M_1	$(0\,1\,0\,0\,1\,0\,1\,0\,0\,1)^{\mathrm{T}}$
M_2	$(0\,0\,1\,0\,1\,0\,0\,1\,0\,1)^{\mathrm{T}}$
M_3	$(0\,0\,1\,0\,1\,0\,0\,0\,1\,1)^{\mathrm{T}}$

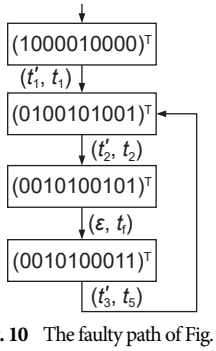


Fig. 10 The faulty path of Fig. 9.

$$v_{t_3} + v_{t_5} \geq 1 \tag{5}$$

To minimize the economic burden, we solve the ILP problem in Eq. (6).

$$\begin{cases} \min \quad \sum\limits_{t \in T_N} c_t v_t \\ v_{t_3} + v_{t_5} \geq 1 \end{cases} \tag{6}$$

Assume that $c_t = 1$ for all transitions $t \in T_o$. By solving the ILP problem in Eq. (6), the optimal solution is obtained by relabeling transition t_3. If we relabel

$$\mathcal{L}_N(t_3) = t_3$$

the system becomes diagnosable.

It can be seen that another optimal solution is given by

$$\mathcal{L}_N(t_5) = t_5$$

6 Conclusions

For λ-free PN systems, this paper presents a method for diagnosability analysis and enforcement. We first construct a verifier net and its RG, and determine system diagnosability by analyzing paths and cycles in the graph. For a nondiagnosable system, all faulty paths violating diagnosability are identified, and relabeling rules are applied to the transitions involved. An ILP model is then solved to obtain the optimal labeling function, ensuring system diagnosability. Future work will extend this approach to unbounded PN models with unobservable transitions, aiming to develop more general methods for diagnosability analysis and enforcement of DESs.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (Grant Nos. 62373132 and 62373234), the Natural Science Foundation of Hebei Province (Grant No. F2025201023), the S&T Program of Shijiazhuang (Grant No. 241791367A), the Excellent Youth Research Innovation Team of Hebei University (Grant No. QNTD202411), and the Interdisciplinary Research Program of Hebei University (Grant No. DXK202409).

Declaration of competing interest

The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Cassandras, C. G.; Lafortune, S. *Introduction to Discrete Event Systems*. New York, USA: Springer USA, 2008.

[2] Bates, I. W.; Karimoddini, A.; Karimadini, M. A learning-based approach for diagnosis and diagnosability of unknown discrete event systems. *IEEE Transactions on Neural Networks and Learning Systems* Vol. 35, No. 4, 5421–5434, 2024.

[3] Lin, F. Diagnosability of discrete event systems and its applications. *Discrete Event Dynamic Systems* Vol. 4, No. 2, 197–212, 1994.

[4] Sampath, M.; Sengupta, R.; Lafortune, S.; Sinnamohideen, K.; Teneketzis, D. Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control* Vol. 40, No. 9, 1555–1575, 1995.

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

[6] Basilio, J. C.; Silva, G. M. O. A diagnoser-based approach to diagnosis and diagnosability of repeated faults of discrete-event systems. *IEEE Transactions on Automatic Control* Vol. 70, No. 8, 5491–5498, 2025.

[7] Gougam, H. E.; Pencolé, Y.; Subias, A. Diagnosability analysis of patterns on bounded labeled prioritized Petri nets. *Discrete Event Dynamic Systems* Vol. 27, No. 1, 143–180, 2017.

[8] Cabasino, M. P.; Giua, A.; Seatzu, C. Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica* Vol. 46, No. 9, 1531–1539, 2010.

[9] Madalinski, A.; Nouioua, F.; Dague, P. Diagnosability verification with Petri net unfoldings. *International Journal of Knowledge-Based and Intelligent Engineering Systems* Vol. 14, No. 2, 49–55, 2010.

[10] Cabasino, M. P.; Giua, A.; Lafortune, S.; Seatzu, C. A new approach for diagnosability analysis of Petri nets using verifier nets. *IEEE Transactions on Automatic Control* Vol. 57, No. 12, 3104–3117, 2012.

[11] Basile, F.; Chiacchio, P.; De Tommasi, G. An efficient approach for online diagnosis of discrete event systems. *IEEE Transactions on Automatic Control* Vol. 54, No. 4, 748–759, 2009.

[12] Dotoli, M.; Fanti, M. P.; Mangini, A. M.; Ukovich, W. On-line fault detection in discrete event systems by Petri nets and integer linear programming. *Automatica* Vol. 45, No. 11, 2665–2672, 2009.

[13] Cabasino, M. P.; Giua, A.; Seatzu, C. Diagnosability of bounded Petri nets. In: Proceedings of the 48th IEEE Conference on Decision and

Control Held Jointly with the 28th Chinese Control Conference, 1254–1260, 2009.

[14] Cabasino, M. P.; Giua, A.; Seatzu, C. Diagnosability of discrete-event systems using labeled Petri nets. *IEEE Transactions on Automation Science and Engineering* Vol. 11, No. 1, 144–153, 2014.

[15] Ma, Z.; Tong, Y.; Li, Z.; Giua, A. Basis marking representation of Petri net reachability spaces and its application to the reachability problem. *IEEE Transactions on Automatic Control* Vol. 62, No. 3, 1078–1093, 2017.

[16] Jiroveanu, G.; Boel, R. K. The diagnosability of Petri net models using minimal explanations. *IEEE Transactions on Automatic Control* Vol. 55, No. 7, 1663–1668, 2010.

[17] Lefebvre, D. On-line fault diagnosis with partially observed Petri nets. *IEEE Transactions on Automatic Control* Vol. 59, No. 7, 1919–1924, 2014.

[18] Lefebvre, D. Fault diagnosis and prognosis with partially observed Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* Vol. 44, No. 10, 1413–1424, 2014.

[19] Paiva, P. R. R.; Carvalho, L. K.; Basilio, J. C. Diagnosability and online diagnosis of discrete-event systems modeled by acyclic labeled Petri nets. In: Proceedings of the 20th IFAC World Congress, 13630–13635, 2017.

[20] Miao, S.; Jančar, P.; Komenda, J.; Masopust, T.; Lai, A. Diagnosability verification for automata and Petri nets: Can we do better? *Automatica* Vol. 189, Article No. 113011, 2026.

[21] Cabasino, M. P.; Lafortune, S.; Seatzu, C. Optimal sensor selection for ensuring diagnosability in labeled Petri nets. *Automatica* Vol. 49, No. 8, 2373–2383, 2013.

[22] Ran, N.; Giua, A.; Seatzu, C. Enforcement of diagnosability in labeled Petri nets via optimal sensor selection. *IEEE Transactions on Automatic Control* Vol. 64, No. 7, 2997–3004, 2019.

[23] Hu, S.; Li, Z.; Wisniewski, R. Optimal sensor selection for diagnosability enforcement in labeled Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* Vol. 54, No. 5, 2965–2977, 2024.

[24] Hu, S.; Duan, W.; Liu, D.; Li, Z. Supervisory control for active diagnosis in labeled Petri nets under observation and control delays. *IEEE Transactions on Automatic Control* Vol. 71, No. 7, 4403–4418, 2026.

[25] Giua, A.; Corona, D.; Seatzu, C. State estimation of λ-free labeled Petri nets with contact-free nondeterministic transitions. *Discrete Event Dynamic Systems* Vol. 15, No. 1, 85–108, 2005.

[26] Cabasino, M. P.; Giua, A.; Seatzu, C. Identification of Petri nets from knowledge of their language. *Discrete Event Dynamic Systems* Vol. 17, No. 4, 447–474, 2007.

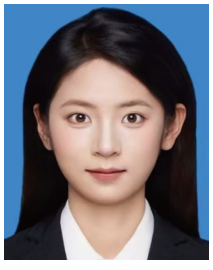
[27] Freitas, B. I. de; Toguyeni, A.; Basilio, J. C. Online representation-based state estimation of lambda-free labeled Petri nets. In: Proceedings of the 12th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, 72–77, 2024.

[28] Peterson, J. L. *Petri Net Theory and the Modeling of Systems*. New Jersey, USA: Prentice Hall PTR, 1981.

[29] Lefebvre, D.; Leclercq, E. Diagnosability of Petri nets with observation graphs. *Discrete Event Dynamic Systems* Vol. 26, No. 3, 539–559, 2016.

[30] Ran, N.; Wang, S.; Su, H.; Wang, C. Fault diagnosis for discrete event systems modeled by bounded Petri nets. *Asian Journal of Control* Vol. 19, No. 4, 1532–1541, 2017.

Ning Ran received his Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2017. He is currently a professor at Hebei University. His current research interests include discrete event systems and fault diagnosis.

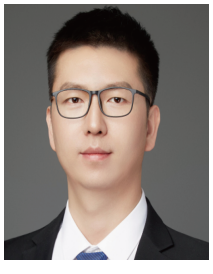


Xinyue Gu is a master student in the College of Electronic and Information Engineering, Hebei University. She received her B.S. degree from Langfang Normal University in 2024. Her research interests include discrete event systems and fault diagnosis.

Haijun Jia received his M.S. degree from Jilin University, Changchun, China, in 2015. He currently serves as a senior engineer at Jingneng Xilingol Energy Co., Ltd. His current research interests include comprehensive energy utilization of electricity.

Ruicai Si received his M.S. degree in control theory and control engineering from North China Electric Power University, Baoding, China, in 2011. He currently serves as a senior engineer at Jilin Electric Power Research Institute Co., Ltd. His current research interests include primary frequency control technology and AGC technology.

Zijian Dong received his Ph.D. degree in control theory and control engineering from North China Electric Power University, Baoding, China, in 2015. He is currently an associate professor at North China Electric Power University. His current research interests include advanced control algorithms and control optimization for thermal power units.



Zhou He (Senior Member, IEEE) received his B.S. degree in mechanical engineering from Shaanxi University of Science and Technology, Xi'an, China, in 2012, and the joint Ph.D. degree in mechatronic engineering and automatique from Xidian University, Xi'an, and Aix-Marseille University, Marseille, France, in 2017. He was a visiting researcher with the University of Cagliari, Cagliari, Italy, and Aix-Marseille University from May 2014 to August

2014 and from August 2015 to July 2016, respectively. In 2017, he joined Shaanxi University of Science and Technology, where he is currently a professor and a doctoral supervisor at the School of Electrical and Control Engineering. He has published more than 25 articles in journals, such as *Automatica, IEEE Transactions on Automatic Control, IEEE Transactions on Automation Science and Engineering*, and *IEEE Transactions on Control Systems Technology*. His research interests include Petri nets, discrete event systems, intelligent control, and optimization with applications to manufacturing systems and robotics.

Jinyuan Hao received her B.A. degree in radio and television journalism from Hebei University, China, in 2013, and her M.A. degree in journalism and communication from the University of New South Wales, Sydney, Australia, in 2015. She is currently a lecturer at Hebei University. Her current research interests include discrete event systems and fault diagnosis.