Online Fault Diagnosis of Discrete Event Systems Modeled With Labeled Petri Nets Using an Overall Fault Status

Guanghui Zhu, Lei Feng, Zhiwu Li, Naiqi Wu

Online Fault Diagnosis of Discrete Event Systems Modeled With Labeled Petri Nets Using an Overall Fault Status

Guanghui Zhu, *Student Member, IEEE,* Lei Feng, Zhiwu Li, *Fellow, IEEE*, Naiqi Wu, *Senior Member, IEEE*

***Abstract*—In this paper we present a fault diagnosis approach using labeled Petri nets, where the faults are modeled by unobservable transitions and the unobservable subnet is acyclic. In contrast to detecting the individual faults separately, a new specification called *an overall fault status* is introduced, which indicates the occurrence of faults from a global system perspective. Due to the introduction of the overall fault status, a more precise and informative diagnosis result can be provided and in some cases, the occurrence of some faults in a system can be detected before the actual faults are isolated, i.e., we are certain about the occurrence of faults but which faults have not been ascertained. An integer linear programming (ILP) problem is built according to the observed word. We prove that all transition sequences determined by solutions to the ILP problem constitute the set of sequences consistent with the observed word. By specifying different objective functions to the ILP problem, the diagnosis results of each individual fault and the overall fault status can be obtained. An online diagnosis algorithm is developed to implement the proposed diagnosis process, which reports the diagnosis results after the occurrence of every observable event.**

***Index Terms*—Fault diagnosis, discrete event system, Petri net, integer linear programming, overall fault status.**

I. Introduction

A. Position of the paper

With the development of contemporary information technology, the man-made mechanical and electronic systems that can be characterized as discrete event systems (DES) are becoming more and more complicated. To ensure their stable and correct operations, fault diagnosis has been an active research area in recent decades. A fault is a kind of event that causes a deviation in system's behavior such that the performance or throughput of the system is degraded. Fault diagnosis aims to detect and isolate a fault when it occurs such that it can be fixed and the system can recover from it.

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 61873342 and 61703321, in part by the Science and Technology Development Fund, MSAR, under Grant Nos. 122/2017/A3 and 106/2016/A3, and in part by the 2017 Sino-French Cai Yuanpei Program. (*Corresponding author: Zhiwu Li.*)

G. Zhu is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, and with Aix Marseille University, Universite de Toulon, CNRS, LIS, Marseille, France (e-mail: zhuguanghui86@gmail.com).

L. Feng is with the Department of Machine Design, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (e-mail: lfeng@kth.se).

Z. Li is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China, and also with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau (e-mail: zhwli@xidian.edu.cn).

N. Wu is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau (e-mail: nqwu@must.edu.mo).

The diagnosis of discrete event systems was originally discussed in [1], [2] using an automaton model with faulty (unobservable) events, where a diagnoser, i.e., a deterministic finite automaton (DFA), is first built and then based on the observed sequence the diagnosis result can be directly obtained by checking the diagnoser. The authors also define the diagnosability of an automation model and provide a necessary and sufficient condition for diagnosability.

An alternative to automata for modeling DES is provided by Petri nets. Their structural properties offer a new perspective for supervisory control [3], [4], model identification [5], [6], performance optimization [7], and knowledge discovery [8], [9]. In addition, the *state equation* of a Petri net provides a linear algebraic technique to deal with the issue of state estimation [10]–[14], which is always more efficient than exhaustively enumerating the reachability graph. In particular, we in this paper deal with the fault diagnosis issue using Petri nets.

In the context of Petri nets, Prock [15] proposed a diagnosis approach for a nuclear power plant by monitoring the number of tokens residing in places associated with P-invariants in a Petri net. Wu and Hadjicostis [16] developed an algebraic approach for fault diagnosis, where both place and transition faults are defined. By introducing additional places into a net, a *redundant Petri net* can be obtained. The place and/or transition faults can be detected by inspecting the current marking of the redundant net. Ramírez-Treviño *et al.* [17] proposed a modeling methodology to build an interpreted Petri net (IPN) model of a system and then a fault detection algorithm based on the built IPN is provided. Benveniste *et al.* [18] reported a net unfolding approach to explore the diagnosis of asynchronous systems, which can be used in a distributed environment.

On the other hand, there is a deluge of studies that use Petri nets by explicitly modeling faults of a system as unobservable transitions [19], i.e, Petri nets, called *faulty Petri nets*, that contain not only regular but faulty behavior of a system. Genc *et al.* [20] originally extended the event-based diagnosis using automata to the case of faulty Petri nets. They construct a diagnoser, i.e., a labeled Petri net, according to the original net model of a system and a diagnosis result can be provided online when observing an event. The main drawback of this approach is the computational complexity due to the reachability analysis at each step. Giua and Seatzu [21] proposed a *basis-marking*-based approach. By means of basis markings and the corresponding *justifications*, the enumeration of paths in a reachability graph can be avoided. Cabasino *et al.* [22]–[24] extended this approach to the case of labeled Petri nets, i.e., nets where two or more transitions can share the same label. They develop a basis reachability graph (BRG), which can be built off-line and provides an efficient method for online diagnosis. In [25], Basile *et al.* defined a new type of marking with negative elements, called *g-marking*, and proposed an online diagnosis algorithm based on it. Integer linear programming (ILP) is a typical technique to deal with the issue of state estimation in a Petri net. An online diagnosis approach using ILP is reported in [26]. More specifically, the authors first build an integer linear programming model without objective function according to the observed transition sequence. Then, by assigning different objective functions to it, the occurrence of a fault can be detected. Ru *et al.* [27] addressed the issue of fault diagnosis using a partially observed Petri net (POPN), where a POPN is first converted into a labeled net and then an algorithm based on the reachability graph is proposed.

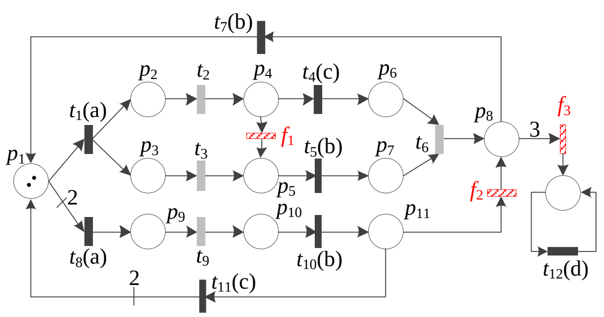

Fig. 1. A plant model producing bolts and nuts.

B. Motivation

In a faulty Petri net model, faults are explicitly modeled as unobservable transitions and the transition set T is divided into two disjoint subsets T_o and T_u with $T = T_o \cup T_u$, where T_o denotes the set of observable transitions and T_u the set of unobservable transitions. Transitions $t_1 \in T_o$ and $t_2 \in T_u$ are assigned a label from the event set E and an empty string ε, respectively. In general, the diagnosis result provided by approaches based on faulty Petri nets [12], [21]–[23], [25], [28] can be represented by a function $\Delta\colon E^* \times T_f \to \{0, 1, 2\}$, where E is the set of events associated with a faulty net, $T_f = \{f_1, \ldots, f_{n_f}\} \subseteq T_u$ is the set of n_f fault transitions, and 0, 1 and 2 denote that fault transition $f_i \in T_f$ does not occur for sure, f_i may occur and f_i occurs with certainty, respectively. For example, assuming that the observed sequence is ω, $\Delta(\omega, f_1) = 1$ indicates that f_1 may (possibly) occur till the observation of ω.

We next provide an example to show the motivation of introducing an *overall fault status*. Consider the net in Fig. 1 that models a plant producing bolts and nuts. This plant has two production lines: one (upper part) produces mini-size bolts and nuts and the other (lower part) produces medium-size ones. In Fig. 1, we have $E = \{a, b, c, d\}$, $T_u = \{t_2, t_3, t_6, t_9, f_1, f_2, f_3\}$, $T_f = \{f_1, f_2, f_3\}$, $T_a = \{t_1, t_8\}$, $T_b = \{t_5, t_7, t_{10}\}$, $T_c = \{t_4, t_{11}\}$, and $T_d = \{t_{12}\}$. The unobservable transitions are represented as gray bars (fault transitions f_1, f_2 and f_3 are colored red), and T_x denotes the set of transitions labeled x with $x \in \{a, b, c, d\}$. It is significant to design an algorithm to solve the following problem.

Problem 1. Consider the net shown in Fig. 1 and assume that the observed sequence is $\omega = abb$. Does the plant modeled by this net run normally till the observation of ω? (i.e., have some faults occurred in the plant?)

The existing approaches for fault diagnosis, such as those in [12], [21], [23], [25], [28], provide an ambiguous answer to Problem 1 only. The procedure in [23], [25], [28] to solve Problem 1 can be summarized as follows.

Step 1: Compute the diagnosis results of f_1, f_2 and f_3, respectively, and obtain that $\Delta(\omega, f_1) = 1$, $\Delta(\omega, f_2) = 1$, and $\Delta(\omega, f_3) = 0$.

Step 2: Because of the ambiguous diagnoses of f_1 and f_2, one cannot unambiguously determine the occurrence of faults and only an answer that the plant *may* run abnormally can be provided.

In fact, the exact answer is that the plant runs abnormally and one of f_1 and f_2 necessarily occurs. This situation can be explained by Fig. 2(a), where the symbol ε denotes an empty string. As a part of reachability graph of the net shown in Fig. 1, it shows all transition sequences consistent with ω. In Fig. 2(a), M_0 is the initial marking, and $M_{\omega 1}$ and $M_{\omega 2}$ are the markings reachable by firing the sequences consistent with $\omega = abb$. We observe that each path from M_0 to $M_{\omega 1}$ or $M_{\omega 2}$ either goes through f_1 or f_2, i.e., there is no path that passes none of faults. Fig. 2(a) can be condensed as Fig. 2(b) which shows all the paths more intuitively. By considering all possible evolutions of the plant consistent with the observation abb, there does not exist a possibility that no fault occurs since each path contains a fault transition. This implies that a fault must occur till the observation of abb and the plant does not run normally.

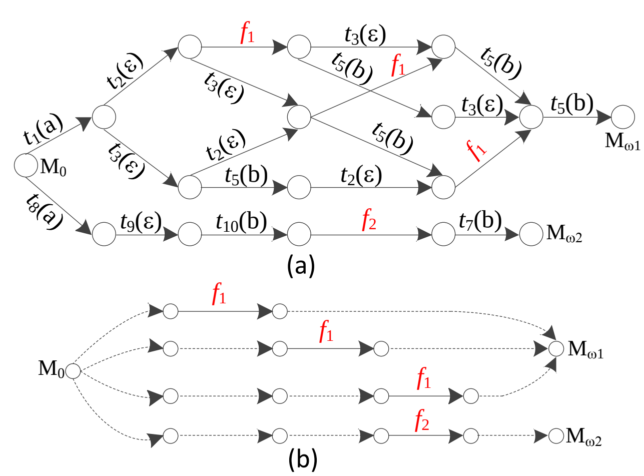

Fig. 2. (a) All paths consistent with abb and (b) an intuitive representation of (a).

This paper will provide a more precise and informative solution to Problem 1 by extending the existing diagnosis function. The new diagnosis function is defined as $\Delta\colon E^* \times T_f S \rightarrow \{0,1,2\}$, where $T_f S = T_f \cup \{\mathcal{F}\}$ and $\mathcal{F}$ stands for the *overall fault status*. In contrast to detecting the individual fault separately, the overall fault status indicates the occurrence of faults from a global system perspective. In particular, $\Delta(\omega, \mathcal{F}) = 0$ denotes that a system is running normally, $\Delta(\omega, \mathcal{F}) = 1$ some faults may occur in the system, and 2 some faults must have occurred. Note that, in some cases, the diagnosis of $\mathcal{F}$, i.e., $\Delta(\omega, \mathcal{F})$, can be directly inferred from $\Delta(\omega, f_i)$ with $i = 1 \ldots n_f$. However, in other cases, novel methodologies have to be proposed to compute $\Delta(\omega, \mathcal{F})$.

Due to the use of the overall fault status, in some cases, one can detect the occurrence of faults before the actual faults are isolated. Let us consider the net in Fig. 1 again. The diagnosis results for observed words ω's are listed in Table I, where $\Delta(\omega, \mathcal{F})$ denotes the diagnosis of the overall fault status. We observe that for $\omega = a$, both f_1 and f_2 do not occur. For $\omega = abb$, both f_1 and f_2 may occur, as indicated by $\Delta(abb, f_1) = 1$ and $\Delta(abb, f_2) = 1$. However, $\Delta(abb, \mathcal{F}) = 2$ implies that at least one fault must occur in the plant even if the diagnoses of f_1 and f_2 are ambiguous, i.e., we detect the occurrence of faults before the definite diagnosis of individual fault transition.

TABLE I
DIAGNOSIS RESULTS FOR OBSERVED WORD ω.

$\Delta(\omega, f)$ \ ω / f	a	ab	abb	$\ldots$	$abbacba$
f_1	0	1	1	$\ldots$	2
f_2	0	0	1	$\ldots$	0
f_3	0	0	0	$\ldots$	0
$\mathcal{F}$	0	1	2	$\ldots$	2

This is particularly meaningful for a system that needs to respond to failures in time. For example, consider an aircraft with five faults from f_1 to f_5. If we detect that $\Delta(\omega, f_1) = 1$, $\Delta(\omega, f_2) = 1$, $\Delta(\omega, f_i) = 0$ for $i = 3, \ldots, 5$, and $\Delta(\omega, \mathcal{F}) = 2$, some faults must occur in the aircraft though we cannot determine which faults occur at present. The captain of the aircraft can deal with this situation immediately before an unambiguous fault is reported.

However, for systems that can tolerate failures, we can wait for a longer observation of an event sequence in order to exactly find the faults with less cost. For example, some components of a system are hard to access and one should exactly identify the location of a fault before taking any corrective action that may involve component inspection and replacement [2]. If the occurrence of faults in a system is detected according to the overall fault status $\mathcal{F}$ and the exact faults cannot be currently ascertained, we can consider the following two ways to find the exact faults:

(1) Observe continuously the output of a net till a sufficiently long sequence is observed if the net is diagnosable (see Section V);
(2) Stop the real-world system and inspect all the faults f's with $\Delta(\omega, f) = 1$ one by one.

The first way provides exact fault locations such that they can be fixed in a short time. But a long wait may be needed to precisely identify the faults. For example, considering Fig. 1 and Table I, we detect the occurrence of faults ($\Delta(abb, \mathcal{F}) = 2$) but cannot determine that the fault is f_1 or f_2 when $\omega = abb$. If the plant continues to run and the observed word is $\omega = abbacba$, we then conclude that fault f_1 must happen ($\Delta(\omega, f_1) = 2$) and f_2 not ($\Delta(\omega, f_2) = 0$). On the other hand, for the second way, one has to stop (part of) the plant and further test it to find the faulty components. This will possibly reduce the throughput of the plant and increase the cost of isolating faults.

When performing diagnosis in a system, we will first try to detect any abnormal behavior of the system based on the observation. If the behavior is found to be abnormal, we will refine the diagnosis by using new observations and possibly further testing the system until the faulty components are found [29]. In this paper, a framework to isolate and fix faults in a system is proposed based on the overall fault status $\mathcal{F}$. This framework is appropriate for faults that cause significant changes in the system state but do not bring the system to a halt. In a real-world system, the overall fault status $\mathcal{F}$ can be viewed as a fault indicator light that has three colors of green ($\Delta(\omega, \mathcal{F}) = 0$), yellow ($\Delta(\omega, \mathcal{F}) = 1$), and red ($\Delta(\omega, \mathcal{F}) = 2$). If the light is green, it implies that the system is running normally and no fault is detected. Whereas a red light indicates that one or more faults have occurred and then one can check $\Delta(\omega, f_i)$ with $i = 1, \ldots, n_f$ to localize and fix the faults. Graphically, the diagnosis flow can be described by Fig. 3.

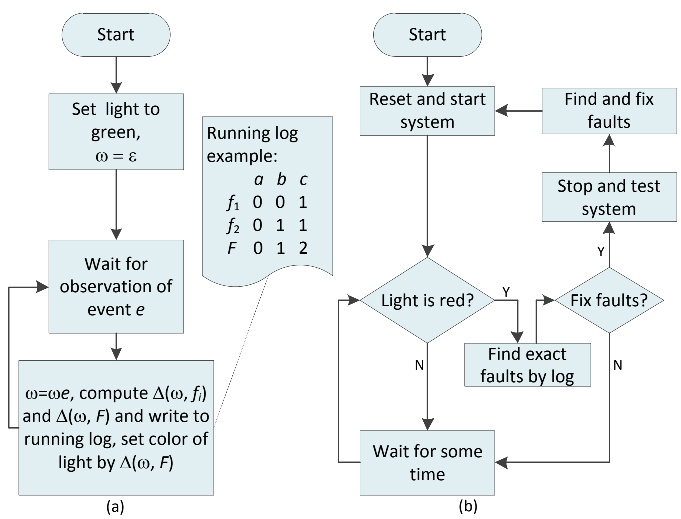

Fig. 3. (a) Diagnosis process and (b) supervision process

As shown in Fig. 3, there are two processes in the system: diagnosis process and supervision process. The diagnosis process makes a diagnosis when observing an event and enters the diagnosis result into a log file (see Fig. 3(a)). Note that a plant may continue to run and record the diagnosis result even if some faults have occurred. The supervision process monitors the operation of the plant and decides whether to stop it to fix faults when the indicator light is red (see Fig. 3(b)).

C. Contribution

In this paper, we extend the approach in [26] to explore an improved diagnosis algorithm based on labeled Petri nets and the overall fault status. As done in the literature [22], [25], the approach in [26] can only be used for a special class of labeled Petri nets, i.e., nets in which each observable transition is assigned a unique label. We first extend the approach in [26] to the case of a labeled Petri net with an arbitrary labeling function, i.e., two or more transitions can share the same label, and then propose some new theoretical results to compute $\Delta(\omega, \mathcal{F})$. An online diagnosis approach is presented, which implements the diagnosis process described in Fig. 3(a).

Note that Fanti *et al.* [30] also extend the approach in [26] to the case of a labeled Petri net. They first build an ILP problem based on a transition sequence $\sigma_o \in T_o^*$ using a special class of labeled Petri net where an observable transition is assigned a unique label. Then, for an observed word $\omega \in E^*$, they exhaustively enumerate all possible sequences σ_o's with $\sigma_o \in T_o^*$ whose projections on E are ω and, for each sequence σ_o, solve a number of ILP problems built from it to perform online diagnosis. For an observed word ω, there may exist a lot of transition sequences σ_o's whose projection on E are ω. Thus, a large number of ILP problems may need to be solved according to the approach in [30], which is infeasible in general due to prohibitive computational cost.

Different from the work in [30], we in this paper extend and improve the approach in [26] from a different perspective. Specifically, our extension is completely based on labeled Petri nets and an ILP problem is constructed based on the observed word ω not a transition sequence $\sigma_o \in T_o^*$. Compared with the approach in [30], the number of times of solving ILP problems of our approach is usually considerably small when perform diagnosis for an observed word (see Section VI for an example).

The main contributions of the paper can be summarized as follows.

(1) We introduce a new diagnosis specification called an overall fault status and extend the diagnosis function such that a more precise and informative diagnosis result can be provided.
(2) We extend and improve the approach in [26] to the case of general labeled Petri nets, i.e., nets where two or more transitions can share the same label. Different from the extension in [30], we do not need to enumerate all possible sequences of observable transitions consistent with an observed word $\omega \in E^*$ and just solve an ILP problem built according to ω to perform diagnosis.
(3) In contrast to the work in [30], our approach is usually more efficient, demonstrated by extensive experimental studies (see Section VI).

This paper is organized in seven sections. We in Section I review the related literature and show the motivation of introducing the overall fault status. The basic definitions and preliminaries on Petri nets are recalled in Section II. Section III defines the problem on which this paper focuses. In Section IV, a solution based on integer linear programming is proposed. Section V discusses the relationship between the diagnosability and the overall fault status. We present an example to compare the proposed approach with the one in [30] in Section VI. Finally, we conclude the paper in Section VII.

II. Preliminaries

This section recalls the Petri net formalism and some preliminary results used throughout the paper. The readers can refer to [31] and [32] for more details on Petri nets. We denote by $\mathbb{N}$ the set of non-negative integers.

A. Basics of Petri nets

A Petri net is a four-tuple $N = (P, T, Pre, Post)$, where $P = \{p_1, \ldots, p_m\}$ is a set of m places, $T = \{t_1, \ldots, t_n\}$ is a set of n transitions with $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$, $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the *pre-* and *post-incidence* matrices, respectively, which specify the structure of the net. Graphically, places and transitions are represented by circles and bars, respectively. For each arc with weight γ from place p (transition t) to transition t (place p), it holds $Pre(p,t) = \gamma$ ($Post(p,t) = \gamma$). The other elements of Pre and $Post$ are 0. The incidence matrix of a net is denoted by $C = Post - Pre$. A Petri net is said to be *acyclic* if there is no directed cycle.

For a transition $t \in T$, its *preset* is defined as ${}^\bullet t = \{p \in P \mid Pre(p,t) > 0\}$, and its *postset* is defined as $t^\bullet = \{p \in P \mid Post(p,t) > 0\}$. A transition t is said to be a *source transition* if ${}^\bullet t = \emptyset$.

A marking of a Petri net is a vector $M : P \to \mathbb{N}$, and $M(p)$ indicates the number of tokens, pictorially denoted by black dots, in place p. We use $x_1 p_1 + \cdots + x_m p_m$ to denote the marking $[x_1, \ldots, x_m]^T$ for economy of space. A net system $\langle N, M_0 \rangle$ is a Petri net N with an initial marking M_0.

A transition $t \in T$ is *enabled* at marking M if for all $p \in {}^\bullet t, M(p) \geq Pre(p,t)$, which is denoted by $M[t\rangle$. An enabled transition t at M can fire yielding a new marking M' such that $M' = M + C(\cdot, t)$, which is denoted by $M[t\rangle M'$.

For a transition sequence $\sigma \in T^*$ and a marking M, $M[\sigma\rangle$ denotes that σ is enabled at marking M and $M[\sigma\rangle M'$ denotes that a new marking M' is *reachable* from M after firing σ. The set of all the markings reachable from M_0 is denoted by $R(N, M_0)$, called the *reachability set* of a Petri net. The set of transition sequences enabled at the initial marking M_0 is defined as

$$\mathcal{L}(N, M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}$$

which is called the *language* of Petri net system $\langle N, M_0 \rangle$.

We define a function $\pi \colon T^* \to \mathbb{N}^n$ that maps a transition sequence $\sigma \in T^*$ to n-dimensional column vector $y = \pi(\sigma)$, called *firing vector*, such that $y(t) = k$ if transition t appears k times in σ. Write $t \in \sigma$ to denote that t is contained in σ.

Given $M_0[\sigma\rangle M$, we have

$$M = M_0 + C \cdot \pi(\sigma). \tag{1}$$

Eq. (1), called the *state equation*, shows that there exists a non-negative integer vector y such that $M = M_0 + C \cdot y$ if M is reachable from M_0, which is a necessary but not sufficient condition for the reachability of marking M from M_0. However, for an acyclic net, it is necessary and sufficient, as verified by the following result.

Theorem 1. [31] Let $\langle N, M_0\rangle$ be an acyclic Petri net. A marking $M \geq 0$ is reachable from M_0 if and only if (iff) there exists a non-negative integer vector y satisfying $M = M_0 + C \cdot y$.

B. Labeled Petri net

Given a Petri net $N = (P, T, Pre, Post)$ and the event set E, a labeling function $\lambda\colon T \to E \cup \{\varepsilon\}$ assigns to each transition $t \in T$ either a symbol from the event set E or an empty string ε. In the case of no confusion, a labeled Petri net in this paper refers to a net with an arbitrary labeling function, i.e., two or more transitions can share the same label. A Petri net system $\langle N, M_0\rangle$ with a labeling function $\lambda : T \to E \cup \{\varepsilon\}$ is called a *labeled Petri net system*, denoted by $\langle N, M_0, E, \lambda\rangle$.

A transition t is said to be *unobservable* or *silent* if it is associated with the label ε, i.e., $\lambda(t) = \varepsilon$. The set of unobservable transitions is denoted by $T_u = \{t \in T \mid \lambda(t) = \varepsilon\}$ with cardinality n_u. All other transitions whose labels come from event set E constitute the set of *observable* transitions $T_o = \{t \in T \mid \lambda(t) \neq \varepsilon\}$ with cardinality n_o. Thus, set T is divided into two disjoint subsets T_o and T_u with $T = T_o \cup T_u$. For a faulty Petri net, the faults are always modeled by unobservable transitions and accordingly the set of unobservable transitions is also divided into two disjoint subsets T_{reg} and T_f, i.e., $T_u = T_{reg} \cup T_f$, where T_{reg} denotes the set of regular unobservable transitions and T_f the set of n_f fault transitions. We use $T_e = \{t \in T \mid \lambda(t) = e\}$ to represent the set of transitions with the same label $e \in E$.

Analogous to the definition of function π, for each sequence $\sigma_o \in T_o^*$, we define a function $\pi_o\colon T_o^* \to \mathbb{N}^{n_o}$ such that $y = \pi_o(\sigma_o)$ and $y(t) = k$ if $t \in T_o$ appears k times in σ_o. Similarly, the function $\pi_u\colon T_u^* \to \mathbb{N}^{n_u}$ associates a sequence $\sigma_u \in T_u^*$ with an n_u-dimensional vector $\pi_u(\sigma_u)$.

We extend the definition of labeling function λ to a sequence $\sigma t \in T^*$ such that $\lambda(\sigma t) = \lambda(\sigma)\lambda(t)$, i.e., for a sequence $\sigma \in T^*$, $\omega = \lambda(\sigma)$ is an *observed word* composed of the labels of observable transitions contained in σ. The set of observed words in a labeled Petri net system $\langle N, M_0, E, \lambda\rangle$ is defined as

$$\mathcal{L}^E(N, M_0) = \{\omega \in E^* \mid \sigma \in \mathcal{L}(N, M_0), \omega = \lambda(\sigma)\}.$$

Given a transition sequence σ, we use $\overleftarrow{\sigma}$ to denote the last transition in σ. For example, if $\sigma = t_1t_2t_3$, we have $\overleftarrow{\sigma} = t_3$. For an observed word $\omega \in \mathcal{L}^E(N, M_0)$, the set

$$\mathcal{C}(\omega) = \{\sigma \in T^* \mid \sigma \in \mathcal{L}(N, M_0), \lambda(\sigma) = \omega\}$$

denotes all sequences that are consistent with ω and

$$\overleftarrow{\mathcal{C}}(\omega) = \{\sigma \in T^* \mid \sigma \in \mathcal{C}(\omega), \overleftarrow{\sigma} \in T_o\}$$

stands for the consistent transition sequences whose last transitions are observable, i.e., the sequences with observable tails. Accordingly, the set of markings that are reachable by firing the consistent transition sequences with observable tails is defined as

$$\overleftarrow{\mathcal{D}}(\omega) = \{M \in \mathbb{N}^m \mid \sigma \in \overleftarrow{\mathcal{C}}(\omega), M_0[\sigma\rangle M\}.$$

Example 1. Consider the net shown in Fig. 1 and a part of its reachability graph shown in Fig. 2. If the observed word is $\omega = a$, then it is readily verified that $\mathcal{C}(\omega) = \{t_1, t_1t_2, t_1t_2f_1, t_1t_2f_1t_3, t_1t_2t_3, t_1t_2t_3f_1, t_1t_3, t_1t_3t_2, t_1t_3t_2f_1, t_8, t_8t_9\}$ and $\overleftarrow{\mathcal{C}}(\omega) = \{t_1, t_8\}$. On the other hand, for $\omega = abb$, we have $\overleftarrow{\mathcal{D}}(\omega) = \{M_{\omega_1}, M_{\omega_2}\}$.

Definition 1. Given a net $N = (P, T, Pre, Post)$ and a subset of transitions $T' \subseteq T$, the T'-*induced* subnet of N is a Petri net $N' = (P, T', Pre', Post')$, where Pre' and $Post'$ are the restrictions of Pre and $Post$ to $P \times T'$, respectively, i.e., the net N' is obtained by removing all transitions in $T \setminus T'$ from N.

Given a net $N = (P, T, Pre, Post)$ with $T = T_o \cup T_u$, according to Definition 1, one can readily obtain from N the *observable subnet* $N_o = (P, T_o, Pre_o, Post_o)$ and the *unobservable subnet* $N_u = (P, T_u, Pre_u, Post_u)$. Moreover, the incidence matrices of nets N_o and N_u are denoted by $C_o = Post_o - Pre_o$ and $C_u = Post_u - Pre_u$, respectively. Note that, as in the literature [22], [25], [26], this paper assumes that the unobservable subnet is acyclic.

III. Problem Statement

Fault diagnosis consists in determining if faults have occurred in a system according to the observed system output, such as the observed word ω in a Petri net. As done in the literature, we can design a diagnoser, i.e., a diagnosis function, to show the diagnosis result. For a labeled Petri net with event set E, a diagnoser is a function $\Delta\colon E^* \times T_f \to \{0, 1, 2\}$ such that a fault $f \in T_f$ does not occur if $\Delta(\omega, f) = 0$, may occur if 1, and necessarily occurs if 2. Next, we provide the formal definition of diagnosis function Δ.

Definition 2. Let $\langle N, M_0, E, \lambda\rangle$ be a labeled Petri net system. The diagnosis function $\Delta\colon E^* \times T_f \to \{0, 1, 2\}$ associates an observed word $\omega \in \mathcal{L}^E(N, M_0)$ and a fault $f \in T_f$ with a diagnosis state such that

(1) $\Delta(\omega, f) = 0$ if for all $\sigma \in \overleftarrow{\mathcal{C}}(\omega)$, $f \notin \sigma$, i.e., each path consistent with ω in the reachability graph does not pass fault transition f.
(2) $\Delta(\omega, f) = 1$ if there exist $\sigma_1, \sigma_2 \in \overleftarrow{\mathcal{C}}(\omega)$ such that $f \notin \sigma_1$ and $f \in \sigma_2$, i.e., there exist two paths, one of which passes f and the other does not.
(3) $\Delta(\omega, f) = 2$ if for all $\sigma \in \overleftarrow{\mathcal{C}}(\omega)$, $f \in \sigma$.

Note that some existing studies [21], [22] additionally consider the possible fault transitions after the last observed event of ω when performing diagnosis, i.e., the definition of Δ is based on $\mathcal{C}(\omega)$ not $\overleftarrow{\mathcal{C}}(\omega)$. However, we in this paper do not adopt this setting and detect only the fault transitions occurring before the last event of ω in order to be consistent with the seminal work in [1].

From Definition 2, we observe that the diagnoser Δ only concerns if an individual fault $f \in T_f$ has occurred during the system evolution and there is no information to indicate the overall system fault state, i.e., the diagnosis results of all fault transitions cannot completely describe the fault state of the system. To overcome this, we introduce the notion of the overall fault status $\mathcal{F}$ in Subsection I-B.

A *diagnostic Petri net system* $\langle N, M_0, E, \lambda, \mathcal{F}\rangle$ is a labeled Petri net system $\langle N, M_0, E, \lambda\rangle$ with an overall fault status $\mathcal{F}$. For $\langle N, M_0, E, \lambda, \mathcal{F}\rangle$, the diagnosis function is extended as $\Delta\colon E^* \times T_fS \to \{0,1,2\}$, where $T_fS = T_f \cup \{\mathcal{F}\}$. The diagnosis of $\mathcal{F}$ is defined as follows.

Definition 3. Let $\langle N, M_0, E, \lambda, \mathcal{F}\rangle$ be a diagnostic Petri net system. For an observed word $\omega \in \mathcal{L}^E(N, M_0)$, the diagnosis of $\mathcal{F}$ is represented by a diagnosis function $\Delta\colon E^* \times T_fS \to \{0,1,2\}$ such that

(1) $\Delta(\omega, \mathcal{F}) = 0$ if for all $\sigma \in \overleftarrow{\mathcal{C}}(\omega)$ and for all $f \in T_f$, $f \notin \sigma$.
(2) $\Delta(\omega, \mathcal{F}) = 1$ if there exist $\sigma_1, \sigma_2 \in \overleftarrow{\mathcal{C}}(\omega)$ such that (i) for all $f \in T_f$, $f \notin \sigma_1$ and (ii) there exists $f \in T_f$, $f \in \sigma_2$.
(3) $\Delta(\omega, \mathcal{F}) = 2$ if for all $\sigma \in \overleftarrow{\mathcal{C}}(\omega)$, there exists $f \in T_f$ such that $f \in \sigma$.

In some cases, the value of $\Delta(\omega, \mathcal{F})$ can be directly deduced from $\Delta(\omega, f_i)$. We provide the following proposition to formalize this situation. However, when there exist two or more f's such that $\Delta(\omega, f) = 1$, we have to figure out a new method to compute $\Delta(\omega, \mathcal{F})$.

Proposition 1. Given a diagnostic Petri net system $\langle N, M_0, E, \lambda, \mathcal{F}\rangle$ and an observed word $\omega \in \mathcal{L}^E(N, M_0)$, the following statements hold:

(1) $\Delta(\omega, \mathcal{F}) = 0$ if for all $f \in T_f$, $\Delta(\omega, f) = 0$,
(2) $\Delta(\omega, \mathcal{F}) = 2$ if there exists $f \in T_f$ such that $\Delta(\omega, f) = 2$,
(3) $\Delta(\omega, \mathcal{F}) = 1$ if there exists only one $f \in T_f$ such that $\Delta(\omega, f) = 1$ and $\Delta(\omega, f') = 0$ for other faults $f' \in T_f$.

Proof. The conclusions (1) and (2) can be directly derived from Definitions 2 and 3. For (3), since there is one $f \in T_f$ such that $\Delta(\omega, f) = 1$, there exist two paths $\sigma_1, \sigma_2 \in \overleftarrow{\mathcal{C}}(\omega)$ satisfying $f \in \sigma_1$ and $f \notin \sigma_2$. Moreover, for any other fault transition f', it holds $\Delta(\omega, f') = 0$. Thus, we have $f \notin \sigma_2$ for each $f \in T_f$, i.e., $\Delta(\omega, \mathcal{F}) = 1$. □

The conclusion (3) in Proposition 1 provides a straightforward method to compute $\Delta(\omega, \mathcal{F})$ in the case that there is only one $f \in T_f$ satisfying $\Delta(\omega, f) = 1$ (for all $f' \in T_f \setminus \{f\}$, $\Delta(\omega, f') = 0$). However, for the case that there are two or more f's such that $\Delta(\omega, f) = 1$, Corollary 3 has to be employed to compute $\Delta(\omega, \mathcal{F})$. The Lines 14–19 of Algorithm 1 show the use of Proposition 1 when performing diagnosis of $\mathcal{F}$.

Next, we formally define the problem considered in this paper. In Section IV, an algorithm based on integer linear programming is provided to solve this problem.

Problem 2 (Online diagnosis)**.** Given a diagnostic Petri net system $\langle N, M_0, E, \lambda, \mathcal{F}\rangle$ with $T = T_o \cup T_u$ and $T_u = T_{reg} \cup T_f$, let the observed word ω be null initially. The diagnosis problem consists in computing $\Delta(\omega, f)$ and $\Delta(\omega, \mathcal{F})$ for each observed event e during the evolution of the system, where $\omega = \omega e$, $f \in T_f$, and $\mathcal{F}$ is the overall fault status.

By Definitions 2 and 3, we know that an intuitive method to solve Problem 2 is to analyze the reachability graph of a net. However, this is usually infeasible because of its huge size. In order to avoid analyzing the reachability graph, an integer linear programming technique is often used for a Petri net. We next present an ILP-based solution to Problem 2.

IV. ILP-BASED SOLUTION

In [26], an approach based on ILP is proposed for fault diagnosis using Petri nets in which each observable transition is associated with a unique label. In this paper, we introduce the overall fault status and extend this approach to the case of a labeled Petri net. First, we give a theorem that is the cornerstone of the proposed ILP-based approach.

Theorem 2. Given a labeled Petri net $\langle N, M_0, E, \lambda\rangle$ and an observed word $\omega = e_1e_2\ldots e_h \in \mathcal{L}^E(N, M_0)$, there exists a sequence $\sigma = \sigma_{u_1}t_{\alpha_1}\ldots\sigma_{u_h}t_{\alpha_h} \in \overleftarrow{\mathcal{C}}(\omega)$ if and only if there exist vectors y_i and binary variables $z_j^{e_i}$ with $i = 1,\ldots,h$ and $j = 1,\ldots,n_{e_i}$ satisfying the following equation

$$\begin{cases} M_0 + C_u \cdot \sum_{\gamma=1}^{i} y_\gamma + \sum_{\gamma=1}^{i-1}\sum_{\delta=1}^{n_{e_\gamma}} C(\cdot, t_\delta^{e_\gamma}) \cdot (1 - z_\delta^{e_\gamma}) \\ \qquad\qquad - Pre(\cdot, t_1^{e_i}) \geq -z_1^{e_i} \cdot K \\ \vdots \\ M_0 + C_u \cdot \sum_{\gamma=1}^{i} y_\gamma + \sum_{\gamma=1}^{i-1}\sum_{\delta=1}^{n_{e_\gamma}} C(\cdot, t_\delta^{e_\gamma}) \cdot (1 - z_\delta^{e_\gamma}) \\ \qquad\qquad - Pre(\cdot, t_{n_{e_i}}^{e_i}) \geq -z_{n_{e_i}}^{e_i} \cdot K \\ z_1^{e_i} + \ldots + z_{n_{e_i}}^{e_i} = n_{e_i} - 1 \\ T_{e_i} = \{t_1^{e_i}, \ldots, t_{n_{e_i}}^{e_i}\} \\ y_i \in \mathbb{N}^{n_u} \\ z_1^{e_i}, \ldots, z_{n_{e_i}}^{e_i} \in \{0,1\} \\ i = 1,\ldots,h, \end{cases} \tag{2}$$

where $h \in \mathbb{N}$ is the length of the observed word, n_{e_i} is the cardinality of set T_{e_i}, $\sigma_{u_i} \in T_u^*$, $t_{\alpha_i} \in T_{e_i}$, $y_i = \pi_u(\sigma_{u_i})$ is a firing vector, and K is a sufficiently large positive integer.

Proof. For $i = 1,\ldots,h$, Eq. (2) contains h groups of constraints.

(*if*) For $i = 1$, we obtain the first group of constraints shown as follows:

$$\begin{cases} M_0 + C_u \cdot y_1 - Pre(\cdot, t_1^{e_1}) \geq -z_1^{e_1} \cdot K \\ \vdots \\ M_0 + C_u \cdot y_1 - Pre(\cdot, t_{n_{e_1}}^{e_1}) \geq -z_{n_{e_1}}^{e_1} \cdot K \\ z_1^{e_1} + \ldots + z_{n_{e_1}}^{e_1} = n_{e_1} - 1 \end{cases}$$

Since $z_k^{e_1}$ with $k = 1,\ldots,n_{e_1}$ are binary variables, there must exist a variable with value 0 among them. Without loss of generality, we assume $z_1^{e_1} = 0$. Then we have $M_0 + C_u \cdot y_1 \geq Pre(\cdot, t_1^{e_1})$. Considering that the unobservable subnet is acyclic, there exists a transition sequence $\sigma_{u_1} \in T_u^*$ such that

$y_1 = \pi_u(\sigma_{u_1})$ and $M_0[\sigma_{u_1} t_1^{e_1}\rangle$. It is clear that $\sigma_{u_1} t_1^{e_1} \in \overleftarrow{\mathcal{C}}(e_1)$ holds.

For $i = 2$, we obtain the second group of constraints:

$$\begin{cases} M_0 + C_u \cdot (y_1 + y_2) + C(\cdot, t_1^{e_1})(1 - z_1^{e_1}) + \ldots \\ \qquad + C(\cdot, t_{n_{e_1}}^{e_1})(1 - z_{n_{e_1}}^{e_1}) - Pre(\cdot, t_1^{e_2}) \geq -z_1^{e_2} \cdot K \\ \vdots \\ M_0 + C_u \cdot (y_1 + y_2) + C(\cdot, t_1^{e_1})(1 - z_1^{e_1}) + \ldots \\ \qquad + C(\cdot, t_{n_{e_1}}^{e_1})(1 - z_{n_{e_1}}^{e_1}) - Pre(\cdot, t_{n_{e_2}}^{e_2}) \geq -z_{n_{e_2}}^{e_2} \cdot K \\ z_1^{e_2} + \ldots + z_{n_{e_2}}^{e_2} = n_{e_2} - 1 \end{cases}$$

We know that $z_1^{e_1} = 0$ and $z_k^{e_1} = 1$ with $k = 2, \ldots, n_{e_1}$. Moreover, analogous to the situation of $i = 1$, we assume $z_1^{e_2} = 0$. Thus, we have $M_0 + C_u \cdot y_1 + C(\cdot, t_1^{e_1}) + C_u \cdot y_2 \geq Pre(\cdot, t_1^{e_2})$. Since there exists a sequence σ_{u_1} satisfying $M_0[\sigma_{u_1} t_1^{e_1}\rangle M_1$ and $y_1 = \pi_u(\sigma_{u_1})$, $M_1 + C_u \cdot y_2 \geq Pre(\cdot, t_1^{e_2})$ holds. Being the unobservable subnet acyclic, there exists $\sigma_{u_2} \in T_u^*$ such that $M_1[\sigma_{u_2} t_1^{e_2}\rangle$ and $y_2 = \pi_u(\sigma_{u_2})$. Thus we conclude that there exists $\sigma = \sigma_{u_1} t_1^{e_1} \sigma_{u_2} t_1^{e_2}$ satisfying $M_0[\sigma\rangle$ and $\sigma \in \overleftarrow{\mathcal{C}}(e_1 e_2)$.

If the theorem is true for $i = k - 1$ with $2 \leq k \leq h$, we next prove that it holds for $i = k$. For $i = k$, without loss of generality, we assume $z_1^{e_j} = 0$ with $j = 1, \ldots, k$. Then, it holds

$$M_0 + C_u \cdot \sum_{\gamma=1}^{k} y_\gamma + \sum_{\gamma=1}^{k-1} C(\cdot, t_1^{e_\gamma}) \geq Pre(\cdot, t_1^{e_k}).$$

Moreover, we have already known that $y_i = \pi_u(\sigma_{u_i})$ with $i = 1, \ldots, k-1$ and $M_0[\sigma_{u_1} t_{\alpha_1} \ldots \sigma_{u_{k-1}} t_{\alpha_{k-1}}\rangle M_{k-1}$. Thus, $M_{k-1} + C_u \cdot y_k \geq Pre(\cdot, t_1^{e_k}) \geq 0$ holds. Since the unobservable subnet is acyclic, there exists a sequence $\sigma_{u_k} \in T_u^*$ such that $M_{k-1}[\sigma_{u_k} t_1^{e_k}\rangle M_k$. Thus, it holds $\sigma_{u_1} t_{\alpha_1} \ldots \sigma_{u_{k-1}} t_{\alpha_{k-1}} \sigma_{u_k} t_{\alpha_k} \in \overleftarrow{\mathcal{C}}(e_1 \ldots e_k)$, i.e., we prove by induction that there exists a sequence $\sigma = \sigma_{u_1} t_{\alpha_1} \ldots \sigma_{u_h} t_{\alpha_h} \in \overleftarrow{\mathcal{C}}(\omega)$.

(*only if*) For the observed word $\omega = e_1 e_2 \ldots e_h$, if there exists a sequence $\sigma = \sigma_{u_1} t_{\alpha_1} \ldots \sigma_{u_h} t_{\alpha_h} \in \overleftarrow{\mathcal{C}}(\omega)$, a solution to Eq. (2) can be found as follows. First, we set $y_i = \pi_u(\sigma_{u_i})$ for $i = 1, \ldots, h$. Second, since $t_{\alpha_i} \in T_{e_i}$, without loss of generality, we assume $t_1^{e_i} = t_{\alpha_i}$ and then it holds $z_1^{e_i} = 0$ and $z_i^{e_i} = 1$ for $i = 2, \ldots, n_{e_i}$. Thus, there exist y_i and $z_j^{e_i}$ with $i = 1, \ldots, h$ and $j = 1, \ldots, n_{e_i}$ satisfying Eq. (2). □

On the basis of Theorem 2, we can infer that all sequences σ's corresponding to the solutions to Eq. (2) constitute the set $\overleftarrow{\mathcal{C}}(\omega)$. Thus, by associating an objective function with Eq. (2), we can determine if a fault has occurred or not. In the following, three corollaries based on Theorem 2 are provided to describe this.

Corollary 1. Given a labeled Petri net $\langle N, M_0, E, \lambda\rangle$ and an observed word $\omega = e_1 e_2 \ldots e_h$, then for each $f \in T_f$, $\Delta(\omega, f) = 0$ if ILPP 1 admits a solution $\gamma = 0$.

$$\text{ILPP 1:} \begin{cases} \gamma = & \max \sum_{i=1}^{h} y_i(f) \\ & \text{s.t. Eq. (2)} \end{cases}$$

Proof. If $\gamma = 0$, by checking all possible solutions to Eq. (2), there does not exist y_i such that $y_i(f) > 0$. By Theorem 2, each σ_{u_i} in $\sigma = \sigma_{u_1} t_{\alpha_1} \ldots \sigma_{u_h} t_{\alpha_h} \in \overleftarrow{\mathcal{C}}(\omega)$ does not contain f, i.e., for all $\sigma \in \overleftarrow{\mathcal{C}}(\omega)$, $f \notin \sigma$, which implies $\Delta(\omega, f) = 0$. □

Corollary 2. Given a labeled Petri net $\langle N, M_0, E, \lambda\rangle$ and an observed word $\omega = e_1 e_2 \ldots e_h$, for each $f \in T_f$, $\Delta(\omega, f) = 2$ if ILPP 2 admits a solution $\gamma > 0$.

$$\text{ILPP 2:} \begin{cases} \gamma = & \min \sum_{i=1}^{h} y_i(f) \\ & \text{s.t. Eq. (2)} \end{cases}$$

Proof. If $\gamma > 0$, then there exists at least one y_i in solutions to Eq. (2) such that $y_i(f) > 0$, i.e., for all $\sigma \in \overleftarrow{\mathcal{C}}(\omega)$, $f \in \sigma$, which indicates $\Delta(\omega, f) = 2$. □

For an observed word ω and each $f \in T_f$, Corollaries 1 and 2 are devoted to computing the value of $\Delta(\omega, f)$. On the other hand, for the overall fault status $\mathcal{F}$, we can compute $\Delta(\omega, \mathcal{F})$ according to Proposition 1 and the following corollary.

Corollary 3. Given a diagnostic Petri net system $\langle N, M_0, E, \lambda, \mathcal{F}\rangle$ and an observed word $\omega = e_1 e_2 \ldots e_h$, $\Delta(\omega, \mathcal{F}) = 2$ if ILPP 3 admits $\gamma > 0$ and $\Delta(\omega, \mathcal{F}) = 0$ or 1 if $\gamma = 0$.

$$\text{ILPP 3:} \begin{cases} \gamma = & \min \vec{1}^{1 \times n_f} \cdot \sum_{i=1}^{h} y_i(T_f) \\ & \text{s.t. Eq. (2)} \end{cases}$$

Proof. Note that objective function $\sum_{i=1}^{h} y_i(T_f)$ denotes the sum of projections of n_u-dimensional column vectors y_i's over the set T_f. If $\gamma = 0$, there exists a group of unobservable sequences σ_{u_i} corresponding to y_i with $i = 1 \ldots h$ such that $\sigma = \sigma_{u_1} t_{\alpha_1} \ldots \sigma_{u_h} t_{\alpha_h}$ and $f \notin \sigma$ for each $f \in T_f$, i.e., there exists at least one path which passes none of fault transitions in the reachability graph from M_0 to $\overleftarrow{\mathcal{D}}(\omega)$. Thus, in this case, $\Delta(\omega, \mathcal{F}) = 0$ or 1 according to Definition 3. On the contrary, if $\gamma > 0$, there does not exist such a path, i.e., each path must pass one or more fault transitions. Thus, we have $\Delta(\omega, \mathcal{F}) = 2$. □

Now, an online algorithm to Problem 2 can be provided. Algorithm 1 describes the basic steps of how to perform diagnosis in a labeled Petri net by using the ILP technique only. The correctness of this algorithm is ensured by Proposition 1 and Corollaries 1, 2 and 3.

We briefly illustrate how Algorithm 1 works. It is described in C-like syntax. For example, we use the symbol "=" to represent an assignment operation and "==" to indicate that two variables are equal. In Line 1, $\mathcal{R} = \vec{0}^{n_f}$ is an n_f-dimensional column vector that records the diagnosis result of each $f \in T_f$. For a fault f, its diagnosis is denoted by $\mathcal{R}(f)$. Note that n_f is the cardinality of set T_f. Variable s represents the diagnosis of the overall fault status $\mathcal{F}$, i.e., $\Delta(\omega, \mathcal{F})$. The diagnosis results $\mathcal{R}$ and s are written into a log file in Line 26. We in Section I discuss how to use the log file to refine the diagnosis result (see Fig. 3(b)). The value of $\Delta(\omega, f)$ with $f \in T_f$, i.e., $\mathcal{R}(f)$, is first computed by Lines 4–13 according to Corollaries 1 and 2. In some cases, $\Delta(\omega, \mathcal{F})$ can be directly derived from $\mathcal{R}$ (see Proposition 1 for details). Thus by Lines

Algorithm 1: Online fault diagnosis using a diagnostic Petri net system

Input: A diagnostic Petri net system $\langle N, M_0, E, \lambda, \mathcal{F}\rangle$
Output: The diagnosis result for each observed event $e \in E$

1 $\omega = \varepsilon$, $\mathcal{R} = \vec{0}^{n_f}$, $s = 0$;
2 Wait until a new event e is observed;
3 $\omega = \omega e$;
4 **for** *each* $f \in T_f$ **do**
5 Set γ_1 be the maximal objective value of ILPP 1;
6 **if** $\gamma_1 == 0$ **then**
7 $\mathcal{R}(f) = \Delta(\omega, f) = 0$;
8 **else**
9 Set γ_2 be the minimal objective value of ILPP 2;
10 **if** $\gamma_2 > 0$ **then**
11 $\mathcal{R}(f) = \Delta(\omega, f) = 2$;
12 **else**
13 $\mathcal{R}(f) = \Delta(\omega, f) = 1$;
14 **if** $\mathcal{R} == \vec{0}$ **then**
15 $s = \Delta(\omega, \mathcal{F}) = 0$;
16 **else if** *there exists an entry* r *in vector* $\mathcal{R}$ *such that* $r == 2$ **then**
17 $s = \Delta(\omega, \mathcal{F}) = 2$;
18 **else if** *there exists only an entry* r *in vector* $\mathcal{R}$ *such that* $r == 1$ **then**
19 $s = \Delta(\omega, \mathcal{F}) = 1$;
20 **else**
21 Set γ_3 be the minimal objective value of ILPP 3;
22 **if** $\gamma_3 > 0$ **then**
23 $s = \Delta(\omega, \mathcal{F}) = 2$;
24 **else**
25 $s = \Delta(\omega, \mathcal{F}) = 1$;
26 Output $\mathcal{R}$ and s and write them into log file; **Goto** 2;

14–19, we compute $\Delta(\omega, \mathcal{F})$ according to the diagnosis of each $f \in T_f$, i.e., $\mathcal{R}$. If $\Delta(\omega, \mathcal{F})$ cannot be directly obtained by $\mathcal{R}$, we have to solve ILPP 3 to determine $\Delta(\omega, \mathcal{F})$ according to Corollary 3 (Lines 21–25). When the algorithm completes the diagnosis of the current step, it returns to Line 2 to wait for a new observed event.

The main computational cost of Algorithm 1 stems from the solutions of ILPPs 1, 2, 3. As known, solving an integer linear programming problem is NP-hard. Further, the computational cost mainly depends on its size,, i.e., the number of integer variables and constraints. We next analyze the size of the programming problem Eq. (2).

For $\omega = e_1 \ldots e_h$, it is readily to verify that the number of variables in Eq. (2) can be represented as

$$\begin{aligned}\mathcal{I} &= (n_u + n_{e_1}) + (n_u + n_{e_2}) + \ldots + (n_u + n_{e_h})\\ &= h \cdot n_u + (n_{e_1} + \ldots + n_{e_h}) \approx h \cdot (n_u + n_f),\end{aligned}$$

where $n_u = |T_u|$[1], $n_f = |T_f|$, $n_{e_i} = |T_{e_i}|$. Eq. (2) contains h groups of constraints and $(n_u + n_{e_i})$ denotes the number of variables in the ith group. Specifically, n_u denotes the length of vector y_i and n_{e_i} the number of binary variables, i.e., $z_1^{e_i}, \ldots, z_{n_{e_i}}^{e_i}$. At the same time, the number of constraints is represented as

$$\begin{aligned}\mathcal{J} &= (m \cdot n_{e_1} + 1) + \ldots + (m \cdot n_{e_h} + 1)\\ &= m \cdot (n_{e_1} + \ldots + n_{e_h}) + h \approx h \cdot (m \cdot n_f + 1),\end{aligned}$$

where m is the number of places in a labeled Petri net. For $e_i \in \omega$, there are n_{e_i} transitions whose labels are e_i. Thus for the ith group of constraints, it contains $m \cdot n_{e_i} + 1$ constraints in scalar form, where "1" denotes the constraint $z_1^{e_i} + \ldots + z_{n_{e_i}}^{e_i} = n_{e_i} - 1$.

In summary, the number of variables and constraints in Eq. (2) is linear in the length of the observed word. If the observed word is very long, we may not be able to obtain the diagnosis result in real time. However, the approach based on ILP is straightforward and easy to implement, and can be used as a basis to develop more efficient approaches.

TABLE II
DIAGNOSIS RESULT AND RUNNING TIME FOR EXAMPLE 2.

	e_1	e_2	e_3	e_4
	a	b	b	a
f_1	0	1	1	1
f_2	0	0	1	1
f_3	0	0	0	0
$\mathcal{F}$	0	1	2	2
Time (s)	0.0115	0.0107	0.0209	0.0233

Example 2. Let us consider the net shown in Fig. 1. Assume that the observed word is $\omega = abba$. The diagnosis result and running time are shown in Table II, where, for each event e_i, there is a column which lists the diagnosis result and running time corresponding to e_i. For example, considering $e_3 = b$ (i.e., $\omega = e_1e_2e_3 = abb$), the diagnosis results are $\Delta(\omega, f_1) = 1$, $\Delta(\omega, f_2) = 1$, $\Delta(\omega, f_3) = 0$, and $\Delta(\omega, \mathcal{F}) = 2$ and the running time is 0.0209s. Note that the time is obtained by executing a MATLAB procedure with GUROBI solver (academic license) [33] on a laptop computer with Intel i5-4200M 2.5GHz processor and 8G DDR3 1600Hz RAM.

V. DIAGNOSABILITY AND OVERALL FAULT STATUS

The diagnosability of a Petri net is discussed in [34]–[36]. If a fault f is diagnosable, then the occurrence of f can be detected in a finite number of steps. Given a Petri net language $\mathcal{L}$, its post-language after a transition sequence $\sigma \in \mathcal{L}$ is defined as $\mathcal{L}/\sigma = \{\tau \in T^* \mid \sigma\tau \in \mathcal{L}\}$. Formally, the diagnosability of f is defined as follows.

Definition 4. [35] Given a labeled net $\langle N, M_0, E, \lambda\rangle$, a fault transition f is *diagnosable* if there exists an integer $K \in \mathbb{N}$ such that

$$\begin{aligned}&\forall \sigma = \tilde{\sigma} f \in \mathcal{L}(N, M_0), \forall \tau \in \mathcal{L}(N, M_0)/\sigma \text{ with } |\tau| > K\\ &\Rightarrow \forall \sigma' \in \overleftarrow{\mathcal{C}}(\lambda(\sigma\tau)), f \in \sigma', \text{ where } \tilde{\sigma} \in \mathcal{L}(N, M_0).\end{aligned}$$

[1] $|\cdot|$ denotes the cardinality of a set.

If all fault transitions of a Petri net system are diagnosable, then the Petri net system is said to be *diagnosable*. For a diagnosable labeled Petri net, we have the following proposition.

Proposition 2. Consider a diagnosable labeled Petri net system $\langle N, M_0, E, \lambda\rangle$ with an overall fault status $\mathcal{F}$. For each sequence $\sigma f \sigma' \in \mathcal{L}(N, M_0)$, it holds $\Delta(\omega, \mathcal{F}) = 2$, where $\sigma \in T^*$, $f \in T_f$, $\sigma' \in T^*$ with $|\sigma'| > K$, and $\omega = \lambda(\sigma f \sigma')$.

Proof. Since f is diagnosable, for each $\sigma_1 \in \overleftarrow{\mathcal{C}}(\omega)$, it holds $f \in \sigma_1$. Thus, we have $\Delta(\omega, \mathcal{F}) = 2$ by Definition 3. □

In plain words, when a system is running, a fault $f \in T_f$ occurs and the process continues. After K steps, we are sure that f has occurred according to the observation ω. Thus, $\Delta(\omega, f) = 2$ is true, which implies $\Delta(\omega, \mathcal{F}) = 2$.

Only when the Petri net model of a system is diagnosable (or I-diagnosable [1]), can the diagnosis algorithms detect unambiguously which faults have occurred. However, due to the use of overall fault status $\mathcal{F}$, even if a net model of a system is not diagnosable, it is possible to detect the occurrence of faults in the system though we do not know exactly which fault occurs. An example is provided to clarify this.

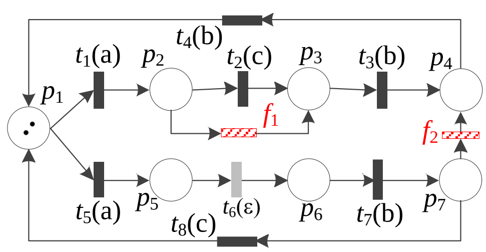

Fig. 4. A Petri net that is not diagnosable.

Example 3. Consider the net shown in Fig. 4, where $T_u = \{t_6, f_1, f_2\}$ and $T_f = \{f_1, f_2\}$. According to Definition 4, both f_1 and f_2 are not diagnosable. Assume that the observed word is $\omega = (abb)(abb)\cdots$. Then it holds $\Delta(\omega, f_1) = \Delta(\omega, f_2) = 1$, i.e., the diagnosis results of f_1 and f_2 are ambiguous for infinite ω. However, for $\omega' = abb$, it holds $\Delta(\omega', \mathcal{F}) = 2$, i.e., when observing only abb, we have already known that some faults occur in the system. In this case, although we are sure that at least a fault has occurred, we are unable to exactly identify which faults have occurred. In real-world systems, a possible strategy is to stop the plant and inspect the faults one by one.

VI. Case study

We in this section explore the computational overhead of the proposed algorithm and compare its efficiency with the one shown in [30] by an example. Consider the labeled Petri net shown in Fig. 5 that is originally introduced in [20] and slightly modified in this paper, where $T_u = \{t_1, f_1, t_4, t_{11}, f_2\}$, $T_f = \{f_1, f_2\}$, $T_a = \{t_3, t_5, t_6, t_8\}$, $T_e = \{t_7, t_9, t_{10}, t_{12}\}$, $T_h = \{t_{16}\}$, $T_g = \{t_{14}, t_{15}, t_{17}\}$. The initial marking is $M_0 = p_1 + \alpha p_2 + \beta p_3$, where α and β are two variables denoting the numbers of tokens in p_2 and p_3, respectively.

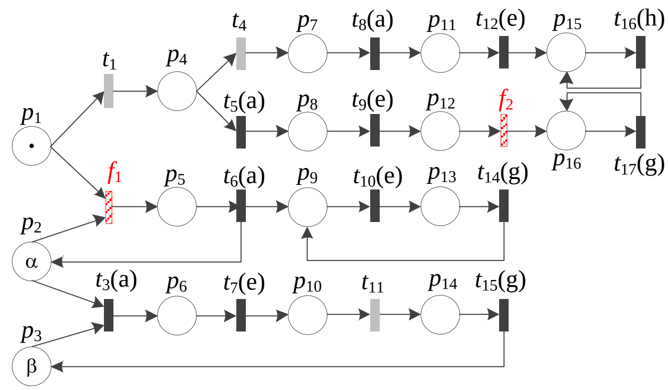

Fig. 5. A Petri net for case study.

Let $\alpha = 10$ and $\beta = 10$, and assume that the observed word is $\omega = ae(aeg)^{10}gggg$, where $(aeg)^{10}$ represents that the sequence aeg repeats 10 times. When observing an event, we make a diagnosis using Algorithm 1. The running time of Algorithm 1 for each observed event is shown in Fig. 6, where the x-axis represents the index of each event in sequence ω.

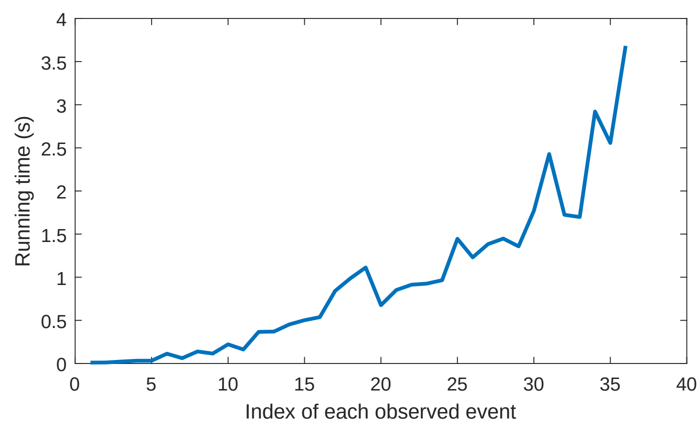

Fig. 6. Running time for each observed event.

We have shown in Section IV that the size of programming model Eq. (2) is linear with respect to the length of the observed word. However, the difficulty of a generic ILP problem always increases exponentially with respect to its size, which is verified by Fig. 6. This implies that one probably cannot obtain the diagnosis result in real time if the observed sequence is very long. However, integer linear programming is a standard mathematical tool for diagnosis of Petri nets based on which some new approaches can be developed to overcome the complexity issue, which will be done in our subsequent work.

Fanti *et al.* [30] also proposed an algorithm for fault diagnosis using labeled Petri nets and integer linear programming. However, the key idea is different from us. They first build an integer programming problem according to a transition sequence $\sigma \in T_o^*$ not an observed word $\omega \in E^*$. Then, for an observed word ω, they explore all possible sequences of observable transitions whose projections over E are equal to ω. The size of the programming problem in [30] is smaller than the one in this paper. However, it has to be solved more times to obtain the diagnosis result.

The output and computational process of the algorithm proposed by Fanti *et al.* [30] are different from ours. In order to compare the efficiency of these two algorithms, we have to modify one of them to make them have the same input and output. We here choose to modify Fanti's algorithm, though it is completely feasible to modify our algorithm (note that modifying our algorithm will lose some diagnostic information). The details of the modification of Fanti's algorithm is discussed in Appendix A and this section mainly focuses on the comparison of these two algorithms. On the other hand, our algorithm, i.e., Algorithm 1, and the modified version of Fanti's algorithm are both implemented in MATLAB language. The readers can refer to [37] for the source code.

Consider the net in Fig. 5 again and assume that $\alpha = \beta = 7$. If the observed word is $\omega = ae(aeg)^7gggg$, the comparison of these two algorithms is shown in Table III. The first row lists all events in ω. The second row shows the number of times to solve the programming problems defined in this paper when performing diagnosis using the proposed algorithm. The third row shows the number of times to solve the programming problems defined in [30] when dealing with the diagnosis issue using the algorithm (modified version) developed in [30]. The fourth and fifth rows demonstrate the running time of diagnosis algorithms proposed by us and Fanti *et al.* [30], respectively. The running time is tested using GUROBI solver [33] on a laptop computer with Intel i5-4200M 2.5GHz processor and 8G DDR3 1600Hz RAM. The sixth row lists the diagnosis result for each event, which is represented as a row vector $[a\ b\ c]$ such that $\Delta(\omega, f_1) = a$, $\Delta(\omega, f_2) = b$, and $\Delta(\omega, \mathcal{F}) = c$. Note that the diagnosis of the fourth event from the last is $[1\ 1\ 2]$, i.e., we detect the occurrence of faults ($\Delta(\omega, \mathcal{F}) = 2$) before the exact faults are ascertained. The detailed comparison of running time is also illustrated by Fig. 7. We observe that our approach is more efficient in this example.

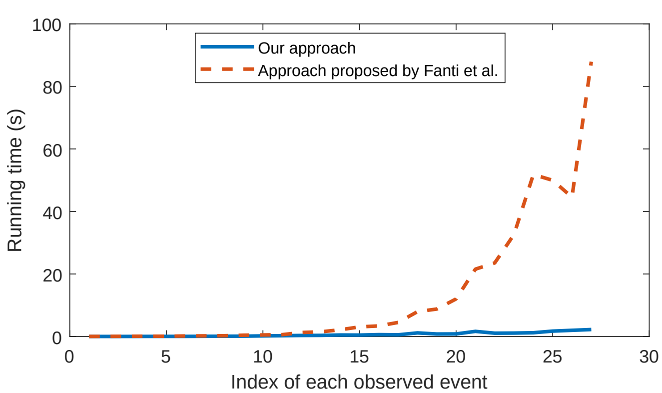

Fig. 7. Comparison of our approach with the one in [30].

VII. Conclusion

This paper addresses the problem of fault diagnosis by formulating and solving ILP problems. The main contributions consist in introducing the overall fault status and proposing an online diagnosis algorithm based on labeled Petri nets, in which two or more transitions can share the same label. The overall fault status provides a more informative diagnosis result, i.e., not only every fault but also the global system fault status can be detected. In addition, we show that, in some cases, a definite conclusion on the occurrence of faults in a system can be given even if the system is not diagnosable. We also compare the efficiency of the proposed approach with the one in [30] by a case study and the result shows that our approach is usually more efficient.

In future work, we plan to extend and modify the other diagnosis approaches, such as those in [25] and [21], to make them have a uniform interface. Then, we will develop a software package to compare their efficiency. On the other hand, we will explore the use of an overall fault status in a distributed environment.

References

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

[2] ——, "Failure diagnosis using discrete-event models," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 2, pp. 105–124, 1996.

[3] Y. F. Chen, Z. W. Li, K. Barkaoui, N. Q. Wu, and M. C. Zhou, "Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 2, pp. 364–379, 2017.

[4] Y. F. Chen, Z. W. Li, K. Barkaoui, and A. Giua, "On the enforcement of a class of nonlinear constraints on Petri nets," *Automatica*, vol. 55, pp. 116–124, 2015.

[5] G. H. Zhu, Z. W. Li, and N. Q. Wu, "Model-based fault identification of discrete event systems using partially observed Petri nets," *Automatica*, vol. 96, pp. 201–212, 2018.

[6] A. Giua and C. Seatzu, "Identification of free-labeled Petri nets via integer programming," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, 2005, pp. 7639–7644.

[7] Z. He, Z. W. Li, and A. Giua, "Performance optimization for timed weighted marked graphs under infinite server semantics," *IEEE Transactions on Automatic Control*, vol. 63, no. 8, pp. 2573–2580, 2018.

[8] W. M. P. van der Aalst, *Process Discovery: An Introduction*. New York, NY, USA: Springer, 2011.

[9] H. M. Zhang, L. Feng, and Z. W. Li, "A learning-based synthesis approach to the supremal nonblocking supervisor of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 63, no. 10, pp. 3345–3360, 2018.

[10] Z. Y. Ma, Y. Tong, Z. W. Li, and A. Giua, "Basis marking representation of Petri net reachability spaces and its application to the reachability problem," *IEEE Transactions on Automatic Control*, vol. 62, no. 3, pp. 1078–1093, 2017.

[11] Y. Tong, Z. W. Li, C. Seatzu, and A. Giua, "Verification of state-based opacity using Petri nets," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2823–2837, 2017.

[12] F. Basile, M. P. Cabasino, and C. Seatzu, "State estimation and fault diagnosis of labeled time Petri net systems with unobservable transitions," *IEEE Transactions on Automatic Control*, vol. 60, no. 4, pp. 997–1009, 2015.

[13] D. Lefebvre, "On-line fault diagnosis with partially observed Petri nets," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1919–1924, 2014.

[14] L. Li and C. N. Hadjicostis, "Minimum initial marking estimation in labeled Petri nets," *IEEE Transactions on Automatic Control*, vol. 58, no. 1, pp. 198–203, 2013.

[15] J. Prock, "A new technique for fault detection using Petri nets," *Automatica*, vol. 27, no. 2, pp. 239–245, 1991.

[16] Y. Wu and C. N. Hadjicostis, "Algebraic approaches for fault identification in discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 50, no. 12, pp. 2048–2055, 2005.

[17] A. Ramírez-Treviño, E. Ruiz-Beltrán, I. Rivera-Rangel, and E. Lopez-Mellado, "Online fault diagnosis of discrete event systems. A Petri net-based approach," *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 1, pp. 31–39, 2007.

TABLE III
COMPARISON OF OUR APPROACH WITH THE ONE IN [30].

event	a	e	a	e	g	$\cdots$	e	g	g	g	g	g
solving times I	3	3	5	5	5	$\cdots$	5	5	5	3	3	3
solving times II	8	22	27	39	41	$\cdots$	5591	7512	10652	9988	9973	18583
running time I (s)	0.02	0.01	0.02	0.04	0.04	$\cdots$	1.07	1.10	1.21	1.73	1.98	2.23
running time II (s)	0.03	0.05	0.08	0.10	0.09	$\cdots$	23.52	32.63	51.80	49.97	44.57	87.89
diagnosis result	[1 0 1]	[1 0 1]	[1 1 1]	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	[1 1 2]	[0 2 2]	$\cdots$	$\cdots$

[18] A. Benveniste, E. Fabre, S. Haar, and C. Jard, "Diagnosis of asynchronous discrete-event systems: a net unfolding approach," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 714–727, 2003.

[19] J. Zaytoon and S. Lafortune, "Overview of fault diagnosis methods for discrete event systems," *Annual Reviews in Control*, vol. 37, no. 2, pp. 308–320, 2013.

[20] S. Genc and S. Lafortune, "Distributed diagnosis of discrete-event systems using Petri nets," in *Proceedings of International Conference on Application and Theory of Petri Nets*, Eindhoven, the Netherlands, 2003, pp. 316–336.

[21] A. Giua and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," in *Proceedings of the 44th IEEE Conference on Decision and Contro*, Seville, Spain, 2005, pp. 6323–6328.

[22] M. P. Cabasino, A. Giua, and C. Seatzu, "Fault detection for discrete event systems using Petri nets with unobservable transitions," *Automatica*, vol. 46, no. 9, pp. 1531–1539, 2010.

[23] M. P. Cabasino, A. Giua, M. Pocci, and C. Seatzu, "Discrete event diagnosis using labeled Petri nets. An application to manufacturing systems," *Control Engineering Practice*, vol. 19, no. 9, pp. 989–1001, 2011.

[24] M. P. Cabasino, A. Giua, A. Paoli, and C. Seatzu, "Decentralized diagnosis of discrete-event systems using labeled Petri nets," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 6, pp. 1477–1485, 2013.

[25] F. Basile, P. Chiacchio, and G. De Tommasi, "An efficient approach for online diagnosis of discrete event systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 4, pp. 748–759, 2009.

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

[27] Y. Ru and C. N. Hadjicostis, "Fault diagnosis in discrete event systems modeled by partially observed Petri nets," *Discrete Event Dynamic Systems*, vol. 19, no. 4, pp. 551–575, 2009.

[28] X. Wang, C. Mahulea, and M. Silva, "Diagnosis of time Petri nets using fault diagnosis graph," *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2321–2335, 2015.

[29] W. Hamscher, L. Console, and J. De Kleer, *Readings in Model-Based Diagnosis*. San Mateo, CA: Morgan Kaufmann, 1992.

[30] M. P. Fanti, A. M. Mangini, and W. Ukovich, "Fault detection by labeled Petri nets in centralized and distributed approaches," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 392–404, 2013.

[31] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541–580, 1989.

[32] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. New York, NY, USA: Springer, 2009.

[33] Gurobi Optimizer. [Online]. Available: http://www.gurobi.com/

[34] M. P. Cabasino, A. Giua, and C. Seatzu, "Diagnosability of bounded petri nets," in *Proceedings of the 48th IEEE Conference on Decision and Control*, Shanghai, China, 2009, pp. 1254–1260.

[35] ——, "Diagnosability of discrete-event systems using labeled Petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 11, no. 1, pp. 144–153, 2014.

[36] F. Basile, P. Chiacchio, and G. De Tommasi, "On k-diagnosability of Petri nets via integer linear programming," *Automatica*, vol. 48, no. 9, pp. 2047–2058, 2012.

[37] G. H. Zhu, "Matlab programs for this paper." [Online]. Available: https://github.com/zhuguanghui86/code_for_mypaper

APPENDIX A
MODIFICATION DETAILS OF THE ALGORITHM IN [30]

The centralized fault diagnosis algorithm proposed in [30] has a different output with the one in this paper and thus we need to modify it, keeping the key idea unchanged, to compare its efficiency with our approach. We first briefly recall the ILP problem defined in [30] based on a sequence $\sigma_o \in T_o^*$.

Given a sequence of observable transitions denoted by $\sigma_o = t_{\alpha_1} t_{\alpha_2} \cdots t_{\alpha_h}$, the ILP problem without objective function can be defined as

$$\begin{cases} M_0 + C_u \cdot \sum\limits_{k=1}^{i} y_k + \sum\limits_{k=1}^{i-1} C(\cdot, t_{\alpha_k}) \geq Pre(\cdot, t_{\alpha_i}) \\ y_i \in \mathbb{N}^{n_u} \\ i = 1, \ldots, h. \end{cases} \tag{3}$$

By specifying different objective functions to Eq. (3), we obtain three ILP models:

$$\begin{cases} \text{ILPP 4: } \phi_1 = \max \sum\limits_{i=1}^{h} y_i(f) & \textit{s.t. } \text{Eq. (3)} \\ \text{ILPP 5: } \phi_2 = \min \sum\limits_{i=1}^{h} y_i(f) & \textit{s.t. } \text{Eq. (3)} \\ \text{ILPP 6: } \phi_3 = \min \vec{1}^{1 \times n_f} \cdot \sum\limits_{i=1}^{h} y_i(T_f) & \textit{s.t. } \text{Eq. (3).} \end{cases}$$

On the basis of these ILP problems, the Fault Detection Algorithm (FDA) proposed in [30] (see Fig. 2 in [30]) is modified as Algorithm 2 which takes a transition sequence $\sigma_o \in T_o^*$ as input. At the same time, the Diagnoser Algorithm (DA) proposed in [30] (see Fig. 3 in [30]) is modified as Algorithm 3 which enumerates all sequences of observable transitions consistent with an observed word and repetitively calls Algorithm 2. The data shown in rows 3 and 5 of Table III is computed by executing Algorithm 3. The readers can inspect the souce code [37] for the details.

Note that the symbol $\otimes$ (not mentioned in [30]) in Line 11 of Algorithm 3 is another contribution of this paper, which is a binary operation defined in Table IV and very appropriate to compute the combination of diagnosis results of transition sequences σ_o's consistent with an observed word ω. We will extend the approaches shown in [21] and [25] to the case of labeled Petri nets using this symbol in the subsequent research. We next show the formal definition of symbol $\otimes$ and prove its correctness.

Analogous to the labeling function λ, we define a new function $\tau\colon T \to T_o \cup \{\varepsilon\}$ such that $\tau(t) = t$ if $t \in T_o$ and $\tau(t) = \varepsilon$ if $t \in T_u$. The function τ is extended to a sequence $\sigma t \in T^*$ such that $\tau(\sigma t) = \tau(\sigma)\tau(t)$, i.e., $\nu = \tau(\sigma)$ represents

Algorithm 2: A fault diagnosis algorithm denoted by $(\mathcal{R}, s, \chi) = \text{new\_FDA}(\mathcal{N}, \sigma_o)$

Input: A diagnostic Petri net system $\mathcal{N} = \langle N, M_0, E, \lambda, \mathcal{F}\rangle$ and a transition sequence $\sigma_o \in T_o^*$
Output: The diagnosis results $\mathcal{R}$ and s of fault transitions and the overall fault status, respectively, a flag χ to denote if an ILP problem admits a solution

$\mathcal{R} = \vec{0}^{n_f}$, $s = 0$, $\chi =$ **true**, build Eq. (3) according to σ_o;
if *Eq. (3) has no feasible solution* **then**
    $\chi =$ **false**; **return** $\mathcal{R}$, s and χ (terminate the procedure);
for *each* $f \in T_f$ **do**
    if $\phi_1 == 0$ **then**
        $\mathcal{R}(f) = \Delta(\omega, f) = 0$;
    else
        if $\phi_2 == 0$ **then**
            $\mathcal{R}(f) = \Delta(\omega, f) = 1$;
        else
            $\mathcal{R}(f) = \Delta(\omega, f) = 2$;
if $\mathcal{R} == \vec{0}$ **then**
    $s = \Delta(\omega, \mathcal{F}) = 0$;
else if *there exists* r *in vector* $\mathcal{R}$ *such that* $r == 2$ **then**
    $s = \Delta(\omega, \mathcal{F}) = 2$;
else if *there exists only* r *in vector* $\mathcal{R}$ *such that* $r == 1$ **then**
    $s = \Delta(\omega, \mathcal{F}) = 1$;
else
    if $\phi_3 == 0$ **then**
        $s = \Delta(\omega, \mathcal{F}) = 1$;
    else
        $s = \Delta(\omega, \mathcal{F}) = 2$;
return $\mathcal{R}$, s and χ;

Algorithm 3: An online fault diagnosis algorithm

Input: A diagnostic Petri net system $\mathcal{N} = \langle N, M_0, E, \lambda, \mathcal{F}\rangle$
Output: The diagnosis results $\mathcal{R}$ and s for each event e

$\mathcal{R} = \vec{0}^{n_f}$, $s = 0$, $\Lambda' = \{\varepsilon\}$, $\alpha =$ **true** (α denotes if e is the first event);
Wait until a new event e is observed; $\Lambda = \emptyset$;
for *each* $t \in T_e$ **do**
    for *each* $\sigma'_o \in \Lambda'$ **do**
        $\sigma_o = \sigma'_o t$; $(\mathcal{R}', s', \chi) = \text{new\_FDA}(\mathcal{N}, \sigma_o)$;
        if χ *is* **true then**
            $\Lambda = \Lambda \cup \{\sigma_o\}$;
            if α *is* **true then**
                $\mathcal{R} = \mathcal{R}'$, $s = s'$, $\alpha =$ **false**;
            else
                $\mathcal{R} = \mathcal{R} \otimes \mathcal{R}'$; $s = s \otimes s'$;
$\Lambda' = \Lambda$;
Output $\mathcal{R}$, s; **Goto** 2;

the projection of $\sigma \in T^*$ on the set of observable transitions. For an observed word $\omega \in \mathcal{L}^E(N, M_0)$, we denote

$$\mathcal{T}(\omega) = \{\nu \in T_o^* \mid \sigma \in \overleftarrow{\mathcal{C}}(\omega), \nu = \tau(\sigma)\}$$

the set of observable projections of transition sequences consistent with ω.

Proposition 3. Given a diagnostic net system $\langle N, M_0, E, \lambda, \mathcal{F}\rangle$ and an observed word $\omega \in E^*$, for each $f \in T_f \cup \{\mathcal{F}\}$, it holds

$$\Delta(\omega, f) = \Delta(\nu_1, f) \otimes \Delta(\nu_2, f) \otimes \ldots \otimes \Delta(\nu_k, f), \quad (4)$$

where $k = |\mathcal{T}(\omega)|$, $\{\nu_1, \ldots, \nu_k\} = \mathcal{T}(\omega)$, and $\otimes\colon \{0,1,2\} \times \{0,1,2\} \to \{0,1,2\}$ is a binary operation defined in Table IV.

TABLE IV
BINARY OPERATION $\otimes$.

$\otimes$	0	1	2
0	0	1	1
1	1	1	1
2	1	1	2

Proof. For $f \in T_f \cup \{\mathcal{F}\}$ and $\nu \in \mathcal{T}(\omega)$, we can compute $\Delta(\nu, f)$ according to Algorithm 2. $\mathcal{T}(\omega)$ denotes all sequences of observable transitions whose projections on E are ω. Assume that there are only two items ν_1 and ν_2 in $\mathcal{T}(\omega)$. For $f \in T_f$, if $\Delta_1 = \Delta(\nu_1, f) = 0$ and $\Delta_2 = \Delta(\nu_2, f) = 0$, there is no fault f to occur in paths consistent with ν_1 and ν_2 according to Definition 2, and thus we have $\Delta(\omega, f) = 0$. On the other hand, if $\Delta_1 = 0$ and $\Delta_2 = 2$, there is no fault f in paths consistent with ν_1 but fault f must occur in all paths consistent with ν_2. Thus, $\Delta(\omega, f) = 1$ is true according to Definition 2. Following a similar procedure, we obtain Table IV for each $f \in T_f \cup \{\mathcal{F}\}$ according to Definitions 2 and 3. If $\mathcal{T}(\omega)$ contains more than two elements, Eq. (4) is readily verified. $\square$