Engineering Applications of Artificial Intelligence 53 (2016) 53–61

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Probabilistic logical approach for testing diagnosability of stochastic discrete event systems

Xuena Geng [a,b], Dantong Ouyang [a,b,*], Xiangfu Zhao [c], Shuang Hao [a,b]

[a] Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun, China
[b] College of Computer Science and Technology, Jilin University, Changchun, China
[c] College of Mathematics, Physics, and Information Engineering, Zhejiang Normal University, Jinhua, China

ARTICLE INFO

Article history:
Received 5 January 2016
Received in revised form
25 March 2016
Accepted 27 March 2016
Available online 13 April 2016

Keywords:
Fault diagnosis
Stochastic discrete event system
Logical diagnoser
Logical diagnosability
A-diagnosability
AA-diagnosability

ABSTRACT

Fault diagnosis plays an important role in the prevention of harmful events in discrete event systems (DESs). Stochastic DES (SDES) is a more precise formulation of DES. In this paper, a novel approach that uses probabilistic logic to diagnose SDES is investigated. SDES is formalized as a set of probabilistic logical formulas. Moreover, a logical diagnoser is presented. Fault diagnosis of SDES has two properties: A-diagnosability and AA-diagnosability. On the basis of resolution principle, an algorithm is proposed to test A-diagnosability and AA-diagnosability of the SDES. Experimental results demonstrate that our algorithm improves the accuracy and efficiency of verifying diagnosability of SDES.

1. Introduction

Discrete event system (DES) is a discrete-state, event-driven system, the states evolution of which depends entirely on the occurrence of asynchronous discrete events over time. However, the states and events of DESs are not sufficiently accurate in most engineering applications. To deal with the diagnosis problem of DESs precisely, stochastic discrete event systems (SDESs) were proposed by Lunze and Schroder (2001). Thereafter, many other works on fault diagnosis of SDES were proposed (Thorsley and Teneketzis, 2005; Chen and Kumar, 2013; Liu and Qiu, 2008; Chen and Kumar, 2013; Liu et al., 2008).

In recent years, the detection of failure events in DES has received considerable attention to guarantee the performance of a reliable system. Diagnosability is an important property in fault diagnosis. Verifying diagnosability of a system, which was proposed in the 1990s (Sampath et al., 1995, 1996), is crucial. Diagnosability property refers to the ability to detect the occurrence of failure events on the basis of observations and using model-based inferencing. In Sampath et al. (1995), the complexity of the method of testing diagnosability is exponential in the number of states of the system and doubly exponential in the number of failure types. To improve efficiency of verifying the diagnosability, Jiang proposed a twin-plant method, whose complexity is of fourth-order in the number of states of the system and linear in the number of the failure types, to diagnose the DES (Jiang et al., 2001). Pencole (2004) and Debouk et al. (2000) extended the diagnosis to the DESs with decentralized information. Carlos et al. (2012), Lafortune et al. (2001), and Li et al. (2014) improved the efficiency of diagnosability.

Considering that DES cannot distinguish highly probable and less probable strings or states, A-diagnosability and AA-diagnosability of SDES were proposed (Thorsley and Teneketzis, 2005). Let logical diagnosability represent diagnosability in DES. The condition for logical diagnosability, A-diagnosability, and AA-diagnosability has a digressive stringency. In Thorsley and Teneketzis (2005), a necessary and sufficient condition for A-diagnosability and a sufficient condition for AA-diagnosability were proposed. Both algorithms are exponential in the number of states. In Chen and Kumar (2013), a polynomial complexity test for checking A-diagnosability and AA-diagnosability was proposed. The algorithm in Chen and Kumar (2013) uses the twin-plant structure and is similar to the method in Jiang et al. (2001). The complexity of the algorithm is of sixth-order in the number of states. After Thorsley and Teneketzis (2005) and Chen and Kumar (2013), some approaches of testing diagnosability of SDES were proposed, including Liu and Qiu (2008), Chen and Kumar (2013), and Liu et al. (2008). Liu and Qiu (2008) defined safe diagnosability for SDES, in which failure detection occurs before any given

* Corresponding author at: Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun, China.
E-mail address: ouyangdantong@163.com (D. Ouyang).

http://dx.doi.org/10.1016/j.engappai.2016.03.008
0952-1976/

forbidden string in the failed mode of system is executed. Liu et al. (2008) and Chen and Kumar (2013) extended the decentralized failure diagnosis to SDES. Decentralized failure diagnosis is more efficiency than centralized diagnosis in DES. Similarly, decentralized failure diagnosis of SDES also performs better than centralized failure diagnosis in SDES. However, the main ideas of these methods are based on Thorsley and Teneketzis (2005) and Chen and Kumar (2013). The performance of the algorithms in Thorsley and Teneketzis (2005) and Chen and Kumar (2013) was not sufficiently satisfactory. Therefore, we propose a novel approach to verify diagnosability of the SDESs.

Testing diagnosability is a path-finding problem. An efficient approach to solve a path-finding problem, such as AI planning (Kautz and Selman, 1996) and model-checking (Biere et al., 1999), is to reduce them to propositional logic problems (Zhao and Ouyang, 2009, 2015). This approach is similar to diagnosability testing, which is what we are pursuing in this paper. Therefore, we use logical expressions to present SDES and propose a logical diagnoser to test A-diagnosability and AA-diagnosability of the SDESs in this paper. Our algorithm can achieve diagnosability directly and avoid any synchronization operations. Consequently, our algorithm is less costly than the previous methods.

The contributions of this paper are three-fold. First, we present the SDES by using logical expressions. Second, we propose an algorithm to construct a logical diagnoser, i.e., using logical formulas to present a diagnoser. Third, we propose an algorithm to test diagnosability of the SDES on the basis of resolution principle in propositional logic. Examples and comparisons are given in this paper to illustrate the efficiency and accuracy of the proposed algorithms.

This paper is organized as follows. Some definitions and frequently used terms are introduced in Section 2. In Section 3, we construct the logical diagnoser. We present the algorithms of testing A-diagnosability and AA-diagnosability in Section 4. In section 5, we apply our algorithms to the system heating, ventilation, and air-conditioning (HVAC). Section 6 shows the experiment results. Finally, Section 7 presents a summary of the results in the paper and gives the concluding remarks.

2. Preliminaries

Some definitions and terms of logical expression and diagnosis are presented in this section.

A literal is either a variable l or its negation $\neg l$. Symbol "$\vee$" represents logical OR of the literals, and symbol "$\wedge$" represents logical AND of the literals. A clause is the logical OR of one or more literals. For example, x, y, and z are three variables. $(x \vee \neg y)$ and $(x \vee z)$ are two clauses.

A DES is usually modeled as a finite state automaton (FSA) defined in Sampath et al. (1995).

Definition 1 (*FSA*). An FSA is defined as a tuple $G = (X, \Sigma, T, x_0)$, where X is the state space, Σ is the set of events, $T \subseteq X \times \Sigma \times X$ is the partial transition function, and x_0 is the initial state of the system.

The event set Σ is partitioned as $\Sigma = \Sigma_o \cup \Sigma_{uo}$, where Σ_o and Σ_{uo} denote the sets of observable and unobservable events, respectively. Note that $\Sigma_f \subseteq \Sigma_{uo} \subseteq \Sigma$ denotes the set of failure events to be diagnosed. Observable events can be directly detected by the sensors; hence, failure events are supposed to be unobservable. Let Σ^* denote the set of all sequences formed by events in Σ. The behavior of the system is described by the prefix-closed language L, where L is a subset of Σ^*. A *path* denotes an arbitrary element of L. A *trace* is a special path that begins from the initial state of G and ends with the infinite loop of G. Suppose t is a trace of G, projection $Pj(t)$ removes the unobservable events from t. The inverse operation of projection is $Pj_L^{-1}(s_0) = \{s \in L: Pj(s) = s_0\}$. L/s denotes the set of possible continuations of a path s. $|s|$ represents the number of events in s.

We treat the abnormal behaviors in the system as failure events. Since the abnormal behaviors are different in the system, the set of failure events can be partitioned into disjoint sets that correspond to different failure types, i.e., $\Sigma_f = \Sigma_{f_1} \cup \ldots \cup \Sigma_{f_m}$. Let s_f denote the final event of a path s. Define

$$\Phi(\Sigma_{f_i}) = \{s \in L: s_f \in \Sigma_{f_i}\}. \tag{1}$$

The result of function $\Phi(\Sigma_{f_i})$ represents the set of paths whose final event is the failure event of a specific type. Hereafter, F_i denotes the failure events whose type is Σ_{f_i}. For the sake of simplicity, we introduce our algorithms by the systems with only a single failure type. In Section 5, we extend our algorithms to multiple failure types.

In FSA, a transition between x and x' is presented as $tran(x, e) = x'$, where $x, x' \in X$, and $e \in \Sigma$, and indicates that x' is reachable from x driven by event e. Furthermore, the transition function can be extended to the sequence of events as follows:

$$tran(x, se) = tran(tran(x, s), e). \tag{2}$$

SDES is usually modeled as a stochastic automaton (SA), which is an FSA with probabilistic structure.

Definition 2 (*SA Thorsley and Teneketzis, 2005*). An SA is defined as a tuple $G_s = (X, \Sigma, P, x_0)$, where X is the state space, Σ is the set of events, $P: X \times \Sigma \times X \to [0, 1]$ is the partial state transition probability function, and x_0 is the initial state of the SDES.

In SA, X, Σ, and x_0 are the same as those in FSA. The difference between FSA and SA is that FSA has a set of transition functions T instead of P in SA. In SA, $p(x', e|x)$ is a state transition probability defined for $tran(x, e) = x'$ in FSA. The value of $p(x', e|x)$ is the probability of the system evolution from x to x' driven by event e. Therefore, each SA corresponds to an FSA by erasing the probability of the transition function. In this paper, let G_s represent an SA, and let G represent G_s's framework of FSA.

In order to facilitate the solution to the diagnosis problems, we formulate three assumptions about the transition probability (Thorsley and Teneketzis, 2005; Sampath et al., 1995):

(1) At most one $x' \in X$ exists, such that $p(x', e|x) > 0$ for a given $x \in X$ and a given $e \in \Sigma$.
(2) For every state in X, the probability of a transition occurring from that state is one or, equivalently, for $\forall\, x \in X$

$$\sum_{x' \in X} \sum_{e \in \Sigma} p(x', e|x) = 1. \tag{3}$$

(3) There does not exist any cycle of unobservable events, i.e.,

$$(\exists\, n_0 \in \mathbb{N})(\forall\, ust \in L)[(s \in \Sigma_{uo}^*) \Rightarrow |s| \le n_0] \tag{4}$$

Intuitively, assumptions (1) and (2) indicate that transitions will continue to occur in any state. Assumption (3) ensures that the DES does not exist an arbitrarily long trace of unobservable events.

Example 1. Consider the SA G_{s1} shown in Fig. 1. $tran(2, b) = 3$ is a transition of G_1. $p(3, b|2) = 0.1$ is the transition probability of $tran(2, b) = 3$. In Fig. 1, three traces in G_{s1} exist, i.e., $\sigma_{uo} a^*$, $\sigma_f a^*$, and $\sigma_f a^* b^*$. The projection of trace $\sigma_f a^*$ is a^*. The inverse projection of a^* is $(\sigma_{uo} a^* + \sigma_f a^*)$.

Transition probability can also be extended to the path. Before computing the probability of a sequence, we introduce a calculation rule.