Diagnosability in Concurrent Probabilistic Systems*

Xiaowei Huang
School of Computer Science and Engineering
University of New South Wales, Australia
xiaoweih@cse.unsw.edu.au

ABSTRACT

Diagnosability is a key attribute of systems to enable the detection of failure events by partial observations. This paper addresses the diagnosability in concurrent probabilistic systems. Four different notions (L-, P-, A-, and AA-diagnosability) are characterised by formulas of a logic of knowledge, time and probability. Also, we investigate the computational complexities of verifying them: the L-diagnosability is NL-complete, the A-diagnosability is PTIME-complete, and the P-diagnosability is in PSPACE.

Categories and Subject Descriptors

D.2.4 [**Software/Program Verification**]: Reliability

General Terms

Reliability, Verification

Keywords

Diagnosis, Concurrent Probabilistic System, Logic of Knowledge, Probabilistic Reasoning, Computational Complexity

1. INTRODUCTION

Diagnosis is an area of artificial intelligence that focuses on the development of algorithms and techniques to determine whether the behaviour of a system is correct. The computation of diagnosis is based on the partial observation on the system's behaviour by utilising e.g., sensors. *Model based diagnosis* simulates the behaviour of the system by a model (e.g., discrete event systems, automata, Petri nets, etc.) and then compares the observations made on the system and on the model to determine the occurrence of failure events. The model based diagnosis systems have been applied in critical systems e.g., Livingstone [22] in spacecraft fault protection and TEAMS and TEAMS-RT systems [1] in UH-60 Helicopter, etc.

To enable the model based diagnosis, the system should be diagnosable. Simply speaking, a system is *diagnosable* if the occurrence of a failure event can be determined no matter what the behaviour of the system is. Although the diagnosability is essential for a critical system, the designer of the system usually needs to balance on the cost (e.g., the price of sensors and the human resource to check the correctness of a complex design) and the benefit (e.g., the probability of detecting a failure event) on the number of sensors to be deployed. Therefore, it is meaningful to enable the automatic verification of diagnosability.

In [17], a diagnosability notion, called L-diagnosability in the paper, is defined on nondeterministic systems. Intuitively, it says that once a failure event occurs, it is unavoidable that the failure will be detected in the future. In the paper, we will give this notion an alternative characterisation as a formula of the logic of knowledge and time [20] which expresses the knowledge of an outside observer who observes the partial information emitted by the sensors: once a failure event occurs, the observer will eventually *know* it. At the same time, the system M is transformed to another one M'. The observer's knowledge is based on the *perfect recall* view on its own observation history. There exists another characterisation [15] that bases observer's knowledge on its current observation and requires the detection to be successful without any delay. Perfect recall makes the optimal assumption on the ability of the observer and is more suitable.

The system model we are concerning about is concurrent probabilistic systems. In every system state, there exist a nonempty set of legal events. After one of them is nondeterministically taken, a probabilistic distribution is imposed to decide the next system state. In such a system, the L-diagnosability may be generalised by taking into consideration the probabilistic information. The main intuition is that the definition can be relaxed from the aspects of either the objective aspect (the system's behaviour) or the subjective aspect (the observer's evaluation on the system's behaviour).

P-diagnosability of this paper relaxes from the L-diagnosability the *sureness* of the observer's knowledge about the occurrence of failure event to the *almost sureness*. In other word, the observer needs only almost certain knowledge (i.e., knows with probability 1) to conclude the occurrence of the failure event. On the other hand, A-diagnosability and AA-diagnosability, adapted from [19] on fully probabilistic systems, can be regarded as relaxing from the L-diagnosability and P-diagnosability the *sureness* about the occurrence of failure events to the *almost sureness*. Simply speaking, they only care about those system behaviour that has a probability more than 0. The above three notions are characterised by formulas of a logic that combines knowledge, time, and probability, whose semantics will be defined in the paper by extending the one for fully-probabilistic systems [8].

In the second part of the paper, we move towards solving the computational complexity of verifying these diagnosability notions. For nondeterministic systems, a polynomial time algorithm is presented in [10] for the L-diagnosability. This PTIME upper bound is then improved in [16], which shows that the problem is NL-

*Research supported by Australian Research Council Discovery Grants DP1097203 and DP120102489.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013)*, Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.

complete. Our first complexity result shows that the verification of the L-diagnosability is also NL-complete in concurrent probabilistic systems.

Regarding to the verification of A-diagnosability, a polynomial algorithm [11] is recently presented. But until now, it is still open on the theoretical computational complexity: can this PTIME upper bound be lowered to NL as the L-diagnosability does? The second complexity result of the paper gives a *negative* answer to this open question and shows that the problem is PTIME-complete. A new polynomial time algorithm, simpler than the one in [11], is presented to reduce the verification problem to the problem of finding *end components*. The later one is then reduced to the almost sure probabilistic reachability problem, to which the monotone circuit value problem is reduced. The circuit value problem is a known PTIME-complete problem. The PTIME-completeness result suggests that, unlike the L-diagnosability, the verification of A-diagnosability is "inherently sequential" and not able to be efficiently improved by parallel algorithms, unless PTIME = NL.

The third complexity result of the paper shows that the verification of P-diagnosability is in PSPACE. To obtain this result, we first show that the general model checking problem on the logic of knowledge, time and probability is undecidable, by a reduction from the value one problem of probabilistic finite automata, which is undecidable [5]. Fortunately, the verification of P-diagnosability is strictly simpler because the system M' has a special structure that makes the verification workable via a subset construction, which pushes the complexity bound down to PSPACE.

2. CONCURRENT PROBABILISTIC SYSTEMS AND L-DIAGNOSABILITY

Let $Prop$ be a set of atomic propositions. A concurrent probabilistic system is a tuple $M = (S, s_{ini}, E, \mu, \pi)$, where S is a finite set of states, $s_{ini} \in S$ is the initial state, E is a finite set of events, $\pi : S \to \mathcal{P}(Prop)$ is a labeling function, and $\mu : S \times E \times S \to [0..1]$ is a probability transition relation such that

- $\sum_{s' \in S} \mu(s, e, s') \in \{0, 1\}$ for all $s \in S$ and $e \in E$, and
- $\exists e \in E \exists s' \in S : \mu(s, e, s') > 0$, for all $s \in S$.

Let $f \in E$ be a failure event. All failure events are simplified as a single dedicated event, which means that we are dealing with failure detection problem, instead of failure identification problem.

Let $s, s' \in S$ and $e \in E$. A *path* ρ from a state s is a finite or infinite sequence of states and events $s_0 e_0 s_1 e_1 \ldots$ such that $s_0 = s$ and $\mu(s_k, e_k, s_{k+1}) > 0$ for all k such that $k < |\rho| - 1$, where $|\rho|$ is the total number of states on ρ. Given a path ρ, we use $s(\rho, m)$ to denote its $(m+1)$-th state and $e(\rho, m)$ to denote its m-th event. Moreover, we use $s(\rho, 0..m)$ to denote the sequence of states $s(\rho, 0)...s(\rho, m)$, $e(\rho, 1..m)$ to denote the sequence of events $e(\rho, 1)...e(\rho, m)$. A *fullpath* from a state s is an infinite path from s. A path ρ is *initialised* if $s(\rho, 0) = s_{ini}$.

Let F be the set of finite paths and R be the set of fullpaths. For the diagnosis purpose, we may deploy a set of sensors in the system to detect the occurrence of the events. Assume that $O \subseteq E$ is a set of events that are observable. We let $f \notin O$, because otherwise the failure can be detected directly by observations. Given a path ρ, we use $E(\rho)$ to denote the sequence of events that occur in that path and $O(\rho)$ the sequence of observable events that occur in that path. Formally, we let $E(se\rho) = eE(\rho)$, and

$$O(se\rho) = \begin{cases} \perp O(\rho) & \text{if } e \notin O \\ eO(\rho) & \text{otherwise} \end{cases}$$

where the symbol $\perp$ denotes the occurrence of an unobservable event. We make one more assumption on the system M by requiring a fairness constraint on the fullpaths about unobservable events, i.e., the system is not allowed to have cycles of unobservable events. Formally, $\exists n \in \mathbb{N} : \forall \rho_1\rho_2\rho_3 \in R : E(\rho_2) \in (E \setminus O)^* \Rightarrow |\rho_2| < n$.

A *probability space* is a triple (W, J, μ) such that W is a set, called the *carrier*, $J \subseteq \mathcal{P}(W)$ is a set of *measurable* sets in $\mathcal{P}(W)$, closed under countable union and complementation, and $\mu : J \to [0, 1]$ is a *probability measure*, such that $\mu(W) = 1$ and $\mu(U \cup V) = \mu(U) + \mu(V)$ if $U \cap V = \emptyset$. As usual, we define the conditional probability $\mu(U|V) = \mu(U \cap V)/\mu(V)$ when $\mu(V) \neq 0$.

Let R_s (resp. F_s) be the set of fullpaths (finite paths) that start from state s. More specifically, $R_{s_{ini}}$ (resp. $F_{s_{ini}}$) is the set of initialised fullpaths (finite paths). Let $\rho|_m$ be the prefix of $\rho \in R$ up to time m. For any $\rho \in F$, we let

$$R_s(\rho) = \{\rho' \in R_s \mid \rho'|_k = \rho,\ k = |\rho| - 1\}$$

be the set of fullpaths in R_s that have ρ as the prefix. We now define probability spaces on R_s, using a well-known construction (e.g., that of [21]). Note that, R_s is not a measurable set, because of the nondeterminism of events from the states. To have a measurable set of runs, a scheduler σ is needed to resolve the nondeterminism. Given an initialised finite path ρ, a scheduler σ maps ρ to an event e such that e is a legal event of the last state in ρ, i.e., $\sigma(\rho) = e$ implies $\exists s' : \mu(s(\rho, |\rho| - 1), e, s') > 0$.

Given a state s and a scheduler σ, the set R_s^σ is the set of fullpaths in R_s such that all nondeterminism are resolved by σ. (One may view this as a *cone* of fullpaths sharing the same prefix under the scheduler σ.) For any $\rho \in F$, we let

$$R_s^\sigma(\rho) = R_s(\rho) \cap R_s^\sigma.$$

Let J_s^σ be the minimal algebra with basis the sets $W_s^\sigma = \bigcup\{R_s^\sigma(\rho) \mid \rho \text{ prefixes some } r \in R_s\}$, i.e., J_s^σ is the set of all sets of fullpaths that can be constructed from the basis by using countable union and complement. We define the measure μ_s^σ on the basis sets by

$$\mu_s^\sigma(R_s^\sigma(\rho)) = \prod_{i=0}^{|\rho|-2} \mu(s(\rho, i), e(\rho, i+1), s(\rho, i+1)), \text{ if } R_s^\sigma(\rho) \neq \emptyset,$$

and $\mu_s^\sigma(R_s^\sigma(\rho)) = 0$, otherwise. There is a unique extension of μ_s^σ that satisfies the constraints on probability measures (i.e., countable additivity and universality), and we also denote it by μ_s^σ.

PROPOSITION 1. *Given a system M, a scheduler σ, and a state s, the triple $(W_s^\sigma, J_s^\sigma, \mu_s^\sigma)$ defines a probability space.*

In such a system M, the failure diagnosis is to determine if the failure event has occurred, given the observable projections of the sequences of events generated by M. Given a finite path ρ, we let

$$F[\rho] = \{\rho' \in F \mid \rho\rho' \in F\}$$

be the set of finite paths that serve as the continuations of ρ, and

$$R_{s_{ini}}\{\rho\} = \{\rho_1\rho_2 \in R_{s_{ini}} \mid \rho_1 \in F,\ O(\rho_1) = O(\rho)\}$$

be the set of initialised fullpaths with the same observation prefix as that of ρ. Moreover, we let

$$X = \{\rho \in F_{s_{ini}} \mid e(\rho, |\rho| - 1) = f,\ \forall 1 \leq j < |\rho| - 1 : e(\rho, j) \neq f\}$$

be the set of initialised finite paths leading to the failure event, and

$$Y = \{\rho \in R_{s_{ini}} \mid \exists j \geq 1 : e(\rho, j) = f\}$$

be the set of initialised fullpaths that containing the failure event. The following is the definition of a diagnosability notion, adapted from [17] for nondeterministic systems.

DEFINITION 1. ***(L-Diagnosability)*** *A system M is L-diagnosable if and only if*

$$\exists n \in \mathbb{N} \forall \rho \in X \forall \rho' \in F[\rho] : |\rho'| \geq n \Rightarrow D(\rho\rho') = 1 \qquad (1)$$

where $D(\rho) = 1$ *if* $\forall \rho' \in R_{s_{ini}}\{\rho\} : \rho' \in Y$.

Intuitively, a system is L-diagnosability if no failure events can avoid detection: for every continuation of every finite path that contains the event f, it is impossible to find an initialised fullpath that has consistent observations and does not include the event f.

3. A LOGIC OF KNOWLEDGE, TIME, AND PROBABILITY

In this section, we introduce a logic PLTLK$_n$ that combines the temporal operator, the knowledge operator, and the probability measure. Let Agt be a set of agents. Its syntax is given by

$$\phi \ ::= \ p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid X\phi \mid \phi_1 U \phi_2 \mid K_i\phi \mid \mathtt{Pr}_i^{\bowtie d}\phi$$

where $p \in Prop$, $i \in Agt$, $\bowtie \in \{\leq, <, >, \geq\}$ is a relation symbol, and $d \in [0, 1]$ is a rational constant. Intuitively, formula $X\phi$ expresses that ϕ holds at the next time, $\phi_1 U \phi_2$ expresses that ϕ_1 holds until ϕ_2 becomes true, $K_i\phi$ expresses that the agent i knows the fact ϕ, and $\mathtt{Pr}_i^{\bowtie d}\phi$ expresses that the agent i knows the fact ϕ with a probability in relation $\bowtie$ with constant d. Other operators can be obtained in the usual way, e.g., $F\phi \equiv true U \phi$, $G\phi \equiv \neg F \neg \phi$, etc. Its semantics extends the one for fully-probabilistic systems [8].

Time is represented discretely by using natural numbers. A *run* is a function $r : \mathbb{N} \to S \times E \times L_1 \times \ldots \times L_n$ from time to the system states, events, and local states of agents. A pair (r, m) consisting of a run r and time m is called a *point*, which may also be written as $r(m)$. If $r(m) = (s, e, s_1, \ldots, s_n)$ then we define $s(r, m) = s$, $e(r, m) = e$, and $s_i(r, m) = s_i$ for $i \in Agt$. If r is a run and m a time, we write $s(r, 0..m)$ for the sequence $s(r, 0) \ldots s(r, m)$, and $e(r, 0..m)$ for $e(r, 0) \ldots e(r, m)$.

From each initialised fullpath ρ of a system M, one may define a run $\rho^{\mathtt{pr}}$ by defining each point $(\rho^{\mathtt{pr}}, m)$ with $m \in \mathbb{N}$ as follows: $s(\rho^{\mathtt{pr}}, m) = s(\rho, m)$, $e(\rho^{\mathtt{pr}}, m) = e(\rho, m)$, and $s_i(\rho^{\mathtt{pr}}, m) = Obs_i(\rho|_m)$, where Obs_i is the observation projection of agent i on the paths. Intuitively, $s_i(\rho^{\mathtt{pr}}, m)$ represents that the agent i remembers all its observations up to time m. Here the $\mathtt{pr}$ represents *perfect recall* [3].

Let a system $\mathcal{R}$ be a set of runs, and we call $\mathcal{R} \times \mathbb{N}$ the *set of points* of $\mathcal{R}$. Relative to a system $\mathcal{R}$, we define the set $\mathcal{K}_i(r, m) = \{(r', m') \in \mathcal{R} \times \mathbb{N} \mid s_i(r', m') = s_i(r, m)\}$ to be the set of points that are, for player i, indistinguishable from the point (r, m).

For a system $\mathcal{R}$ of runs, we define a *cell* c to be a subset of runs such that $\mathcal{R}_c \subseteq \mathcal{R}$. $\mathcal{R}_c$ corresponds with the set of runs that are *compatible* with a scheduler σ_c. The set of indistinguishable points for agent i in (r, m) assuming c is $\mathcal{K}_i^c(r, m) = \mathcal{K}_i(r, m) \cap \{(r', m') \mid r' \in \mathcal{R}_c, m' \in \mathbb{N}\}$.

Let $\mathcal{R}(U) = \{r \in \mathcal{R} \mid \exists m : (r, m) \in U\}$ be the set of runs in $\mathcal{R}$ going through some point in the set $U \subseteq \mathcal{R} \times \mathbb{N}$. The probability information over c is $\mathbf{P}^c = \{PR_i^c \mid i \in Agt\}$, where PR_i^c is a function mapping each point (r, m) to a probability space $PR_i^c(r, m) = (\mathcal{K}_i^c(r, m), F_i^c(r, m), \mu_{r,m,i}^c)$ such that $F_i^c(r, m) \subseteq \mathcal{P}(\mathcal{K}_i^c(r, m))$ and for $U \in \mathcal{P}(\mathcal{K}_i^c(r, m))$,

$$\mu_{r,m,i}^c(U) = \mu_c(\mathcal{R}(U) \mid \mathcal{R}(\mathcal{K}_i^c(r, m))).$$

where

$$\mu_c(\mathcal{R}_c(\rho)) = \mu_{s_{ini}}^{\sigma_c}(R_{s_{ini}}^{\sigma_c}(\rho))$$

and $\mathcal{R}_c(\rho) = \{r \in \mathcal{R}_c \mid s(r, 0..k) = s(\rho, 0..k), e(r, 1..k) = e(\rho, 1..k)\}$ for $k = |\rho| - 1$. Intuitively, at each point, each player has a probability space in which the carrier is the set of points $\mathcal{K}_i^c(r, m)$.

A *probabilistic interpreted system* (PIS) is a tuple $(\mathcal{R}, C, \{\mathbf{P}^c\}_{c \in C}, \pi)$, where $\mathcal{R}$ is a system of runs, C is a set of cells in $\mathcal{R}$ such that $\mathcal{R} = \bigcup\{\mathcal{R}_c \mid c \in C\}$, $\{\mathbf{P}^c\}_{c \in C}$ is a set of probability information for all cells in C, and $\pi : \mathcal{R} \times \mathbb{N} \to \mathcal{P}(Prop)$ is an interpretation.

The semantics of the language in a PIS $\mathcal{I}$ is given by interpreting formulas ϕ at points (r, m) of $\mathcal{I}$, using a satisfaction relation $\mathcal{I}, (r, m) \models \phi$, which is defined inductively as follows.

- $\mathcal{I}, (r, m) \models p$ for $p \in Prop$ if $p \in \pi(r, m)$,
- $\mathcal{I}, (r, m) \models \neg\phi$ if not $\mathcal{I}, (r, m) \models \phi$
- $\mathcal{I}, (r, m) \models \phi \wedge \psi$ if $\mathcal{I}, (r, m) \models \phi$ and $\mathcal{I}, (r, m) \models \psi$
- $\mathcal{I}, (r, m) \models X\phi$ if $\mathcal{I}, (r, m + 1) \models \phi$.
- $\mathcal{I}, (r, m) \models \phi U \psi$ if there exists $m' \geq m$ such that $\mathcal{I}, (r, m') \models \psi$ and $\mathcal{I}, (r, m'') \models \phi$ for all m'' with $m \leq m'' < m'$
- $\mathcal{I}, (r, m) \models K_i\phi$ if $\mathcal{I}, (r', m') \models \phi$ for all $(r, m) \in \mathcal{K}_i(r, m)$
- $\mathcal{I}, (r, m) \models \mathtt{Pr}_i^{\bowtie d}\phi$ if for all cells $c \in [\simeq_i]_C$,
  - either $\mathcal{K}_i^c(r, m) = \emptyset$,
  - or for all $(r', m') \in \mathcal{K}_i^c(r, m)$, we have

$$\mu_{r',m',i}^c(\{(r'', m'') \mid (r'', m'') \in \mathcal{K}_i^c(r', m') \wedge \mathcal{I}, (r'', m'') \models \phi\}) \bowtie d.$$

Intuitively, the knowledge formula $K_i\phi$ is satisfiable if ϕ holds on all indistinguishable points of agent i, and the probabilistic knowledge formula $\mathtt{Pr}_i^{\bowtie d}\phi$ is satisfiable if in all consistent cells, the conditional probability of ϕ being true, given the indistinguishable points of agent i, is in a relation $\bowtie$ to the constant d.

The system M gives us an interpretation π on its states, and we may lift this to an interpretation on the points (r, m) of $\mathcal{R}$ by defining $\pi(r, m) = \pi(s(r, m))$. Using the construction above, we then obtain the probabilistic interpreted system $\mathcal{I}(M) = \mathcal{I}(\mathcal{R}, C, \{\mathbf{P}^c(M)\}_{c \in C}, \pi)$. We will be interested in the problem of model checking formulas in this system. A formula ϕ is said to hold in M, written $M \models \phi$, if $\mathcal{I}(M), (r, 0) \models \phi$ for all $r \in \mathcal{R}$. The model checking problem is then to determine, given a concurrent probabilistic system M and a formula ϕ, whether $M \models \phi$.

We should note that $K_i\phi$ (sure knowledge) is not equivalent to $\mathtt{Pr}_i^{\geq 1}\phi$ (almost sure knowledge, or knows with probability 1).

In the following, we will show that the L-diagnosability of a concurrent probabilistic system M can be redefined as a verification problem $M' \models \phi_L$. The system $M' = (S', s'_{ini}, E, \mu', \pi')$ is defined as

- $S' = S \times \{s_f, s_{\neg f}\}$, $s'_{ini} = (s_{ini}, s_{\neg f})$,
- $\mu'((s, s_1), e, (s', s'_1)) = \mu(s, e, s')$, such that $s'_1 = s_f$ if $s_1 = s_{\neg f}$ and $e = f$, and $s'_1 = s_1$ otherwise, and
- $p_f \in \pi'((s, s_f))$, $p_f \notin \pi'((s, s_{\neg f}))$, and for all $p \in Prop$, $p \in \pi'((s, s_1))$ iff $p \in \pi(s)$.

Intuitively, in M', once p_f turns $True$, it will stay $True$ forever. Note that, the size of M' is quadratic over the size of M. We define two agents (outside observers) on the system M':

- Let $\mathtt{A}$ be the agent that can observe every event (including the failure event f) occurring in the system, i.e., $Obs_{\mathtt{A}} = E$. The view of $\mathtt{A}$ reflects the objective aspect of the system.
- Let $\mathtt{B}$ be the agent that can observe the occurrence of events in O, i.e., $Obs_{\mathtt{B}} = O$.

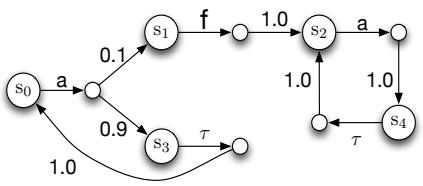

Figure 1: non-L, P, non-A, AA-diagnosable

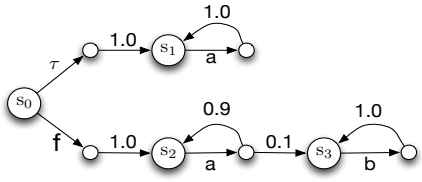

Figure 2: non-L, non-P, A, AA-diagnosable

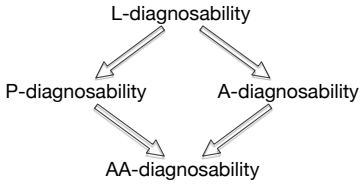

Figure 3: Relations

Note that the definitions of projection functions E and O suggest that the observers can observe the time.

THEOREM 1. *Let M be a concurrent probabilistic system. The following three statements are equivalent:*

1. *M is L-diagnosable*
2. $M' \models G(p_f \Rightarrow F\ K_{\mathtt{B}}\ p_f)$
3. $M' \models G(p_f \Rightarrow K_{\mathtt{A}}\ F\ K_{\mathtt{B}}\ p_f)$

4. DIAGNOSABILITY NOTIONS IN CONCURRENT PROBABILISTIC SYSTEMS

Theorem 1 gives the L-diagnosability two intuitive characterisations from the *subjective* views of the outside observers B and A. However, when working with concurrent probabilistic systems, the 0-1 nature of L-diagnosability may classify some reasonable systems as non-diagnosable. In this section, we investigate several proposals on the diagnosability in concurrent probabilistic systems and discuss their relations.

First, consider the system as displayed in Figure 1. We use τ to denote an unobservable but not failure event, i.e., $\tau \in E \setminus O$ and $\tau \neq f$. From the initial state s_0, the occurrence of failure event f in the next step has probability 0.1 and the occurrence of unobservable event τ has probability 0.9. After that, the failure path will be trapped into an infinite loop and the non-failure path will return to the initial state. This system is not L-diagnosable, as the agent B can never have the sure knowledge that the failure event has occurred: there exists a fullpath $(s_0 a s_3 \tau)^*$, containing no failure event, that is indistinguishable for the agent B with the fullpath $\rho = s_0 a s_1 f (s_2 a s_4 \tau)^*$, containing a failure event. On the other hand, the observer B has the almost sure knowledge about the failure, because $\lim_{m\to\infty} \mu_{r,m,\mathtt{B}}(\{(r', m) \mid p_f \in \pi(r', m),\ O(r') = O(r)\}) \geq 1$ for any run r such that $O(r) = (a\perp)^*$.

Based on this observation, we introduce a new notion named P-diagnosability. Intuitively, for a system to be P-diagnosable, the observer need not have sure knowledge about the occurrence of failure event. Instead, the observer is requested to almost surely know the occurrence of failure event.

DEFINITION 2. ***(P-Diagnosability)*** *A system M is P-diagnosable if and only if*

$$M' \models G(p_f \Rightarrow F\ \mathtt{Pr}_{\mathtt{B}}^{\geq 1} p_f)$$

As the L-diagnosability, the above statement can be rewritten as $M' \models G(p_f \Rightarrow K_{\mathtt{A}}\ F\ \mathtt{Pr}_{\mathtt{B}}^{\geq 1} p_f)$ by considering the knowledge of the agent A. It is not difficult to check that the system of Figure 1 is P-diagnosable.

Now we move to the definitions of other two notions that relaxes the sure knowledge of agent A from L-diagnosability and P-diagnosability. The motivation can be seen from the system in Figure 2. After the failure event f occurs in a finite path $\rho = s_0 f s_2$, there exists an infinite sequence of events $a...a$ such that the path $\rho' = s_2 (a s_2)^*$ is possible to avoid the detection, i.e., there exists a fullpath $\rho_2 = s_0 \tau s_1 (a s_1)^*$ such that $O(\rho_2) = O(\rho\rho') = \perp a^*$ and $f \notin \rho_2$. Furthermore, the observer B can not almost surely know (and thus can not know) the occurrence of the failure event, because $\lim_{m\to\infty} \mu_{r,m,\mathtt{B}}(\{(r', m) \mid p_f \in \pi(r', m),\ O(r') = O(r)\}) = 0$ for any run r such that $O(r) = \perp a^*$.

However, we notice that the fullpath ρ' has the occurrence probability of 0. In other word, for the system in Figure 2, the agent A has the almost sure knowledge, instead of the sure knowledge, about the detectability of failure event by the observer B.

Therefore, we have the following two notions, which have the same meanings as those from [19] for fully probabilistic systems.

DEFINITION 3. ***(A-Diagnosability)*** *A system M is A-diagnosable if and only if*

$$M' \models G(p_f \Rightarrow \mathtt{Pr}_{\mathtt{A}}^{\geq 1}\ (F\ K_{\mathtt{B}}\ p_f)).$$

Intuitively, a system is A-diagnosable if once the failure event occurs, the agent A has almost sure knowledge that in the future the agent B will know the occurrence of failure event. In other word, the A-diagnosability relaxes from the L-diagnosability the "sureness" on the agent A's knowledge.

DEFINITION 4. ***(AA-Diagnosability)*** *A system M is AA-diagnosable if and only if*

$$M' \models G(p_f \Rightarrow \mathtt{Pr}_{\mathtt{A}}^{\geq 1}\ (F\ \mathtt{Pr}_{\mathtt{B}}^{\geq 1} p_f)).$$

Intuitively, a system is AA-diagnosable if once a failure event occurs, the agent A almost surely knows that in the future the agent B can almost surely know the occurrence of failure event. In other word, the AA-diagnosability relaxes from the P-diagnosability the "sureness" of the agent A's knowledge.

Before proceeding, let's see the relations between the four notions. Let D_X be the set of systems satisfying the X-diagnosability for $X \in \{L, P, A, AA\}$. We say that a diagnosability notion X is stronger than the other one Y, denoted as $X \geq Y$, if for all probabilistic systems M, we have that $M \in D_X$ implies $M \in D_Y$. Furthermore, X is strictly stronger than Y, denoted as $X > Y$, if $X \geq Y$ and there exists a system M such that $M \in D_X$ and $M \notin D_Y$. We write $X <> Y$ if neither $X \geq Y$ nor $Y \geq X$.

The relations between the diagnosability notions are shown in Figure 3, where the arrows denote the "strictly stronger than" relation between the tail notion and the head notion.

THEOREM 2. *$L > A > AA$, $L > P > AA$, and $A <> P$.*

The fact that $L \geq A \geq AA$ and $L \geq P \geq AA$ can be seen from their definitions. The strictness of them can be seen by the examples in Figure 1 and Figure 2.

5. COMPLEXITY OF VERIFYING DIAGNOSABILITY NOTIONS

Given a system M and a notion X-diagnosability, the verification problem of the diagnosability is to decide whether M is X-diagnosable. Logic characterisations of the diagnosability notions make it possible for us to borrow results from the area of model checking. E.g., the formula (2) of Theorem 1 for the L-diagnosability falls within the single player fragment of the temporal epistemic logic LTLK$_n$, whose verification complexity is PSPACE-complete [2]. Therefore, the verification of L-diagnosability is in PSPACE.

However, the complexity bound obtained in this way is not tight. In this section, we will give complexity results for three notions (L-, P-, and A-diagnosability). The results and proofs can be interesting by their own.

- The NL-complete of L-diagnosability shows that the above-mentioned PSPACE-complete complexity for the verification of single-player fragment of LTLK$_n$ logic can be lowered if we consider a smaller fragment, in this case the positive fragment with a single knowledge operator, and the model complexity (that is, the complexity is measured by the size of system model, by fixing the formula).
- The proof of A-diagnosability can be adapted to show that the model complexity, and thus the combined complexity (measured by both the size of system model and the size of the formula), of the verification of PCTL logic on MDPs [6] has PTIME as its lower bound, which matches its current upper bound. This result is not unknown but does not appear in the literature.
- Before giving a PSPACE algorithm for the P-diagnosability, we will show that the verification of the PLTLK$_n$ logic under perfect recall interpretation is undecidable in general and for its single player fragment.

5.1 L-Diagnosability

On verifying the L-diagnosability in nondeterministic systems, [10] gives a polynomial time algorithm and [16] shows that the problem is NL-complete (i.e., can be solved by a nondeterministic Turing machine using a logarithmic amount of memory space). We here show that the NL-completeness holds also in concurrent probabilistic systems. The complexity class NL is included in the class PTIME, but it is still open on the strictness.

The membership problem.

We reduce the verification problem on the system M to another problem on *the twin-plant* G. The twin plant G of a concurrent probabilistic system $M = (S, s_{ini}, E, \mu, \pi)$ and a set of observable events $O \subseteq E$ is a tuple $(S^G, s^G_{ini}, E^G, \mu^G, \pi^G)$ such that

- $S^G = S \times S$, $s^G_{ini} = (s_{ini}, s_{ini})$, $E^G = E \times E$,
- $\mu^G((s_1, s_2), (e_1, e_2), (s'_1, s'_2)) = (\mu(s_1, e_1, s'_1), \mu(s_2, e_2, s'_2))$, if
  - $e_1 = e_2 \in O$ or $\{e_1, e_2\} \cap O = \emptyset$, and
  - $\mu(s_1, e_1, s'_1) > 0$ and $\mu(s_2, e_2, s'_2) > 0$,
- $\pi^G((s_1, s_2)) = (\pi(s_1), \pi(s_2))$.

Intuitively, G is the co-simulation of two copies of the original system M such that the observable events are executed *synchronously*. Furthermore, we let Π_i be the function mapping a pair (x, y) to its ith element for $i \in \{1, 2\}$, and let

$$\mu^G_i(s, e, s') = \Pi_i(\mu^G(s, e, s')),\ \pi^G_i(s) = \Pi_i(\pi^G(s))$$

and $G_i = (S^G, s^G_{ini}, E^G, \mu^G_i, \pi^G_i)$ for $i \in \{1, 2\}$. The following is a direct observation from the definition that every path in the twin plant G represents two paths in the original system that have the same observation, and vice versa.

PROPOSITION 2. *Let M be a probabilistic system and G the twin plant of M. For all paths $\rho^G \in \mathcal{R}^G$, we have that $O(\Pi_1(\rho^G)) = O(\Pi_2(\rho^G))$. Also, for every two paths in M such that $O(\rho_1) = O(\rho_2)$, there exists a path ρ^G such that $\Pi_1(\rho^G) = \rho_1$ and $\Pi_2(\rho^G) = \rho_2$.*

An event $e^G = (e_1, e_2)$ of the twin plant G is a *faulty event* if $e_1 = f$, and is an *evidential event* if $e_2 = f$. A path ρ^G is a *faulty path* if it contains a faulty event, and is an *evidential path* if it contains an evidential event. As usual, the system size is measured by the number of nodes plus the number of events, that is, $|M| = |S| + |E|$. The number of transitions in M is $O(|S|^2|E|)$. The size of G is $O(|S|^2 + |E|^2)$ and the number of transitions in G is $O(|S|^4|E|^2)$.

A finite path $\rho^G = (s^1_0, s^1_0)...(s^1_k, s^2_k)...(s^1_m, s^2_m)$ of the twin plant G is *undiagnosable* if

- $(s^1_m, s^2_m) = (s^1_k, s^2_k)$ (cycle path),
- $\exists 1 \le i \le k : f = \Pi_1(e(\rho^G, i))$ (faulty event occurs before the loop), and
- $\forall 1 \le i \le m : f \ne \Pi_2(e(\rho^G, i))$ (non-evidential path).

Intuitively, together with Proposition 2, the existence of an undiagnosable path in G indicates that, in the original system M, a fullpath $\Pi_1(\rho^G)$ that contains the failure event can be simulated by another one $\Pi_2(\rho^G)$ that does not, without being distinguished by the agent B.

Then we have that the system M is L-diagnosability if and only if there exists no initialised undiagnosable path in the twin plant G. To verify the later condition, we do a transformation on G by removing all evidential events and their related transitions. Let G' be the obtained twin plant. The size of G' is linear with respect to the size of G. We say that a state t^G is reachable from another state s^G if there exist states $s^G_1, ..., s^G_k$ such that $s = s^G_1$, $t = s^G_k$, and $\exists e_i \in E^G : \mu^G_1(s_i, e_i, s_{i+1}) > 0$ for all $1 \le i \le k - 1$.

The L-diagnosability of the original system M can be decided by the following algorithm.

ALGORITHM 1. *Let G be the twin plant of the system M and G' be the one obtained by removing all evidential events from G. We check on G' the existence of a faulty event e^G and three states s^G_1, s^G_2, and s^G, such that*

1. *$\mu^G_1(s^G_1, e^G, s^G_2) > 0$,*
2. *the reachability from s^G_{ini} to s^G_1,*
3. *the reachability from s^G_2 to s^G,*
4. *the reachability from s^G to itself.*

The algorithm returns False if the existence holds, and return True, otherwise.

The membership of the verification of L-diagnosability in NL complexity class can be done by a similar procedure as the one in the Savitch's theorem. Any of the above reachability problem can be solved by a nondeterministic algorithm of logarithmic space with respect to the twin plant G'. The existence of faulty event e^G and states s^G_1, s^G_2, and s^G can be done by a nondeterministic algorithm of constant space. In total, the Algorithm 1 can be concretised as a nondeterministic algorithm of logarithmic space with respect to G', which, transformed into the measurement of the original system M, needs logarithmic space.

The hardness problem.

The NL-hardness can be seen by a reduction from st-connectivity problem, a well-known NL-complete problem. The st-connectivity problem determines if t is reachable from s, given that s and t are vertices of a directed graph $G(V, Ed)$. A logarithmic space reduction proceeds by transforming G into a system $M = (S, s_{ini}, E, \mu)$, such that $S = V$, $s_{ini} = s$, $E = \{o, \tau, f\}$, representing an observable event, an unobservable event, and a failure event, respectively, and $\mu(u, o, v) = 1$ iff $Ed(u, v)$ for all $u, v \in S$, and $\mu(t, \tau, u) = \mu(t, f, u) = 1$ for some $u \in S$. Then the reachability from s to t in G is equivalent to the non-L-diagnosability of M.

Put them together, we have the following conclusion.

THEOREM 3. *Verification of L-diagnosability is NL-complete.*

5.2 A-Diagnosability

For the verification of A-diagnosability, an algorithm is recently presented in [11] without a formal proof on its computational complexity. In this section, we will show that this verification problem is PTIME-complete (i.e., can be solved by a deterministic Turing machine using a polynomial amount of computation time), which indicates that the problem, unlike the L-diagnosability, can not be solved by using only polylogarithmic space and does not admit efficient parallel algorithms: the problem is inherently sequential and requires storing a polynomial number of intermediary results, unless NL=PTIME.

The membership problem.

A probabilistic directed graph is a tuple $G = (V, E, \mu)$ such that V is a set of vertices, E is a set of edges, and $\mu : V \times E \times V \to [0, 1]$ is a probabilistic transition. A directed graph is strongly connected if there is a path from each vertex in the graph to every other vertex. The strongly connected components (SCCs) of a directed graph are its maximal strongly connected subgraphs. Let $\delta : U \to E$ be a function mapping vertices to the edges. A set $U \subseteq V$ of vertices are δ-closed, if for all $s \in U$, $\mu(s, \delta(s), t) > 0$ implies $t \in U$. U is an δ-end component if U is δ-closed and the underlying graph of U and δ is an SCC. U is an end component if there exists a function δ such that U is an δ-end component.

Recall that G_1 is obtained from the twin plant G by taking the probabilistic transition relation and labelling function of the first component. Let G_1' be the one obtained from G_1 by removing all evidential events and their related transitions. A δ-end component in G_1' means that 1) from any of its state, all the future states will remain in the end component by following the function δ, and 2) any fullpath from any of its states suggests the undetectability of a failure event (if any) by the agent B since then.

Then by definition, the system M is A-diagnosability if and only if there exists no faulty path in G_1' that leads to an end component. The existence of such a path is equivalent to the non-zero probability of avoiding detection. Therefore, the A-diagnosability of a system M can be decided by the following algorithm.

ALGORITHM 2. *Let G be the twin plant of the system M and G_1' as defined before. The algorithm proceeds by doing the following steps sequentially on G_1':*

1. *find all end components,*
2. *find a reachable faulty state such that it can reach a state of any end component.*

The algorithm returns False if the second step succeeds, and return True, otherwise.

The finding of all end components can be solved in PTIME, by taking a variant of Tarjan's algorithm [18]. The finding of all reachable faulty state can be solved in PTIME by enumerating all faulty states and checking the reachability from the initial state. The reachability from a faulty state to the states of end components can be decided by enumerating all states in end components and then checking the reachability between two states. Because st-connectivity problem is in PTIME, this algorithm is also in PTIME.

Before proceeding, we make a remark that, the above algorithm is substantially different with the one presented in [11], which instead of utilising the end components, checks the existence of two kinds of faulty cycles in the twin plant: α-recurrent faulty cycle and m-total faulty cycle. Simply speaking, the α-recurrent faulty cycle is a cycle of probability 1, and the m-total faulty cycle is a set of cycles with the total probability 1. Their claim of polynomial time in deciding the m-total faulty cycle is dubious. A subset construction seems unavoidable, which will push the upper bound to PSAPCE.

The hardness problem.

Let s be a state and U be a sets of states. We use $\rho \prec R_s^\sigma$ to denote that ρ is a finite path that starts from state s and is consistent with the scheduler σ, or formally $s(\rho, 0) = s$ and $e(\rho, m+1) = \sigma(\rho|_m)$ for all $0 \le m \le |\rho| - 2$. We define $reach(s, U, \sigma) = \{\rho \prec R_s^\sigma \mid s(\rho, k) \in U, \forall 0 \le m < k : s(\rho, m) \notin U, k = |\rho| - 1\}$ to be the set of finite paths that reach from state s to states in U under the scheduler σ. The probability of reaching from s to U under σ is then defined as

$$probReach(s, U, \sigma) \equiv \mu_s^\sigma(reach(s, U, \sigma)).$$

We say that U is almost surely reachable from state s if there exists a scheduler σ such that $probReach(s, U, \sigma) = 1$.

It's not hard to see that the finding of end components can be reduced to the finding of all states that are in end components. In the remaining part of this section, we will prove the PTIME lower bound on deciding if a state s is in an end component. The problem can be decided by taking the following algorithm to see if $\exists\sigma : probReach(s, \{s\}, \sigma) = 1$.

ALGORITHM 3. *Let $M = (S, s_{ini}, E, \mu, \pi)$ be a concurrent probabilistic system and s, t be any two states. The following algorithm decides if $\exists\sigma : probReach(s, \{t\}, \sigma) = 1$:*

1. *nondeterministically guess a set of states U, and then*
2. *verify the following two statements:*
   (a) *$\exists\sigma : probReach(s, U, \sigma) = 1$,*
   (b) *for all states $s' \in U \setminus \{s\}$, recursively taking this algorithm to decide if $\exists\sigma : probReach(s', \{t\}, \sigma) = 1$.*

The algorithm has a recursion depth of $O(\log |S|)$ level, and needs $O(\log |S|)$ space to store the states s and t. The space can be reused in different levels. Note that, we don't need to store the set U. Therefore, it reduces the problem of deciding if a state s is in an end component to $\exists\sigma : probReach(s, U, \sigma) = 1$ by taking a logarithmic space.

Now we need only show that it is PTIME-hard to decide $\exists\sigma : probReach(s, U, \delta) = 1$. The proof is done by encoding the monotone circuit value problem, which is well-known to be PTIME-complete (cf. [4]). A monotone boolean circuit $C = (B, B_I, b_O, T, \pi)$ is a directed acyclic graph, in which B is a set of gates, $B_I \subseteq B$ is a set of input gates, b_O is a single output gate, $T : B \to \mathcal{P}(B)$ is a set of directed connections between gates such that

- $\forall b \in B_I : \{b' \in B \mid b \in T(b')\} = \emptyset$ (the indegree of input gates is zero),

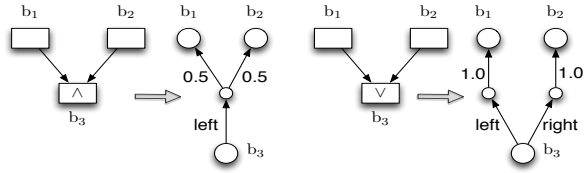

Figure 4: The transformation from boolean circuit to concurrent probabilistic system

- $T(b_O) = \emptyset$ (the outdegree of the output gate is zero), and
- $\forall b \in B \setminus B_I : |\{b' \in B \mid b \in T(b')\}| = 2$ (the indegree of non-input gates are 2).

and $\pi : (B \setminus B_I) \to \{\wedge, \vee\}$ is a labelling function mapping every non-input gate to a boolean operator $\wedge$ or $\vee$. An input of the circuit C is an assignment $\alpha : B_I \to \{0, 1\}$ of boolean values to input gates. The value of a non-input gate is computed as the result of boolean operation (the one on its label) on the values of its ancestors. The output value of the circuit $v(C, \alpha)$ is the value on the output gate. A monotone circuit value problem (MCVP) takes as input a monotone boolean circuit C and its input α, and decides if the output value $v(C, \alpha)$ is 1.

Now we will show that the MCVP problem can be reduced to the almost sure reachability problem. Let $M_C = (S, s_{ini}, E, \mu, \pi_C)$ be a concurrent probabilistic system such that $S = B$, $s_{ini} = b_O$, $E = \{left, right\}$, and

$$\mu(s, e, s') = \begin{cases} 0.5 & \text{if } \pi(s) = \wedge \text{ and } e = \text{left and } s' = lanc(s) \\ 0.5 & \text{if } \pi(s) = \wedge \text{ and } e = \text{left and } s' = ranc(s) \\ 1 & \text{if } \pi(s) = \vee \text{ and } e = \text{left and } s' = lanc(s) \\ 1 & \text{if } \pi(s) = \vee \text{ and } e = \text{right and } s' = ranc(s) \end{cases}$$

where $lanc(s)$ denotes the left ancestor of the gate s and $ranc(s)$ denotes the right ancestor of the gate s. Intuitively, we reverse the direction of the connections in circuit C and do the transformation for every gate as depicted in Figure 4. Let $S_I = \{b \mid b \in B_I,\ \alpha(b) = 1\}$ be the set of states representing the input gates of value 1. We have the following claim.

LEMMA 1. *Given a monotone circuit C and an input α, we can construct a system M_C such that, $v(C, \alpha) = 1$ if and only if $\exists \sigma : probReach(s_{ini}, S_I, \sigma) = 1$ in M_C.*

Combining the membership and hardness results, we have the following conclusion.

THEOREM 4. *Verification of A-diagnosability is PTIME-complete.*

5.3 P-Diagnosability

Now, we will show that the verification of P-diagnosability is PSPACE-complete (i.e., can be solved by a Turing machine using a polynomial amount of space). First of all, we show that the verification of PLTLK$_n$ logic is undecidable in general. Fortunately, the definition of P-diagnosability makes its verification decidable in PSPACE. The main reason is the special structure of M'.

The Verification of PLTLK$_n$ Logic is Undecidable

The undecidability result is obtained by a reduction from the value one problem of probabilistic finite automata, which is a known undecidable problem [5]. The following proof resembles the one in [9] for the PATL* logic. A probabilistic automaton PA is a tuple $\langle Q, A, (M_a)_{a \in A}, q_0, \mathcal{F} \rangle$, where Q is a finite set of states and q_0 is the initial state, $\mathcal{F} \subseteq Q$ is a set of accepting states, A is the finite input alphabet, and $(M_a)_{a \in A}$ is the set of transition matrix.

For each $a \in A$, $M_a \in [0, 1]^{Q \times Q}$ defines transition probabilities, such that given $q, q' \in Q$, $M_a(q, q')$ is the probability that q makes a transition to q' when a is the input. For every $q \in Q$ and $a \in A$, we have $\sum_{q' \in Q} M_a(q, q') = 1$. Plainly, given a state q, an input a makes a transition to a distribution on Q, and we further extend M_a to be a transformer from distributions to distributions. Given $\Delta \in \mathcal{D}(Q)$, we write $M_a(\Delta)$ for the distribution transformed from Δ by a, such that for all $q' \in Q$, $M_a(\Delta)(q') = \sum_{q \in Supp(\Delta)} \Delta(q) \cdot M_a(q, q')$. Given $w = a_1 \cdot a_2 \cdot \ldots \cdot a_n \in A^*$, we write M_w for the function $M_{a_n} \circ M_{a_{n-1}} \circ \cdots \circ M_{a_1}$ (we assume function application is right associative).

Given a probabilistic automaton PA and $\lambda \in [0, 1]$, the (strict) emptiness problem is to decide whether there exists a word w such that $M_w(q_0)(\mathcal{F}) \geq (>)\lambda$, where λ is called a cut-point. The above problem is undecidable in general [14, 13, 12], and for the case of value-one [5], i.e., $\lambda = 1$ or $\lambda = 0$.

We define a translation $\mathfrak{F}$ mapping probabilistic automata to concurrent probabilistic systems. Let $PA = \langle Q, A, (M_a)_{a \in A}, q_0, \mathcal{F} \rangle$, define $M = \mathfrak{F}(PA) = (S, s_{ini}, E, \mu, \pi)$, where

- $S = Q$, $s_{ini} = q_0$, $E = A$,
- μ is the same as that of $(M_a)_{a \in A}$, i.e., $\mu(q, a, q') = M_a(q, q')$ for all $q \in Q$, $q' \in Q$, and $a \in A$,
- $p_f \in \pi(q)$ if and only if $q \in \mathcal{F}$.

Moreover, the observable set is defined to be $O = E$. Given a probabilistic automaton PA, there exists a word w such that $M_w(q_0)(\mathcal{F}) \geq 1$ in PA iff $M \not\models G\ \mathtt{Pr}_{\mathtt{B}}^{<1} p_f$. Therefore, we have the following conclusion.

THEOREM 5. *The verification of PLTLK$_n$ logic is undecidable.*

The verification of P-diagnosability is in PSPACE

Although the undecidability is pessimistic, the verification of P-diagnosability can be strictly simpler because of the special structure of $M' = (S', s'_{ini}, E, \mu', \pi')$ that once the atomic proposition p_f holds on a state s, it will hold on all its successor states.

Let $Q \subseteq S$ be a set of states. We let

$$Q \cdot e = \{t \in S' \mid s \in Q,\ \mu'(s, e, t) > 0\}$$

for $e \in O$ be the set of successor states of Q by taking event e, and

$$Q \cdot \tau = \{t \in S' \mid s \in Q,\ \exists e \in E \setminus O : \mu'(s, e, t) > 0\}$$

be the set of successor states of Q by taking an unobservable event.

Let $M'' = (S'', s''_{ini}, E, \mu'', \pi'')$ be a system such that $S'' = \mathcal{P}(S')$, $s''_{ini} = \{s'_{ini}\}$, and $\mu''(Q, e, Q') = 1$, if $Q' = Q \cdot e$ for $e \in O \cup \{\tau\}$, and = 0, otherwise. Intuitively, the system M'' is obtained from M' by a subset construction as defined by μ''. In system M'', the fullpath $Q_0 Q_1 ...$ captures the set of fullpaths in M' whose observation projections are the same.

Let Q be a state of M''. We call Q a *normal state* if $\forall s \in Q : p_f \notin \pi(s)$, a *flaw state* if $\exists s, t \in Q : p_f \in \pi(s) \wedge p_f \notin \pi(t)$, and a *terminal state* if $\forall s \in Q : p_f \in \pi(s)$. The special structure of M' enables the following property of the system M''.

LEMMA 2. *The system M'' has and only has three kinds of end components C: 1) all states in C are normal states, 2) all states are flaw states, and 3) all states are terminal states.*

It is straightforward that if from the initial state $\{s'_{ini}\}$, it can only reach normal end components or terminal end components then the system M is P-diagnosability. The case of flaw end components is more involved. Assume that Q is a flaw state. Let $Q^n = \{s \in Q \mid p_f \notin \pi(s)\}$ be the normal part of Q. A normal end component suggests a set of fullpaths that never contain failure event. Therefore, if $Q_1 \subseteq Q^n$ is in a normal end component then we can find a subset of fullpaths starting from states in Q such that it has a non-zero probability of not containing failure event.

Therefore, the P-diagnosability of a system M can be decided by the following algorithm.

ALGORITHM 4. *Let $M = (S, s_{ini}, E, \mu, \pi)$ be a concurrent probabilistic system and M'' as defined before. The algorithm proceeds by doing the following steps sequentially on M'':*

1. *nondeterministically guess a flaw state Q, and then*
2. *verify the following two statements:*
   (a) *Q is reachable from s''_{ini},*
   (b) *there exists $\emptyset \neq Q_1 \subseteq Q^n$ such that Q_1 is in a normal end component.*

The algorithm returns False if the second step succeeds, and return True, otherwise.

As for the complexity, it's not hard to see that the system M'' is of size exponential with respect to the system M' but can be constructed on-the-fly. We can take a nondeterministic turing machine to guess the sets Q and Q_1, and then determine if Q_1 is in a normal end component. All these can be done by using polynomial size of spaces with respec to the number of states in M'. Therefore, it is in NPSPACE=PSPACE.

THEOREM 6. *Verification of P-diagnosability is in PSPACE.*

6. CONCLUSIONS

In the paper, we investigate the diagnosability in concurrent probabilistic systems. Four diagnosability notions are characterised by formulas of a logic that combines knowledge, time, and probability. The computational complexities of verifying them are studied.

We leave open several questions related to the complexity to the future works: the lower bound of verifying P-diagnosability, and the complexity for AA-diagnosability.

Moreover, it is meaningful to develop verification tools and conduct experiments on practical systems. Especially, it is interesting to compare the performance between designated algorithms as described in the paper and general algorithms for the PLTLK$_n$ logic and its fragments, generalised from [8, 7].

7. REFERENCES

[1] C. Deb, K.R. Pattipati, V. Raghavan, M. Shakeri, and R. Shrestha. Multi-signal flow graphs: A novel approach for system testability analysis and fault diagnosis. *IEE AES Systems Magazine*, 1995.

[2] Kai Engelhardt, Peter Gammie, and Ron van der Meyden. Model Checking Knowledge and Linear Time: PSPACE Cases. In *LFCS*, volume 4514 of *LNCS*, pages 195–211. Springer, 2007.

[3] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning About Knowledge*. The MIT Press, 1995.

[4] A. Gibbons and W. Rytter. *Efficient Parallel Algorithms*. Cambridge University Press, 1988.

[5] Hugo Gimbert and Youssouf Oualhadj. Probabilistic Automata on Finite Words: Decidable and Undecidable Problems. In *ICALP (2)*, pages 527–538, 2010.

[6] Hans Hansson and Bengt Jonsson. A logic for reasoning about time and reliability. *Formal Asp. Comput*, 6(5):512–535, 1994.

[7] Xiaowei Huang. Bounded planning for strategic goals with incomplete information and perfect recall. In *12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS2013)*, 2013.

[8] Xiaowei Huang, Cheng Luo, and Ron van der Meyden. Symbolic Model Checking of Probabilistic Knowledge. In *13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK XII)*, pages 177–186, 2011.

[9] Xiaowei Huang, Kaile Su, and Chenyi Zhang. Probabilistic Alternating-time Temporal Logic of Incomplete information and Synchronous Perfect Recall. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI-12)*, pages 765–771, 2012.

[10] Shengbing Jiang, Zhongdong Huang, V. Chandra, and R. Kumar. A polynomial algorithm for testing diagnosability of discrete-event systems. *Automatic Control, IEEE Transactions on*, 46(8):1318–1321, aug 2001.

[11] Minnan Luo, Fuchun Sun, and Yongming Li. A polynomial algorithm for testing diagnosability of stochastic discrete event systems. In *8th Asian Control Conference (ASCC)*, pages 1048–1053, 2011.

[12] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147:5–34, 2003.

[13] Azaria Paz. *Introduction to probabilistic automata (Computer science and applied mathematics)*. Academic Press, 1971.

[14] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[15] Franco Raimondi. *Model checking multi-agent systems*. PhD thesis, Department of Computer Science University College London University of London, 2006.

[16] Jussi Rintanen. Diagnosers and Diagnosability of Succinct Transition Systems. In *IJCAI*, pages 538–544, 2007.

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

[18] Robert Endre Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[19] David Thorsley and Demosthenis Teneketzis. Diagnosability of stochastic discrete-event systems. *IEEE Transactions on Automatic Control*, 50(4):476–492, 2005.

[20] Ron van der Meyden and Nikolay V. Shilov. Model Checking Knowledge and Time in Systems with Perfect Recall. In *Foundations of Software Technology and Theoretical Computer Science*, volume 1738, pages 432–445, 1999.

[21] Moshe Y. Vardi. Automatic Verification of Probabilistic Concurrent Finite-State Programs. In *FOCS*, pages 327–338, 1985.

[22] Brian C. Williams and P. Pandurang Nayak. A model-based approach to reactive self-configuring systems. In *Procs. AAAI-96* , pages 971–978, 1996.