Event Diagnosis of Discrete-Event Systems with Uniformly and Nonuniformly Bounded Diagnosis Delays

Tae-Sic Yoo and Humberto E. Garcia {tyoo,garcia}@anlw.anl.gov
Systems Modeling, Analysis, and Control Group, Argonne National Laboratory

***Abstract*— Various notions of diagnosability reported in literature deal with uniformly bounded finite detection or counting delays. The uniformity of delays can be relaxed while delays remain finite. We introduce various notions of diagnosability allowing nonuniformly bounded finite delays. A polynomial-time verification algorithm for diagnosability with nonuniformly bounded finite indefinite-counting delays is presented. A similar technique is applied to give a computationally better verification algorithm for diagnosability with uniformly bounded finite indefinite-counting delays than algorithms previously reported in literature. Finally we develop a new on-line diagnosis algorithm that has a lower time and space complexity than on-line diagnosis algorithms reported in literature for counting the occurrence of repeated/intermittant faults.**

I. INTRODUCTION

The objective of diagnosis is to monitor a system and infer the occurrences of special behaviors under partial observations. The special behavior may be specified by the occurrence of special events, such as faults. In this context, the problem of fault diagnosis of logical discrete-event systems has drawn considerable attention recently. For more literature references regarding fault diagnosis of logical discrete-event systems, we direct the reader to [1]. In [2], the notion of diagnosability was first introduced, which is to detect the occurrence of faults within uniformly bounded finite delays.

In [3], fault counting problems regarding repeated faults were addressed and three notions of diagnosability counting the number of the occurrence of faults were introduced. First the notion of K-diagnosability characterizes the capability of counting, with uniformly bounded finite delays, if at least K faults have occurred. Second the notion of $[1, K]$-diagnosability is related to the problem of deciding within uniformly bounded finite delays if at least J faults have occurred for all J where $1 \leq J \leq K$. Finally, the occurrence of any number of faults can be inferred within an uniformly bounded finite delays with the notion of $[1, \infty]$-diagnosability. Polynomial-time algorithms deciding the properties of diagnosability described above were also provided in [3]. Using the algorithms presented in [3], the computational complexity of verifying K- and $[1, K]$-diagnosability with uniformly bounded finite delays is $O(K^2 \cdot |Q^A|^2 \cdot |\Sigma^A|^2)$ where $|Q^A|$ is the number of states of *deterministic* automaton A describing the behavior of system and $|\Sigma^A|$ is the number of events defined over A. On the other hand, the computational complexity of verifying $[1, \infty]$-diagnosability with uniformly bounded finite delays is $O(|Q^A|^4 \cdot |\Sigma^A|^2)$ using the algorithm in [3]. These concepts were motivated by the model-based supervision of dynamic item/entitity flows network problem arising in manufacturing systems presented in [4]. However, the developed algorithms in [3] are of high degree polynomials. Therefore, the implementations of the algorithms have limited practical use.

To our best knowledge, we first note that all notions of diagnosability appeared in literature deal with uniformly bounded finite delays only. The concept of uniformly bounded delay is useful when a fixed finite maximum detection delay needs to be guaranteed. However, one may just want to know if the occurrence of faults can be detected and/or counted eventually while the uniformity of finite delays may not hold. In order to capture the nonuniformity of finite delays, we introduce various notions of diagnosability allowing nonuniformly bounded finite detection or counting delays. Being implied in the term "nonuniform", finite delays are not uniform and may depend on the trace executed by the system. We relate these notions of diagnosability allowing nonuniformly bounded finite delays with the conventional notions of diagnosability allowing only uniformly bounded finite delays. Furthermore, we devise a polynomial-time verification algorithm for $[1, \infty]$-diagnosability allowing nonuniformly bounded finite delays. Similar verification technique is applied to the verification of $[1, \infty]$-diagnosability with uniformly bounded finite delay. The new verification algorithm for $[1, \infty]$-diagnosability with uniformly bounded finite delays carries a lower worst case time and space computational complexities than those of the verification algorithms in [3]. We also provide an on-line diagnosis algorithm for counting the occurrence of faults, which has a lower time and space complexities than those of the one reported in [3]. Both verification and on-line diagnosis algorithms requires the shortest path computation. Because of the involvement of the shortest path computation, existing algorithms and heuristics regarding the shortest path computation can be readily applied [5–7]. Indeed, the theoretical computational saving of our algorithms partially comes from the application of existing efficient shortest path computation algorithms in [5, 6].

The developed verification and on-line diagnosis algorithms are successfully implemented and used for the model-based detection of routing events in discrete flow networks discussed in [4].

All proofs are omitted due to space limitations, which can be found in [1].

II. PRELIMINARIES

In this section, we define the model of discrete-event systems under consideration and related necessary notation. First we model the untimed discrete-event system as a deterministic finite-state automaton: $A = (Q^A, \Sigma^A, \delta^A, q_0^A)$ where Q^A is the finite state space, Σ^A is the set of events, and q_0^A is the initial state of the system. δ^A is the partial transition function and $\delta^A(q_1, \sigma) = q_2$ implies the existence of a transition from state q_1 to state q_2 with event label σ. The superscript A may be dropped if this is not likely to cause confusion. The language generated by A is denoted by $\mathcal{L}(A)$ and is defined in the usual manner [8].

To reflect limitations on observation, we define the observation mask function $M : \Sigma^A \rightarrow \Delta^A \cup \{\epsilon\}$ where Δ^A is the set of observed symbols and it may be disjoint with Σ^A. The definition of M can be extended to sequences of events (traces) inductively as follows: $\forall s \in (\Sigma^A)^*$, $\forall \sigma \in \Sigma^A$, $M(s\sigma) = M(s)M(\sigma)$.

We define a set of events to be diagnosed Σ_f. In order to facilitate the fault diagnosis problem of multiple type of faults, Σ_f is defined to be the set of fault events and partitioned into a set of fault types, which is denoted by Π_f with: $\Pi_f = \{\Sigma_{f_i} : \Sigma_f = \Sigma_{f_1} \dot\cup \ldots \dot\cup \Sigma_{f_n}\}$. The essence of the framework of this paper (following [2, 3]) is in events detection and classfication. Here, the events to be detected and classified are defined as faults/failures in the context of fault/failure diagnosis. However, the events of interest need not to be faults or failures but can be any special events of interests, in general.

Given a trace $s \in \mathcal{L}(A)$, we denote the number of faults of type i occurred in s by N_s^i. The post-language $\mathcal{L}(A)/s$ is the set of possible suffixes of a trace s: $\mathcal{L}(A)/s = \{t \in \Sigma^* : st \in \mathcal{L}(A)\}$.

A. Uniform and Non-uniform Diagnosability

We start by recalling the definition of diagnosability for *detecting* the occurrence of faults that is first appeared in [2]. Observe that the following definition is based on the uniformly bounded finite detection delays, which does not depend on the trace executed by the system. For brevity, we call the following notion by *uniform diagnosability*.

Definition 1: A prefix-closed live language L is said to be uniformly diagnosable with respect to a mask function M and Π_f on Σ_f if the following holds:

$$(\exists n_d \in \mathbb{N})(\forall i \in \Pi_f)(\forall s, N_s^i > 0)(\forall t \in L/s)$$
$$[|t| \geq n_d \Rightarrow D_e]$$

where $\mathbb{N}$ is the set of non-negative integers and the diagnosability condition D_e is

$$D_e : \; (\forall w \in M^{-1}M(st) \cap L) \; [\; N_w^i > 0 \;].$$

The above definition of diagnosability only deals with live languages. In general, the behavior of system may block. In [9], the notion of diagnosability accounting for blocking was presented. We observe that attaching the self-loop of the nonfaulty silent event ϵ at blocking states does not affect the property of diagnosability presented in [9]. Therefore, the above definition of diagnosability with the liveness assumption does not lose generality compared to the one without the liveness assumption presented in [9]. In this spirit, we only consider live language in this paper without loss of generality.

The following definition relaxes the uniformity of detection delays by letting the detection delays depend on the current trace executed by the system.

Definition 2: A prefix-closed live language L is said to be nonuniformly diagnosable with respect to a mask function M and Π_f on Σ_f if the following holds:

$$(\forall i \in \Pi_f)(\forall s, N_s^i > 0)(\exists n_{d_i} \in \mathbb{N})(\forall t \in L/s)$$
$$[|t| \geq n_{d_i} \Rightarrow D_e]$$

where $\mathbb{N}$ is the set of non-negative integers and the diagnosability condition D_e is

$$D_e : \; (\forall w \in M^{-1}M(st) \cap L) \; [\; N_w^i > 0 \;].$$

Observe that the difference of the above two definitions is in the order of quantifiers. By placing the existential quantifier for the detection delay after the universal quantifier for the current trace, now the detection delay depend on the type of faults and the current trace executed by the system. It is clear to see that uniform diagnosability implies nonuniform diagnosability but not vice versa, in general. An example showing that the two notions are inequivalent is presented in [1]. However, if the language L describing the behavior of the system is regular, the notions of nonuniform and uniform diagnosability become equivalent since detection delays are uniformly bounded by n^2 where n denotes the number of states of a finite state automaton generating L.

In [3], the notions of K-diagnosability and $[1, K]$-diagnosability were introduced in order to count the occurrence of repeated faults. These notions appeared in [3] are based on faults modelled as states. We modify the definitions of [3] to address faults characterized as events. Note that the detection delay of the following definitions are independent of the current trace of the system. For brevity, we call the following notions by uniform K-diagnosability and uniform $[1, K]$-diagnosability.

Definition 3: A prefix-closed live language L is said to be uniformly K-diagnosable with respect to a mask function M and Π_f on Σ_f if the following holds:

$$(\exists n_d \in \mathbb{N})(\forall i \in \Pi_f)(\forall s \in L, N_s^i \geq K)(\forall t \in L/s)$$
$$[|t| \geq n_d \Rightarrow D_K]$$

where $\mathbb{N}$ is the set of non-negative integers and the diagnosability condition D_K is

$$D_K : \; (\forall w \in M^{-1}M(st) \cap L) \; [\; N_w^i \geq K \;].$$

Definition 4: A prefix-closed live language L is said to be uniformly $[1, K]$-diagnosable with respect to a mask function M and Π_f on Σ_f if the following holds:

$$(\exists n_d \in \mathbb{N})(\forall i \in \Pi_f)(\forall J, 1 \leq J \leq K)$$
$$(\forall s \in L, N_s^i \geq J)(\forall t \in L/s) \; [|t| \geq n_d \Rightarrow D_J]$$

where $\mathbb{N}$ is the set of non-negative integers and the diagnosability condition D_J is

$$D_J: \ (\forall w \in M^{-1}M(st) \cap L) \ [\ N_w^i \geq J \ ].$$

Letting the counting delay depend on the current trace executed by the system and the type of faults, we define nonuniform K-diagnosability and nonuniform $[1,K]$-diagnosability as below.

Definition 5: A prefix-closed live language L is said to be nonuniformly K-diagnosable with respect to a mask function M and Π_f on Σ_f if the following holds:

$$(\forall i \in \Pi_f)(\forall s \in L, N_s^i \geq K)(\exists n_{d_i} \in \mathbb{N})(\forall t \in L/s)$$
$$[|t| \geq n_{d_i} \Rightarrow D_K]$$

where $\mathbb{N}$ is the set of non-negative integers and the diagnosability condition D_K is

$$D_K: \ (\forall w \in M^{-1}M(st) \cap L) \ [\ N_w^i \geq K \ ].$$

Definition 6: A prefix-closed live language L is said to be nonuniformly $[1,K]$-diagnosable with respect to a mask function M and Π_f on Σ_f if the following holds:

$$(\forall i \in \Pi_f)(\forall J, 1 \leq J \leq K)(\forall s \in L, N_s^i \geq J)$$
$$(\exists n_{d_i} \in \mathbb{N})(\forall t \in L/s) \ [|t| \geq n_{d_i} \Rightarrow D_J]$$

where $\mathbb{N}$ is the set of non-negative integers and the diagnosability condition D_J is

$$D_J: \ (\forall w \in M^{-1}M(st) \cap L) \ [\ N_w^i \geq J \ ].$$

It is clear that uniform K-diagnosability ($[1,K]$-diagnosability) imply nonuniform K-diagnosability ($[1,K]$-diagnosability). However, the converse does not hold, in general. This can be shown by observing that uniform 1-diagnosability and $[1,1]$-diagnosability are equivalent to uniform diagnosability. Again, if we assume the regularity of system behavior, counting delays are bounded by $K \cdot |Q^A|^2$, which is independent of the current execution of the system. Thus, a prefix-closed live regular language L is nonuniformly K-diagnosable ($[1,K]$-diagnosable) if and only if L is uniformly K-diagnosable ($[1,K]$-diagnosable).

It is shown in [3] that L is uniformly $[1,K]$-diagnosable with respect to a mask function M and Π_f on Σ_f iff L is uniformly J-diagnosable with respect to a mask function M and Π_f on Σ_f for all J such that $1 \leq J \leq K$. The corresponding result of nonuniform diagnosability with finite counting can be shown as follows:

Proposition 1: A prefix-closed live language L is nonuniformly $[1,K]$-diagnosable with respect to a mask function M and Π_f on Σ_f iff L is nonuniformly J-diagnosable with respect to a mask function M and Π_f on Σ_f for all J such that $1 \leq J \leq K$.

The notions of diagnosability presented up to this point are related to finite counting capability. In order to facilitate indefinite counting with uniformly bounded finite counting delays, the notions of $[1,\infty]$-diagnosability was also introduced in [3], which are recalled below with proper modifications to account for faulty events instead of faulty states. Note that the counting delay of the following definition is independent of the current trace of the system. For brevity, we call the following notion by uniform $[1,\infty]$-diagnosability.

Definition 7: A prefix-closed language L is said to be uniformly $[1,\infty]$-diagnosable with respect to a mask function M and Π_f on Σ_f if the following holds:

$$(\exists n_d \in \mathbb{N})(\forall i \in \Pi_f)(\forall t \in L/s)$$
$$[|t| \geq n_d \Rightarrow D_\infty]$$

where $\mathbb{N}$ is the set of non-negative integers and the diagnosability condition D_∞ is

$$D_\infty: \ (\forall w \in M^{-1}M(st) \cap L) \ [\ N_w^i \geq N_s^i \ ].$$

Letting the counting delay depend on the current trace executed by the system and the type of faults, we introduce the notion of nonuniform $[1,\infty]$-diagnosability as below.

Definition 8: A prefix-closed language L is said to be nonuniformly $[1,\infty]$-diagnosable with respect to a mask function M and Π_f on Σ_f if the following holds:

$$(\forall i \in \Pi_f)(\forall s \in L)(\exists n_{d_i} \in \mathbb{N})(\forall t \in L/s)$$
$$[|t| \geq n_{d_i} \Rightarrow D_\infty]$$

where $\mathbb{N}$ is the set of non-negative integers and the diagnosability condition D_∞ is

$$D_\infty: \ (\forall w \in M^{-1}M(st) \cap L) \ [\ N_w^i \geq N_s^i \ ].$$

It is clear that uniform $[1,\infty]$-diagnosability implies nonuniform $[1,\infty]$-diagnosability. Remind that the notions of uniform diagnosability for finite counting and the corresponding notions of nonuniform diagnosability for finite counting become equivalent if we assume that the language describing the system behavior is regular. However, this does not hold when the occurrence of faults are to be counted indefinitely.

Example 1: Let us consider the regular language $\mathcal{L}(A)$ generated by the automaton A described in Fig. 1. Let $\Sigma_f = \{f\}$. The observation constraint of the system is below.

$$M(a) = M(b) = M(f) = \epsilon \text{ and } M(c) = c.$$

Intuitively, it is clear that if we observe event c n-times, then we can infer that fault event f has occurred at least n-times. In this sense, we can count the occurrence of fault event f eventually as we have more observations of event c. Hence, $\mathcal{L}(A)$ is nonuniformly $[1,\infty]$-diagnosable. However, if the executed trace is $a(ffc)^n$, the actual number of occurrence of fault event f is $2n$. Formally, we can see that $2n^{th}$ occurrence of fault event f in $a(ffc)^{n-1}ff$ can be counted after the system executes $a(ffc)^{2n}$ for $n \geq 1$. Therefore the minimal counting delay for trace $a(ffc)^{n-1}ff$ becomes $3n+1$, which clearly depends on trace $a(ffc)^{n-1}$. Therefore, $\mathcal{L}(A)$ is not uniformly $[1,\infty]$-diagnosable.

The following propositions relate the notions of diagnosability regarding finite fault counting and nonuniform $[1,\infty]$-diagnosability.

Proposition 2: A prefix-closed live language L is nonuniformly $[1,\infty]$-diagnosable with respect to a mask

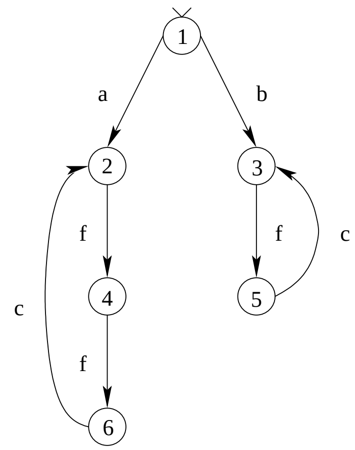

Fig. 1. Automaton A for Example 1

function M and Π_f on Σ_f iff L is nonuniformly K-diagnosable with respect to a mask function M and Π_f on Σ_f for all $K \in \mathbb{N}^+$.

Proposition 3: A prefix-closed regular live language L is nonuniformly $[1, \infty]$-diagnosable with respect to a mask function M and Π_f on Σ_f iff L is uniformly K-diagnosable with respect to a mask function M and Π_f on Σ_f for all $K \geq 1$.

When the language describing the behavior of the system is regular, nonuniform diagnosability is equivalent to uniform diagnosability. Therefore, we can utilize the verification algorithm for uniform diagnosability in [10] to verify the property of nonuniform diagnosability. In the same context, we can use the algorithms in [3] with minor modifications in order to verify nonuniform $K-$ and $[1, K]-$diagnosability.

Though Proposition 3 relates nonuniform $[1, \infty]$-diagnosability and uniform K-diagnosability where we have a verification algorithm, the verification algorithm for uniform K-diagnosability reported in [3] depends on value K. Therefore, the direct application of the verification algorithm for uniform K-diagnosability is not feasible in order to verify uniform $[1, \infty]$-diagnosability. In the following section, we will develop an algorithm for verifying the property of nonuniform $[1, \infty]$-diagnosability. A simple variation of the verification algorithm for the property of nonuniform $[1, \infty]$-diagnosability will be presented in order to verify uniform $[1, \infty]$-diagnosability. This algorithm carries a lower time and space computational complexities than those of the algorithm reported in [3].

III. VERIFICATION OF REPEATED FAULT DIAGNOSABILITY

Let A be a finite-state automaton generating the behavior of the system and let M be a mask function for events defined over Σ^A. We construct a directed graph $G(A, M) = (V(A), E(A, M))$. For notational convenience, we may drop the dependency notation of $G(A, M)$, when it is considered to be clear from the context. The set of vertexes V is

$$V \subseteq Q^A \times Q^A \text{ and } (q_0^A, q_0^A) \in V,$$

and an edge function w is defined as $w : E \to 2^S$ where $S = \{-1, 0^+, 0^-, 0, \hat{0}, +1\}$. The implication of the directed graph G will be explained after we complete the description of G.

Before we proceed to define the edges of G, for the sake of readability, let us define the following transition notation:

$$\delta^A(q_1, \sigma) = q_1' \text{ and } \delta^A(q_2, \sigma') = q_2'.$$

Note that we use event σ to define q_1'. On the other hand, event σ' is used to define q_2'. Also, observe that σ and σ' *can* be identical.

The notation $p \xrightarrow{i} q$ below implies that there exist an edge $(p, q) \in E$ and $i \in S = \{-1, 0^-, 0, \hat{0}, 0^+, +1\}$ such that $i \in w[(p, q)]$. Now we define the edges of G as follows.

For $\sigma, \sigma' \in \Sigma_P$ such that $M(\sigma) = M(\sigma') = \epsilon$,

$(q_1, q_2) \xrightarrow{0^-} (q_1', q_2)$ if $(\sigma \notin \Sigma_f) \wedge (q_1'$ is defined$)$

$(q_1, q_2) \xrightarrow{0^+} (q_1, q_2')$ if $(\sigma' \notin \Sigma_f) \wedge (q_2'$ is defined$)$

$(q_1, q_2) \xrightarrow{-1} (q_1', q_2)$ if $(\sigma \in \Sigma_f) \wedge (q_1'$ is defined$)$

$(q_1, q_2) \xrightarrow{+1} (q_1, q_2')$ if $(\sigma' \in \Sigma_f) \wedge (q_2'$ is defined$)$

For $\sigma, \sigma' \in \Sigma_P$ such that $M(\sigma) = M(\sigma') \neq \epsilon$,

$(q_1, q_2) \xrightarrow{0} (q_1', q_2')$
if $(\sigma \notin \Sigma_f) \wedge (\sigma' \notin \Sigma_f) \wedge (q_1'$ and q_2' are defined$)$

$(q_1, q_2) \xrightarrow{-1} (q_1', q_2')$
if $(\sigma \in \Sigma_f) \wedge (\sigma' \notin \Sigma_f) \wedge (q_1'$ and q_2' are defined$)$

$(q_1, q_2) \xrightarrow{+1} (q_1', q_2')$
if $(\sigma \notin \Sigma_f) \wedge (\sigma' \in \Sigma_f) \wedge (q_1'$ and q_2' are defined$)$

$(q_1, q_2) \xrightarrow{\hat{0}} (q_1', q_2')$
if $(\sigma \in \Sigma_f) \wedge (\sigma' \in \Sigma_f) \wedge (q_1'$ and q_2' are defined$)$

Hereafter, we only consider the accessible part of the weighted, directed graph G from the vertex (q_0^A, q_0^A) when G is referred. With the above construction, we have the directed graph G with the edge function w.

Now, we explain the implication of G. The weighted, directed graph G is designed to track traces $s \in \mathcal{L}(A)$ and $s' \in \mathcal{L}(A)$ such that $M(s) = M(s')$ from the vertex (q_0^A, q_0^A). Specifically, the vertex space and the edge relation are defined to track the traces in the following manner:

$$\underbrace{Q^A}_{s} \times \underbrace{Q^A}_{s'}.$$

Observe that the value of edge function w is designed to indicate if traces s and s' are about to track fault events or not. In particular, when s is about to track a fault event and s' is about to track a normal event or nothing (ϵ), the value -1 is given to edge. On the other hand, the value $+1$ is assigned to edge when s' is about to track a fault event and s is about to track a normal event or nothing. $\hat{0}$ weight is given when both traces are about to track fault events. If no fault events are involved, 0^-, 0^+, and 0 weights are given based on the observability of tracked events.

Now we assign value 0 to various zero notation 0^-, 0^+, $\hat{0}$, and 0. With this, we define the weight of edge as follows: for a given $(p, q) \in E$,

$$w_s[(p, q)] = min(w[(p, q)]).$$

Over G equipped with the weight function w_s, compute the shortest paths from single source (q_0^A, q_0^A) to all reachable vertexes. Denote the shortest path weight of vertex $v \in V$ as $short[v]$. We define the following for further argument.

Definition 9: A cycle in G is called T-cycle where $T \subseteq S$, if for all $t \in T$ there is an edge $(p, q) \in E$ in the cycle such that $t \in w[(p, q)]$.

With G, we claim the following results for the verification of nonuniform and uniform $[1, \infty]$-diagnosability.

Theorem 1: $\mathcal{L}(A)$ is nonuniformly $[1, \infty]$-diagnosable w.r.t. M and Σ_f iff the following three conditions hold:

1) For all T-cycle, if $-1 \in T$ then $\hat{0} \in T$ or $+1 \in T$.
2) For all $v \in \{0^-\}$-cycle, $short[v] \geq 0$; this condition handles unobservable cycles.
3) For all $v \in T$-cycle where $0 \in T \subseteq 2^{\{0^-, 0, 0^+\}}$, $short[v] = 0$.

Theorem 2: $\mathcal{L}(A)$ is uniformly $[1, \infty]$-diagnosable w.r.t. M and Σ_f iff the following three conditions hold:

1) For all $v \in V$, $short[v]$ is finite.
2) For all $v \in \{0^-\}$-cycle, $short[v] \geq 0$; this condition handles unobservable cycles.
3) For all $v \in T$-cycle where $0 \in T \subseteq 2^{\{0^-, 0, 0^+\}}$, $short[v] = 0$.

The above results can be utilized for the polynomial-time verification of nonuniform and uniform $[1, \infty]$-diagnosability. Let $|Q^A| = n_1$ and $|\Sigma_A| = n_2$. The worst case time and space computational complexities for verifying nonuniform and uniform $[1, \infty]$-diagnosability using Theorems 1 and 2 are obtained as follows:

Theorem 3: Let A be a deterministic automaton. The nonuniform $[1, \infty]$-diagnosability and uniform $[1, \infty]$-diagnosability of $\mathcal{L}(A)$ with respect to M and Σ_f can be decided with $O(\min(n_1^3 \cdot n_2^2, n_1^5))$ time and $O(\min(n_1^2 \cdot n_2^2, n_1^4))$ space.

Again we direct the reader to [1] for an illustrating example demonstrating the constructions of the directed, weighted graph G and the application of verification algorithms for the properties of nonuniform $[1, \infty]$-diagnosability and uniform $[1, \infty]$-diagnosability.

IV. ON-LINE DIAGNOSIS FOR REPEATED FAULTS

Building the deterministic observer automaton of a partially-observed automaton takes exponential time and space w.r.t the number of state of the partially-observed automaton. The basic building block of off-line diagnoser construction relies on the construction of observer automaton and exponential computational complexity is carried over. To overcome this computational difficulty, on-line diagnosis approach was suggested in [2] to handle the case of permanent faults. Rather than constructing whole diagnoser off-line, the state of diagnoser is updated whenever observations occur. The space and time complexity of updating the state of diagnoser for reporting permanent faults are $O(|Q^A|)$ and $O(|\Sigma| \cdot |Q^A|)$, respectively. For the case of repeated faults, it is required to count the occurrence of faults in order to reach diagnostic results. Based on the algorithm presented in [3], the space and time complexity of updating the state of diagnoser for counting the occurrence of faults are $O(|Q^A|^3)$ and $O(|\Sigma| \cdot |Q^A|^3)$, respectively.

In this section, we propose an algorithm improving the computational complexity of the proposed algorithm presented in [3]. Similar to the algorithm in [3], we maintain a set of state and corresponding minimum number of occurrence of faults as the state of diagnoser, i.e., $Q_d \in 2^{Q^A \times \mathbb{N}}$ where $Q_d = \{(q_1, i_1), (q_2, i_2), \ldots, (q_n, i_n)\}$. In [3], tagged fault count number i does not have to be unique for each state component q_i. In contrast to [3], the tagged fault count number of our algorithm is unique, that is, $q_i \neq q_j$ if $i \neq j$. The tagged integer value i_j of state q_j represents the minimum number of faults in the traces that are reachable to state q_j and consistent with the current observed trace. The main routine of the algorithm is described in Algorithm 1. In the loop of the main routine, MODIFIED-MULTI-SOURCES-DIJIKSTRA and GET-NEW-DIAGNOSER-STATE routines are called. Algorithms 2 and 3 describes the two subroutines.

In MODIFIED-MULTI-SOURCES-DIJIKSTRA, successive application of a modified version of Dijikstra algorithm is used to compute the minimum number of the occurrence of faults. Elements in Q_d are used as source vertexes in the modified Dijikstra algorithm. In order to count the number of faults with the modified Dijikstra algorithm, the weight of faulty transitions is set to "1". On the other hand, the zero weight is given to non-faulty transitions. Under this weight setting, the shortest path weight of states implies the number of faulty events along the shortest path, which is minimal by the structure of the weight setting. Note that only unobservable transitions are considered when the modified Dijikstra algorithm is applied. With this procedure, we identify the minimum number of faulty events in the possible transitions to the states in unobservable reach from Q_d. The resulting set of states and tagged integers are stored as new Q_d. After a new observation σ_m becomes available, in GET-NEW-DIAGNOSER-STATE(A, Q_d, σ_m), we collect the states of system reached by the observed event σ_m from the unobservable reach of Q_d. The corresponding integer values indicating the minimum number faults are updated based on the weights of possible observed events. Updated states and tagged integer values become Q_d. Based on the updated Q_d, the minimum of the tagged integer values is reported as the number of faults occurred. The procedures from line 3 to 8 of the main routine can be conducted in $O((|\Sigma| + \log |Q^A|) \cdot |Q^A|^2)$ by implementing the priority queue Q_{temp} with a Fibonacci heap [5]. The space used for realizing diagnoser state is $O(|Q^A|)$.

V. CONCLUSION

The previous notions of diagnosability consider only uniform detection delay. In this paper, we extended these notions by considering nonuniform detection delay. We presented a set of new algorithms verifying various notions of diagnosability regarding repeated faults with uniform and nonuniform delays. Our algorithms carry a lower time and space computational complexity than those previously reported in [3]. We also presented a new on-line diagnosis algorithm that also has a lower time and space complexity than the previously reported on-line diagnosis algorithm for repeated faults.

ACKNOWLEDGEMENT

The research reported in this paper was supported by the U.S. Department of Energy under contract W-31-109-Eng-38.

REFERENCES

[1] T. Yoo and H. E. Garcia, "Fault diagnosis of discrete-event systems with uniformly and nonuniformly bounded diagnosis delays," 2003, submitted.

[2] M. Sampath, R. Sengupta, K. S. S. Lafortune, and D. Teneketzis, "Diagnosability of discrete event systems," *IEEE Trans. on Automat. Contr.*, vol. 40, no. 9, pp. 1555–1575, September 1995.

[3] S. Jiang, R. Kumar, and H. E. Garcia, "Diagnosis of repeated/intermittent failures in discrete event systems," *IEEE Trans. on Robotics and Automation*, vol. 19, no. 2, pp. 310–323, 2003.

[4] H. E. Garcia and T. Yoo, "A methodology for detecting routing events in discrete flow networks," in *Proc. 2004 Ameri. Contr. Conf.*, 2004.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*. The MIT Press, 1990.

[6] A. V. Goldberg, "Scaling algorithms for the shortest paths problem," *SIAM J. Comput.*, vol. 24, pp. 494–504, 1995.

[7] B. V. Cherkassky, A. V. Goldberg, and T. Radzik, "Shortest paths algorithms: Theory and experimental evaluation," in *SODA: ACM-SIAM Symposium on Discrete Algorithms (A Conference on Theoretical and Experimental Analysis of Discrete Algorithms)*, 1994. [Online]. Available: citeseer.nj.nec.com/cherkassky93shortest.html

[8] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

[9] T. Yoo and H. E. Garcia, "Computation of fault detection delay in discrete-event systems," in *Proc. DX 2003, 14th International Workshop on Principles of Diagnosis*, 2003, pp. 207–212.

[10] T. Yoo and S. Lafortune, "Polynomial time verification of diagnosability of partially-observed discrete-event systems," *IEEE Trans. Automat. Contr.*, vol. 47, no. 9, pp. 1491–1495, 2002.

Algorithm 1 On-line Diagnosis(A, M)

$Q_d \leftarrow (q_0^A, 0)$
loop
  Find the minimum i_k from $Q_d = \{(q_1, i_1), \ldots, (q_k, i_k), \ldots, (q_n, i_n)\}$ and report it.
  $Q_d \leftarrow$ MODIFIED-MULTI-SOURCES-DIJIKSTRA(A, Q_d, M) [5]
  wait until a next observation (σ_m) is available
  $Q_d \leftarrow$ GET-NEW-DIAGNOSER-STATE(A,Q_d,σ_m)
end loop

Algorithm 2 $Q_d \leftarrow$ MODIFIED-MULTI-SOURCES-DIJIKSTRA(A, Q_d, M)

For all $q' \in Q^A$ where $q' = \delta(q, t)$ for some $(q, i) \in Q_d$ and $M(t) \in \epsilon^*$, set $short[q'] = \infty$, $Q = \{q' : short[q'] = \infty\}$, and $Q_{temp} = \emptyset$.
while $Q_d \neq \emptyset$ **do**
  Pick $(q, i) \in Q_d$ where i is the minimum.
  if $i < short[q]$ **then**
    $short[q] \leftarrow i$ and $Q_{temp} = Q_{temp} \cup \{q\}$
    while $Q_{temp} \neq \emptyset$ **do**
      Find $q \in Q_{temp}$ where $short[q]$ is the minimum and remove q from Q_{temp}
      for each neighbor vertex q' reached from q with an unobservable normal event **do**
        if $i < short[q']$ **then**
          $short[q'] \leftarrow i$ and $Q_{temp} = Q_{temp} \cup \{q'\}$
        end if
      end for
      for each neighbor vertex q' reached from q with an unobservable fault event **do**
        if $i + 1 < short[q']$ **then**
          $short[q'] \leftarrow i + 1$, $Q_{temp} = Q_{temp} \cup \{q'\}$
        end if
      end for
    end while
    $Q_{temp} = \emptyset$
  end if
  $Q_d \leftarrow Q_d \setminus \{(q, i)\}$
end while
$Q_d \leftarrow \{(q, short[q]) : q \in Q\}$

Algorithm 3 $Q_d \leftarrow$ GET-NEW-DIAGNOSER-STATE(A, Q_d, σ_o)

For all $q' \in Q^A$ where $q' = \delta(q, \sigma_o)$ for some $(q, i) \in Q_d$, set $short[q'] = \infty$, $Q = \{q' : short[q'] = \infty\}$, and $Q_{temp} = \emptyset$
while $Q_d \neq \emptyset$ **do**
  Pick $(q, i) \in Q_d$ where i is the minimum
  for each neighbor vertex q' reached from q with a normal event σ s.t. $M(\sigma) = \sigma_o$ **do**
    if $i < short[q']$ **then**
      $short[q'] \leftarrow i$ and $Q_{temp} = Q_{temp} \cup \{q'\}$
    end if
  end for
  for each neighbor vertex q' reached from q with a fault event σ s.t. $M(\sigma) = \sigma_o$ **do**
    if $i + 1 < short[q']$ **then**
      $short[q'] \leftarrow i + 1$, $Q_{temp} = Q_{temp} \cup \{q'\}$
    end if
  end for
  $Q_d \leftarrow Q_d \setminus \{(q, i)\}$
end while
$Q_d \leftarrow \{(q, short[q]) : q \in Q\}$