Optimal Sensor Selection for Discrete Event Systems with Partial Observation [1]

Shengbing Jiang
GM R&D and Planning
Mail Code 480-106-390
30500 Mound Road
Warren, MI 48090-9055
shengbing.jiang@gm.com

Ratnesh Kumar
Department of Electrical & Computer Engineering
Iowa State University
2215 Coover Hall
Ames, IA 50011
rkumar@iastate.edu

Humberto E. Garcia
Argonne National Laboratory
Idaho Falls, ID 83403-2528
garcia@anl.gov

[1]The research was supported in part by the U.S. Department of Energy contract W-31-109-Eng-38, and also in part by the National Science Foundation under the grants NSF-ECS-9709796 and NSF-ECS-0099851, a DoD-EPSCoR grant through the Office of Naval Research under the grant N000140110621, and a KYDEPSCoR grant. The work was performed while the first two authors were at the University of Kentucky.

Abstract

For discrete event systems under partial observation, we study the problem of selection of an optimal set of sensors that can provide sufficient yet minimal events observation information needed to accomplish the task at hand such as that of control or estimation. The sufficiency of the observed information is captured as the fulfillment of a desired formal property such as (co-)observability or normality (for control under partial observation), state-observability (for state estimation under partial observation), diagnosability (for failure diagnosis under partial observation), etc. A selection of sensors can be viewed as a selection of an observation mask and also of an equivalence class of events. A sensor set is called optimal if any coarser selection of the corresponding equivalence class of events results in some loss of the events observation information so that the task at hand cannot be accomplished, or equivalently the desired formal property cannot be fulfilled. We study an optimal selection of sensors over the set of general "non-projection" observation masks. We show that this problem is $\mathcal{NP}$-hard in general. For mask-monotonic properties (that are preserved under increasing precision in events observation information, such as (co)-observability, normality, state-observability, diagnosability, etc.), we present a "top-down" and a "bottom-up" algorithm each of polynomial complexity. We show that observer-ness is not mask-monotonic. We show that the computational complexity can be further improved if the property is preserved under the projection via an intermediary observation mask that is an observer. Our results are obtained in a general setting so that they can be adapted for an optimal selection of sensors for a variety of applications in discrete event systems including (co-)observability, normality, diagnosability (single failure as well as repeated failures), state-observability, and invertibility.

Keywords: Discrete event systems, partial observation, sensor selection

1 Introduction

Discrete event systems are systems that possess discrete states which are event-driven, i.e., evolve in response to certain discrete qualitative changes called events. Examples include manufacturing systems, communication networks and protocols, embedded systems, traffic and transportation systems, asynchronous digital circuits, process control systems, and also many other systems can be modeled as discrete event systems at a certain level of abstraction. These systems react to the events occurring in their environments and also generate output events in response to the input ones. One way to determine/estimate the states of such systems is to track through sensors the events they react to and the events they generate.

The untimed, logical, or qualitative behavior of a discrete event system is given by the collection of all event sequences that can occur starting from its initial state. A feasible event sequence is called a trace, and the collection of all traces starting from the initial state is called a language. Some of the applications that require on-line determination of system states include estimation and control. Several works based upon the supervisory control theory [13] address the control of discrete event systems, and various notions of state estimation under partial observations were first developed in [11].

For a deterministic system, the current state of the system can be uniquely determined by the knowledge of its initial state and the occurred event sequence. This requires that each event be completely observed through sensors. For many applications, however, it is not necessary to determine the current state of the system exactly, and an estimate consisting of a set of potential current states suffices. For example, to know whether a buffer of capacity one in a manufacturing system is full or empty, it is enough to sense the arrival and departure events for the buffer, and the actual types of the incoming parts need not be known (or sensed). Thus, even if more powerful sensors can be installed to sense the actual types of the incoming parts, they are redundant, and simpler motion sensors to sense the arrival and departure of parts suffice.

As argued above, applications may be such that a partial observations of occurring events is adequate for the specific task at hand, and a complete observation of events although feasible is unnecessary. There may also be situations where a complete observation of events is infeasible. For example, it may not be possible to sense the occurrence of a failure event (such as a conveyor mechanism getting jammed). Thus, a lack of complete observation of events may arise either by choice (since application does not warrant a complete observation) or otherwise (such as the infeasibility of sensing the occurrence of a failure event). In either case, it is desirable that the events observation information provided by the available sensors be adequate for the task at hand.

The adequacy of the events observation information provided by the available sensors can be expressed as a formal property that is a function of the selected sensor set. For control under partial observation for example, if it is needed that control actions following a pair of traces be different, then the events observation information provided by the sensors must be able to discriminate between such a pair of traces. This formal property is called observability [10]. If it is further required that all events that a controller can control must also be observed, which for example is the case with the setting of local control [10, 9], then the required formal property is stronger than that of observability, and is called normality [10, 8]. For decentralization control under partial observation, if it is needed that

control actions following a pair of traces be different, then the events observation information provided by the sensors must be such that at least one controller is able to discriminate between such a pair of traces. This formal property is called co-observability [1, 15]. For diagnosis under partial observation, it is required that we be able to discriminate between a nominal and a faulty trace within a bounded delay from the time of the occurrence of a fault, and so the events observation information provided by the sensors must have the required discriminating capability. This formal property is called diagnosability [16]; an extension of this property for discriminating among the numbers of failures, called repeated failure diagnosability is given in [7]. For state estimation under partial observation, it is required that the estimates of the current state converge to a singleton value within a bounded number of state transitions, and so the events observation information provided by the sensors must be able to discriminate among the traces leading to different states within a bounded number of state transitions. This formal property is called state-observability [11]. Other formal properties of interest may include generalized co-observability [21], with-delay-state-observability [11], invertibility (with-delay and with-delay-with-unknown-initial-state) [12], and observer-ness [19]. Algorithms of polynomial complexity for testing the above mentioned properties exist [17, 14, 6, 7, 11, 12, 22, 21].

The presence of a partial observation of events can be modeled via a map, called an observation mask, defined from the set of system events to the set of their observed values. An event is said to be unobservable if it is not sensed, or equivalently, its mask value is the null event. For example, a failure event may be unobservable. A pair of events are called indistinguishable if their observed values are identical. For example, arrival of two different types of parts may be sensed identically by a motion sensor. An observation mask is said to be projection type, if it maps an event to either itself or the null event. Thus, only those events are indistinguishable from each other under a projection observation mask are the ones that are unobservable. Other more general observation masks are called non-projection masks. An observation mask can also be viewed as an equivalence relation over the set of system events together with the null event, where all indistinguishable events belong to the same equivalence class, and the equivalence class of the null event consists of all the unobservable events. An observation mask is said to be finer than another observation mask if each of its equivalence class of events is a subset of some equivalence class of events of the second observation mask. In this case, the second observation mask is said to be coarser than the first one.

It is desirable to have a set of sensors that provides adequate events observation information so that a formal property of interest is satisfied (as noted above the satisfaction of such a formal property implies the capability to accomplish a desired task at hand). Further, the set of sensors should be such that the corresponding equivalence class of events is as coarse as possible. We call a sensor set to be optimal if any equivalence class of events that is coarser than the one induced by the given sensor set is inadequate for the fulfillment of the desired formal property. In this paper we study the problem of selecting an optimal set of sensors, or equivalently an optimal observation mask, over the set of general non-projection observation masks. This extends the earlier works (such as [5, 22, 2, 3]) that select an optimal observation mask over the set of projection observation masks.

It is known that a unique optimal observation mask does not exist for a variety of applications (such as (co-)observability, normality, diagnosability) even when the search space is

the set of projection observation masks [5, 22, 3]. This is because optimal sensor sets may exist for which the induced equivalence classes of events are not comparable (in the degree of coarseness). Further, it is also known that the computation of an optimal observation mask that possesses the least number of equivalence classes of events, called a minimum cardinality optimal sensor set, is of exponential complexity in the number of system events, for a variety of applications including (co-)observability, normality, and diagnosability, even when the search space is the set of projection observation masks [22]. We show that even the problem of selection of an optimal sensor set (not necessarily one with the minimal cardinality) is $\mathcal{NP}$-hard in general. However, for properties such as (co)-observability, normality, state-observability, and diagnosability, that are preserved under increasing precision in events observation information, called *mask-monotonicity property*, the problem of optimal sensor set selection has polynomial complexity in the number of system events. We present algorithms of polynomial complexity for computing optimal sensor sets for mask-monotonic properties. These algorithms guarantee the optimality of the computed sensor set, but not the minimum cardinality of the equivalence classes they induce. We show that the observer-ness does not possess the mask-monotonicity property, and so the computational complexity of an optimal sensor set for the observer-ness property remains open. However, it is interesting to note that the computation of an optimal reporter map for the observer-ness property is known to have a polynomial complexity [20].

For mask-monotonic properties, two different algorithms of polynomial complexity for computing an optimal sensor set (over the set of non-projection observation masks) is presented. The first algorithm is based upon a top-down method that starts with the observation mask that is the least-upper-bound of all the adequate observation masks, i.e., it is the coarsest observation mask finer than all the adequate ones. Since the "identity" observation mask in which all equivalence classes of events are singletons is obviously an adequate observation mask (this we assume without loss of generality, since otherwise no adequate observation mask exists), the least-upper-bound observation mask is the identity observation mask. The second algorithm is based upon a bottom-up method that starts with the observation mask that is the greatest-lower-bound of all the adequate observation masks, i.e., it is the finest observation mask coarser than all the adequate ones. We show that such an observation mask, which we call a necessary observation mask, exists, and present an algorithm of polynomial complexity in the number of system events for computing it. It should be noted that the polynomiality of complexity in the number of system states of all our algorithms follows from the fact that the desired formal properties of interest can be verified polynomially in the number of system states (which is known from earlier works as pointed above).

The complexity of our algorithms can be further improved if the desired formal property possesses an additional property that it is preserved under the projection via an intermediary observation mask, called *mask-preserving property*. Specifically, a property is mask-preserving if it holds that the property is satisfied under a composed observation mask $M_2 \circ M_1$ by an unmasked system if and only if the property is satisfied under the observation mask M_2 by the system masked by M_1. We show that normality possesses the mask-preserving property. It was shown in [18] that whenever the observation mask M_1 is an observer, the system masked by M_1 is less complex, i.e., has fewer states, than the unmasked system. Thus, for a mask-preserving property such as normality, verification of the property under the composed observation mask $M_2 \circ M_1$ for the unmasked system, is equivalent to

verification of the property under the observation mask M_2 for the system masked by M_1, resulting in further computational saving whenever M_1 is an observer.

Our algorithms for computing an optimal sensor set are general in the sense that they can be adapted for computing an optimal sensor set for a specific application at hand such as (co-)observability, normality, diagnosability, repeated diagnosability, state-observability, with-delay-state-observability, invertibility, etc. The rest of the paper is organized as follows. Section 2 presents some notations and preliminaries. The problem of optimal sensor selection and its $\mathcal{NP}$-hardness is shown in Section 3. Two different polynomial complexity algorithms for finding an optimal set of sensors for mask-monotonic properties is given in Section 4. Section 5 explores the computational savings resulting from mask-preserving properties, when the intermediary mask is an observer. Section 6 presents an illustrative example. Finally, Section 7 concludes the work presented.

2 Notation and Preliminary

In this section, we introduce some notations and preliminaries taken from [8].

Let A be a set, a binary relation $R \subset A \times A$ over A is called a *partial ordering relation* if it satisfies the following:

- Reflexivity: $\forall a \in A$, aRa;
- Antisymmetry: $\forall a, b \in A$, $[aRb] \wedge [bRa] \Rightarrow [a = b]$;
- Transitivity: $\forall a, b, c \in A$, $[aRb] \wedge [bRc] \Rightarrow [aRc]$.

A pair (A, R) is called a *partially ordered set* or *poset* if R is a partial ordering relation over A. In the following, we use $\leq$ to denote a partial ordering relation, and use $(A, \leq)$ to denote a poset.

Let $(A, \leq)$ be a poset and $B \subseteq A$, then

1. $b \in B$ is the *least* element of B if $\forall x \in B$, $b \leq x$.
2. $b \in B$ is a *minimal* element (or *minimum*) of B if $\nexists x \in B$ *s.t.* $[x \leq b] \wedge [x \neq b]$.
3. $b \in B$ is the *greatest* element of B if $\forall x \in B$, $x \leq b$.
4. $b \in B$ is a *maximal* element (or *maximum*) of B if $\nexists x \in B$ *s.t.* $[b \leq x] \wedge [x \neq b]$.
5. $a \in A$ is an *infimum* (or the *greatest lower bound*) of B in $(A, \leq)$, denoted $infB$, if $a \leq b$ for all $b \in B$ and $\nexists a' \in A$ with $[a \leq a'] \wedge [a' \neq a]$ such that $a' \leq b$ for all $b \in B$.
6. $a \in A$ is an *supremum* (or the *least upper bound*) of B in $(A, \leq)$, denoted $supB$, if $b \leq a$ for all $b \in B$ and $\nexists a' \in A$ with $[a' \leq a] \wedge [a' \neq a]$ such that $b \leq a'$ for all $b \in B$.

It can be easily shown that for a poset $(A, \leq)$ and $B \subseteq A$, the following holds.

1. $b \in B$ is the *least* element of B if and only if $b = infB$.
2. $b \in B$ is the *greatest* element of B if and only if $b = supB$.

Let $C \subseteq 2^A$, then C is called a *partition* of A if

- $\forall c_1, c_2 \in C,\ c_1 \neq c_2 \Rightarrow c_1 \cap c_2 = \emptyset$;
- $\cup_{c \in C} c = A$.

We use $\mathcal{C}_A$ to denote the set of all partitions of A. $\forall C_1, C_2 \in \mathcal{C}_A$, C_2 is called *finer* than C_1, denoted $C_1 \leq_A C_2$, if $\forall X \in C_2$, $\exists Y \in C_1$ such that $X \subseteq Y$. In this case, C_1 is called *coarser* than C_2. $\forall C_1, C_2 \in \mathcal{C}_A$, C_2 is called *one step finer* than C_1, denoted $C_1 \leq_A^1 C_2$, if $C_1 \leq_A C_2$ and there does not exist $C \in \mathcal{C}_A$ other than C_1 and C_2 such that $C_1 \leq_A C \leq_A C_2$. In this case, C_1 is called *one step coarser* than C_2. It is easy to verify that $\leq_A$ is a partial ordering relation over $\mathcal{C}_A$. Thus, $(\mathcal{C}_A, \leq_A)$ is a poset. We also have that $C_A^{le} = \{A\}$ is the least element of $\mathcal{C}_A$ and $C_A^{ge} = \{\{a\} \mid a \in A\}$ is the greatest element of $\mathcal{C}_A$.

A discrete event system G is modeled as a state machine, which is a 5-tuple:

$$G = (Q, \Sigma, R, q_0, Q_m),$$

where

- Q is a finite state set,
- Σ is a finite event set,
- $R \subseteq Q \times (\Sigma \cup \{\epsilon\}) \times Q$ is the state transition set,
- $q_0 \in Q$ is the initial state,
- $Q_m \subseteq Q$ is the set of marked state.

We use $L(G) \subseteq \Sigma^*$ (resp., $L_m(G)$) to denote the generated language (resp., marked language) of G, where Σ^* denotes the set of all finite length event traces. The partial observation of events is modeled as an observation mask $M : \Sigma \cup \{\epsilon\} \to \Delta \cup \{\epsilon\}$ with $M(\epsilon) = \epsilon$, and Δ is the set of observed values. For notational simplicity, we adopt the following: $\overline{\Sigma} := \Sigma \cup \{\epsilon\}$ and $\overline{\Delta} := \Delta \cup \{\epsilon\}$. M can be extended to the event traces in Σ^* as follows: $\forall s \in \Sigma^*,\ \sigma \in \Sigma,\ M(s\sigma) = M(s)M(\sigma)$. The mask of the system G, denoted by $M(G)$, is the state machine obtained from G by masking the event label of each transition in G, i.e., $M(G) = (Q, \Delta, R_M, q_0, Q_m)$ with $R_M = \{(q_1, M(\sigma), q_2) \mid (q_1, \sigma, q_2) \in R\}$. It is easy to verify that $L(M(G)) = M(L(G))$.

Let $\mathcal{C}_{\overline{\Sigma}}$ be the set of all partitions of $\overline{\Sigma}$. Then each observation mask M induces a partition $C^M \in \mathcal{C}_{\overline{\Sigma}}$ of $\overline{\Sigma}$, where $C^M = \{[\sigma]_M \mid \sigma \in \overline{\Sigma}\}$ with $[\sigma]_M = \{\sigma' \in \overline{\Sigma} \mid M(\sigma') = M(\sigma)\}$. Conversely, each partition $C \in \mathcal{C}_{\overline{\Sigma}}$ of $\overline{\Sigma}$ induces an observation mask $M_C : \overline{\Sigma} \to \overline{\Delta}_C$, where $\Delta_C = \{Y \in C \mid \epsilon \notin Y\}$, and $\forall \sigma \in \Sigma$, if $\sigma \in Y \in \Delta_C$ then $M_C(\sigma) = Y$, otherwise $M_C(\sigma) = \epsilon$.

As mentioned earlier in the introduction, we are interested in selecting an observation mask M such that certain properties hold, these include, among other properties, observability, normality, and diagnosability. Given a system G, an observation mask M, and a language $K \subseteq \Sigma^*$:

- K is said to be *observable* with respect to G and M if

$$\forall s, t \in pr(K), \sigma \in \Sigma : \ M(s) = M(t), s\sigma \in pr(K), t\sigma \in L(G) \Rightarrow t\sigma \in pr(K)$$

where $pr(K) = \{s \in \Sigma^* \mid \exists t \in \Sigma^* \ s.t. \ st \in K\}$ is the set of all prefixes of event traces in K.

- K is said to be *normal* with respect to G and M if

$$\forall s, t \in L(G) : \ s \in pr(K), t \in L(G), M(s) = M(t) \Rightarrow t \in pr(K).$$

Next we give the definition of diagnosability as in [16]. Let G be a non-terminating (or dead-lock free) system, $\mathcal{F} = \{F_i, i = 1, 2, \ldots, m\}$ be a set of failure types, and $\psi : \Sigma \rightarrow \mathcal{F} \cup \{\emptyset\}$ be a failure assignment function for each event in Σ. G is said to be *diagnosable* with respect to the observation mask M and the failure assignment function ψ if the following holds:

$(\forall F_i \in \mathcal{F}) \ (\exists n_i \in N) \ (\forall s \in L(G), \psi(s_f) = F_i) \ (\forall v = st \in L(G), ||t|| \geq n_i)$

$\Rightarrow (\forall w \in L(G), M(w) = M(v)) \ (\exists u \in pr(\{w\}), \psi(u_f) = F_i)$

where s_f and u_f denote the last events in traces s and u respectively, $pr(\{w\})$ is the set of all prefixes of w. In [7], the notion of diagnosability is extended to the cases of repeated failures.

All the above properties, i.e., observability, normality, and diagnosability, can be tested in polynomial time in the size of the system. For the test of the diagnosability, we can use the algorithm presented in [7] (instead of the one in [6]), which does not require the non-existence of cycles of unobservable events in the system.

In the following, we introduce the definition of an observer ([19]) for an observation mask, which is useful in this paper. Given a system G and an observation mask M, M is said to be an *observer* if

$$\forall s, t \in L(G) : \ M(s) = M(t) \Rightarrow [\forall su \in L(G), \exists tv \in L(G), \ s.t. \ M(v) = M(u)].$$

Note that in the above definition of an observer we only consider the observation mask, and not the more general report map as in [19].

3 Optimal Sensor Selection Problem & its Complexity

In this section, we formulate the optimal sensor selection problem and study its complexity. From the definitions of the properties such as observability, normality, and diagnosability, we know that whether or not a property holds for a given observation mask M, depends on the partition C^M induced by M, and it has nothing to do with the set Δ itself. Thus, we can give the following definition.

Definition 1 Given an event set Σ and a property P over Σ, let I denote the collection of input specifications for the property P other than an observation mask (such as a system G, a control specification K, a failure assignment function ψ, etc.). Let $\mathcal{C}_{\overline{\Sigma}}$ be the set of all partitions of $\overline{\Sigma}$, then for a partition $C \in \mathcal{C}_{\overline{\Sigma}}$, the property P is said to hold under the partition C, denoted $< I, C > \models P$, if P holds under the observation mask M_C induced by the partition C.

We use $\mathcal{C}^P \subseteq \mathcal{C}_{\overline{\Sigma}}$ to denote the set of all partitions under each of which P holds, i.e., $\mathcal{C}^P = \{C \in \mathcal{C}_{\overline{\Sigma}} \mid < I, C > \models P\}$.

Next we give the definition of an optimal observation mask.

Definition 2 Let Σ be an event set, M be an observation mask, and P be a property, M is said to be *optimal* for the property P if C^M is a minimum in $\mathcal{C}^P$, where C^M is the partition induced by M.

The problem of selecting an optimal sensor set is stated as follows:

> Given an event set Σ and a property P find an optimal observation mask M over Σ for the property P.

The above problem is a search problem over the set of all observation masks $\mathcal{C}_{\overline{\Sigma}}$. The decision version of this search problem can be stated as follows.

OPTIMAL-SENSOR-EXISTENCE
INSTANCE: An event set Σ and a property P over Σ.
QUESTION: Does there exist an optimal observation mask M for P?

In the following we show that the OPTIMAL-SENSOR-EXISTENCE problem is $\mathcal{NP}$-hard ([4]) in the size of the event set. Then it follows directly that the optimal sensor selection problem is $\mathcal{NP}$-hard in the size of the system events.

Theorem 1 OPTIMAL-SENSOR-EXISTENCE is $\mathcal{NP}$-hard.

Proof: The $\mathcal{NP}$-hardness of OPTIMAL-SENSOR-EXISTENCE is established by the reduction from the SATISFIABILITY problem (a known $\mathcal{NP}$-complete problem).

Let ϕ be a boolean formula over n variables, $V = \{v_k, 1 \leq k \leq n\}$, in the conjunctive normal form (CNF) consisting of m clauses:

$$\phi = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{l_i} u_{ij};\ u_{ij} = v_k \text{ or } \neg v_k \text{ for some } k \in \{1, \cdots, n\},$$

where $l_i \leq n$ is the number of variables in the ith clause. A truth assignment for V is a function $f : V \to \{0, 1\}$. ϕ is said to be satisfiable if there exists a truth assignment f such that

$$\phi|_f = \left(\bigwedge_{i=1}^{m} \bigvee_{j=1}^{l_i} w_{ij} \right) = 1; \quad \text{where } w_{ij} = f(v_k) \text{ if } u_{ij} = v_k, \text{ and } w_{ij} = \neg f(v_k) \text{ if } u_{ij} = \neg v_k,$$

i.e., ϕ is TRUE under the truth assignment f. The SATISFIABILITY problem ([4]) is specified as follows.

SATISFIABILITY
INSTANCE: A variable set V and a boolean formula ϕ over V in CNF.
QUESTION: Is ϕ satisfiable?

From an instance (V, ϕ) of SATISFIABILITY, we construct an instance (Σ_ϕ, P_ϕ) of OPTIMAL-SENSOR-EXISTENCE as follows (see the example given below for illustration):

- The event set is defined as

$$\Sigma_\phi = \{e \mid e = u_{ij}, i = 1, \cdots, m,\ j = 1, \cdots, l_i\} \cup V,$$

  where we treat u_{ij} and v_k as two different events even if $u_{ij} = v_k$.

Prior to defining the property P_ϕ, we first show that there exists a property $\hat{\phi}$ defined over the variables of the set Σ_ϕ such that $\hat{\phi}$ is satisfiable if and only if ϕ is satisfiable.

Definition of $\hat{\phi}$:

$$\hat{\phi} := \phi \wedge \bigwedge_{(v \in V, E_v \neq \emptyset)} [(v \wedge \bigwedge_{e \in E_v} e) \vee (\neg v \wedge \bigwedge_{e \in E_v} \neg e)] \wedge \bigwedge_{(v \in V, E_{\neg v} \neq \emptyset)} [(v \wedge \bigwedge_{e \in E_{\neg v}} \neg e) \vee (\neg v \wedge \bigwedge_{e \in E_{\neg v}} e)],$$

where $E_v = \{e \in \Sigma_\phi - V \mid e = v\}$ and $E_{\neg v} = \{e \in \Sigma_\phi - V \mid e = \neg v\}$.

To see that $\hat{\phi}$ is satisfiable if and only if ϕ is satisfiable, let f_1 be a truth assignment for V with $\phi|_{f_1} = 1$, then we can extend f_1 to a truth assignment f_2 for Σ_ϕ as follows: $\forall e \in \Sigma_\phi$,

$$f_2(e) = \begin{cases} f_1(e) & \text{if } e \in V \\ f_1(v) & \text{if } (e \in \Sigma_\phi - V) \wedge (v \in V) \wedge (e = v) \\ \neg f_1(v) & \text{if } (e \in \Sigma_\phi - V) \wedge (v \in V) \wedge (e = \neg v) \end{cases}$$

It can be verified that $\hat{\phi}|_{f_2} = 1$. Conversely, for a truth assignment f_2 for Σ_ϕ with $\hat{\phi}|_{f_2} = 1$, we can obtain a truth assignment f_1 for V by restricting f_2 on $V \subseteq \Sigma_\phi$, i.e., $f_1 = f_2|_V$. Then it is obvious that $\phi|_{f_1} = 1$.

We next show that given an observation mask M over Σ_ϕ, it induces a truth assignment function f_M over the symbols of Σ_ϕ.

Definition of f^M induced by M: We first arrange the events in Σ_ϕ in the following order:

$$e_1 = v_1, \cdots, e_n = v_n, e_{n+1} = u_{11}, \cdots, e_{n+l_1} = u_{1l_1}, \cdots, e_{(n+\sum_{i=1}^{m} l_i)} = u_{ml_m},$$

and given an observation mask M over Σ_ϕ, iteratively define the truth assignment function $f^M : \Sigma_\phi \to \{0, 1\}$ as:

$$f^M(e_1) = \begin{cases} 1 & \text{if } M(e_1) \neq \epsilon \\ 0 & \text{otherwise} \end{cases}$$

$$f^M(e_k) = \left\{ \begin{cases} f^M(e_{k-1}) & \text{if } M(e_k) = M(e_{k-1}) \\ \neg f^M(e_{k-1}) & \text{otherwise} \end{cases} \right\},\ k = 2, \cdots, |\Sigma|$$

We now use the boolean formula $\hat{\phi}$ and the truth assignment function f^M to define the property P_ϕ.

- We define P_ϕ to be the property over Σ_ϕ such that it is satisfied under an observation mask M if and only if the boolean formula $\hat{\phi}$ is satisfied under the truth assignment f^M, i.e., $\hat{\phi}|_{f^M} = 1$.

It is easy to verify that the size of Σ_ϕ is $O(m \times n)$, and the length of $\hat{\phi}$ is $O(|\phi| + m \times n)$. Also, the complexity of construction of f^M from M is polynomial in the size of Σ_ϕ. It follows that given an instance of the satisfiability problem, an instance of an optimal sensor existence problem can be constructed polynomially in the size m and n of the satisfiability problem specification.

Next we prove that ϕ is satisfiable if and only if there is an optimal observation mask for the property P_ϕ defined above. Since Σ_ϕ is finite, we know that there is an optimal observation mask for the property P_ϕ if and only if $\mathcal{C}^{P_\phi} \neq \emptyset$, i.e., if and only if there is an observation mask M such that P_ϕ is satisfied under M, i.e., if and only if the following holds

$$\exists M,\ \hat{\phi}|_{f^M} = 1.$$

From the definition of $\hat{\phi}$, it follows that ϕ is satisfiable if and only if $\hat{\phi}$ is satisfiable. Thus we only need to prove that $\hat{\phi}$ is satisfiable if and only if there exists an observation mask M such that $\hat{\phi}$ is TRUE under f^M, i.e.,

$$(\exists f : \Sigma_\phi \to \{0,1\} \quad s.t. \quad \hat{\phi}|_f = 1) \Longleftrightarrow (\exists M \quad s.t. \quad \hat{\phi}|_{f^M} = 1).$$

The implication from the right to the left follows directly from the fact that f^M is a truth assignment for Σ_ϕ. For the implication from the left to the right, it suffices to show that from a truth assignment f for Σ_ϕ, an observation mask M_f for Σ_ϕ with $f^{M_f} = f$ can be constructed. Assuming the events in Σ_ϕ are in the same order as in the above definition of f^M, then the mask M_f can be constructed iteratively from f as follows:

$$M_f(e_1) = \left\{ \begin{array}{ll} e_1 & \text{if } f(e_1) = 1 \\ \epsilon & \text{otherwise} \end{array} \right\}$$

$$M_f(e_k) = \left\{ \begin{array}{ll} M_f(e_{k-1}) & \text{if } f(e_k) = f(e_{k-1}) \\ e_k & \text{otherwise} \end{array} \right. ,\ k = 2, \cdots, |\Sigma|$$

It is easy to verify that $f^{M_f} = f$. This completes the proof. ∎

The above theorem states that unless $\mathcal{P} = \mathcal{NP}$, it is not possible to obtain a polynomial algorithm in the size of the event set for the selection of optimal sensors for satisfying a general property of observation masks.

The following example illustrates the construction of an instance of the optimal sensor existence problem from that of the SATISFIABILITY problem.

Example 1 Let (V, ϕ) be an instance of the SATISFIABILITY problem, where

$$\begin{aligned} V &= \{v_1, v_2, v_3\} \\ \phi &= (v_1 \vee v_2) \wedge (v_1 \vee v_3) \wedge (v_2 \vee \neg v_3) \end{aligned}$$

Using new symbols u_{ij}, ϕ can be expressed as

$$\phi = (u_{11} \vee u_{12}) \wedge (u_{21} \vee u_{22}) \wedge (u_{31} \vee u_{32}),$$

with $u_{11} = u_{21} = v_1$, $u_{12} = u_{31} = v_2$, $u_{22} = v_3$, and $u_{32} = \neg v_3$. An instance (Σ_ϕ, P_ϕ) of the optimal sensor existence problem is constructed from (V, ϕ) as:

- $\Sigma_\phi = \{v_1, v_2, v_3, u_{11}, u_{12}, u_{21}, u_{22}, u_{31}, u_{32}\}$;
- The property P_ϕ over Σ_ϕ is defined such that it holds under an observation mask M over Σ_ϕ if and only if the statement "$\hat{\phi}|_{f^M} = 1$", where

$$\begin{aligned}\hat{\phi} \;=\; & \phi \wedge [(v_1 \wedge u_{11} \wedge u_{21}) \vee (\neg v_1 \neg u_{11} \wedge \neg u_{21})] \\ & \wedge [(v_2 \wedge u_{12} \wedge u_{31}) \vee (\neg v_2 \neg u_{12} \wedge \neg u_{31})] \\ & \wedge [(v_3 \wedge u_{22}) \vee (\neg v_3 \wedge \neg u_{22})] \\ & \wedge [(v_3 \wedge \neg u_{33}) \vee (\neg v_3 \wedge u_{33})]\end{aligned}$$

  and f^M is obtained from M as in the proof of Theorem 1.

Suppose a mask M is given as: $M(v_1) = M(v_2) = M(v_3) = w_1$, $M(u_{11}) = M(u_{12}) = M(u_{32}) = \epsilon$, $M(u_{21}) = w_2$, and $M(u_{22}) = M(u_{31}) = w_3$, where $w_i \neq \epsilon$ $(i = 1, 2, 3)$ are different from each other. Then f^M is constructed as (assuming the events in Σ_ϕ are in the order as listed above): $f^M(v_1) = f^M(v_2) = f^M(v_3) = 1$, $f^M(u_{11}) = f^M(u_{12}) = 0$, $f^M(u_{21}) = 1$, $f^M(u_{22}) = f^M(u_{31}) = 0$, and $f^M(u_{32}) = 1$. It is easy to verify that $\hat{\phi}|_{f^M} = 0$, i.e., P_ϕ is not satisfied under M.

Now suppose M is given as: $M(v_1) = M(u_{12}) = w_1$, $M(v_2) = M(v_3) = w_2$, $M(u_{11}) = M(u_{21}) = M(u_{32}) = \epsilon$, and $M(u_{22}) = M(u_{31}) = w_3$, where $w_i \neq \epsilon$ $(i = 1, 2, 3)$ are different from each other. Then f^M is obtained as: $f^M(v_1) = f^M(u_{11}) = f^M(u_{21}) = f^M(u_{32}) = 1$ and $f^M(v_2) = f^M(v_3) = f^M(u_{12}) = f^M(u_{22}) = f^M(u_{31}) = 0$. It can be verified that $\hat{\phi}|_{f^M} = 1$, i.e., P_ϕ is satisfied under M.

4 Optimal Sensor Selection Under Mask-Monotonicity

We next introduce the notion of mask-monotonic properties, and show that for such properties the complexity of optimal sensor selection problem is polynomial in the size of the event set, and also provide two algorithms of polynomial complexity for computing optimal sensor sets.

Definition 3 Given a property P, P is said to be mask-monotonic if

$$\forall C_1, C_2 \in \mathcal{C}_{\overline{\Sigma}},\ (C_1 \leq_{\overline{\Sigma}} C_2) \wedge (< I, C_1 > \models P) \Rightarrow (< I, C_2 > \models P).$$

The mask-monotonicity simply states that the property P is preserved under an increase in events observation information. It can be verified that the properties of observability, normality, and diagnosability are mask-monotonic. The following example shows that the property of observer-ness is not mask-monotonic.

Example 2 Consider the system G shown in Figure 1, where $\Sigma = \{a_1, a_2, b_1, b_2\}$. Let M_1 and M_2 be two observation masks such that

$$\begin{aligned}& M_1(a_1) = M_1(a_2) = a \neq \epsilon,\ M_1(b_1) = M_1(b_2) = b \neq \epsilon; \\ & M_2(a_1) = M_2(a_2) = a \neq \epsilon,\ M_2(b_1) = b_1, M_2(b_2) = b_2.\end{aligned}$$

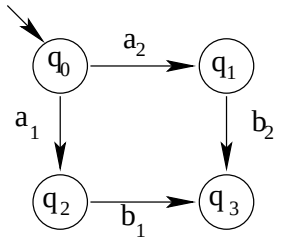

Figure 1: Observer-ness is not $\leq_{\overline{\Sigma}}$-monotonic

Then the partitions C^{M_1} and C^{M_2} that are induced by M_1 and M_2 respectively can be obtained as

$$C^{M_1} = \{\{a_1, a_2\}, \{b_1, b_2\}, \{\epsilon\}\};$$
$$C^{M_2} = \{\{a_1, a_2\}, \{b_1\}, \{b_2\}, \{\epsilon\}\}.$$

It is obvious that $C^{M_1} \leq_{\overline{\Sigma}} C^{M_2}$. It is also easy to verify that M_1 is an observer and M_2 is not an observer. This is because $M_2(a_1) = M_2(a_2)$, $a_1 b_1 \in L(G)$, and there does not exist a trace $u \in \Sigma^*$ such that

$$a_2 u \in L(G) \cap M_2^{-1} M_2(a_1 b_1) = L(G) \cap \{a_1 b_1, a_2 b_1\} = \{a_1 b_1\}.$$

Thus, the property of observer-ness is not mask-monotonic. Also note that in the above example, we have that the identity mask $M_3 = Id$ (that masks every event to itself), is an observer, and $C^{M_2} \leq_{\overline{\Sigma}} C^{M_3}$. Thus, we can find an observer mask M_3 by refining a non-observer mask M_2.

The following theorem presents some properties of the set $\mathcal{C}^P$ when P is mask-monotonic.

Theorem 2 Let Σ be an event set, $\mathcal{C}_{\overline{\Sigma}}$ be the set of all partitions of $\overline{\Sigma}$, P be a mask-monotonic property, and $\mathcal{C}^P$ be the set of all partitions under which P holds. Then we have:

1. $\mathcal{C}^P \neq \emptyset \iff C^{ge}_{\overline{\Sigma}} \in \mathcal{C}^P$
2. $sup\mathcal{C}^P = \begin{cases} C^{ge}_{\overline{\Sigma}} & \text{if } \mathcal{C}^P \neq \emptyset \\ C^{le}_{\overline{\Sigma}} & \text{otherwise} \end{cases}$
3. $\forall C \in \mathcal{C}^P \neq \emptyset$: $inf\mathcal{C}^P \leq_{\overline{\Sigma}} C \leq_{\overline{\Sigma}} C^{ge}_{\overline{\Sigma}}$, where $inf\mathcal{C}^P$ may not be in $\mathcal{C}^P$, i.e., the least element of $\mathcal{C}^P$ may not exist.

Proof: For the first assertion, the forward implication comes from the mask-monotonicity of P, and the backward implication is obvious.

For the second assertion, if $\mathcal{C}^P \neq \emptyset$ then from the first assertion we have $sup\mathcal{C}^P = C^{ge}_{\overline{\Sigma}}$; otherwise $sup\mathcal{C}^P = sup\emptyset$. From the definition of supremum, we know that $sup\emptyset$ is the least element of $\mathcal{C}_{\overline{\Sigma}}$, i.e., $sup\emptyset = C^{le}_{\overline{\Sigma}}$.

The first part of the third assertion follows directly from the second assertion and the definitions of infimum and supremum. Since P is mask-monotonic, it is easy to verify that $inf\mathcal{C}^P \in \mathcal{C}^P$ if and only if it holds that,

$$C_1, C_2 \in \mathcal{C}^P \Rightarrow inf\{C_1, C_2\} \in \mathcal{C}^P.$$

In general, the above does not hold for a property P. (For example, this does not hold for observability.) This implies that $inf\mathcal{C}^P$ may not be in $\mathcal{C}^P$, i.e., the least element of $\mathcal{C}^P$ may not exist. This completes the proof. ■

From Definition 2, we know that the optimal sensor selection problem is to find a minimum partition in $\mathcal{C}^P$. Noting the fact of Theorem 2 that $\forall C \in \mathcal{C}^P$, $inf\mathcal{C}^P \leq_{\overline{\Sigma}} C \leq_{\overline{\Sigma}} C^{ge}_{\overline{\Sigma}}$, we have two methods to find a minimum partition in $\mathcal{C}^P$:

Top-down method: Starting from $C_0 = C^{ge}_{\overline{\Sigma}}$, recursively find a partition C_{i+1}, with $i \geq 0$, satisfying $[C_{i+1} \leq^1_{\overline{\Sigma}} C_i] \wedge [C_{i+1} \neq C_i] \wedge [< I, C_{i+1} > \models P]$, until none can be found; then the partition obtained is a minimum.

Bottom-up method: Starting from $C_0 = inf\mathcal{C}^P$, recursively find a partition C_{i+1}, with $i \geq 0$, satisfying $[C_i \leq^1_{\overline{\Sigma}} C_{i+1}] \wedge [C_{i+1} \neq C_i]$, until $< I, C_{i+1} > \models P$; next apply the top-down method to the partition obtained.

Remark 1 The top-down search method is self-explanatory, but the bottom-up search method deserves further explanation. The first part of the bottom-up method is for finding a partition $C \in \mathcal{C}^P$ starting from $inf\mathcal{C}^P$. Although the property P holds under this partition C, C may not be a minimum in $\mathcal{C}^P$, and so we need to apply the top-down method from this partition C for finding a minimum in $\mathcal{C}^P$.

The following examples illustrates this further.

Example 3 Suppose $\Sigma = \{a, b, c\}$, and for an observation mask of Σ, the property P is defined as the statement "the partition C^M of $\overline{\Sigma}$ induced by M is finer than either C_3 or C_4", where $C_3 = \{\{a, b\}, \{c, \epsilon\}\}$ and $C_4 = \{\{a, c\}, \{b, \epsilon\}\}$. It is obvious that P is mask-monotonic; $\mathcal{C}^P \neq \emptyset$; C_3 and C_4 are two minimums in $\mathcal{C}^P$; and $inf\mathcal{C}^P = \{\{a, b, c, \epsilon\}\}$. For finding a partition in $\mathcal{C}^P$ starting from $C_0 = inf\mathcal{C}^P$, first since a is not masked to ϵ in both C_3 and C_4, we obtain $C_1 = \{\{a\}, \{b, c, \epsilon\}\}$ which is not in $\mathcal{C}^P$; and next since b has a different masked value from c in both C_3 and C_4, we obtain $C_2 = \{\{a\}, \{b\}, \{c, \epsilon\}\}$. Since $C_3 \leq_{\overline{\Sigma}} C_2$, C_2 is in $\mathcal{C}^P$. But C_2 is not a minimum in $\mathcal{C}^P$. Thus, we need to use the top-down method for finding a minimum starting from C_2, from which C_3 is obtained finally. The above situation is shown in Figure 2, where $C_0 \rightarrow C_1 \rightarrow C_2 \leftarrow C_3$ is the path we searched.

In the following, we describe these two different methods. Given an event set Σ and a property P over Σ, the top-down method is given next. The algorithm starts from the least-upper-bound of all the adequate partitions of the set $\overline{\Sigma}$, which is the partition in which all equivalence classes are singleton. It then iteratively searches over all one-step coarser partitions in a "depth-first" manner until a partition is found such that all one-step coarser partition of it violate the property P.

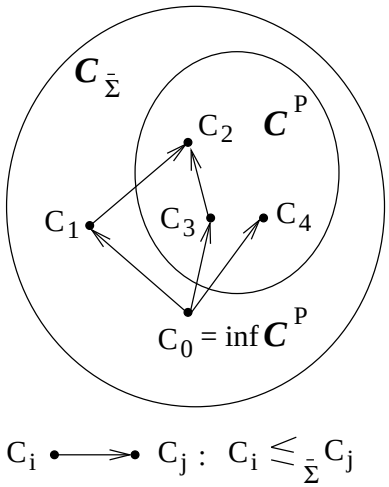

Figure 2: Illustrating the bottom-up method

Algorithm 1 Top-down algorithm for finding a minimum partition in $\mathcal{C}^P$

1. Initialization: $i = 0$, $C_0 = C^{ge}_{\overline{\Sigma}} = \{\{\sigma\} \mid \sigma \in \overline{\Sigma}\}$, and label each pair $X, Y \in C_0$ as "unsearched". If $< I, C_0 > \not\models P$ then stop and output that "no minimum exists"; otherwise continue to the next step.

2. If there does not exist an unsearched pair $X, Y \in C_i$, then stop and output C_i as a minimum in $\mathcal{C}^P$; otherwise pick an unsearched pair $X, Y \in C_i$ and set $\hat{C} = (C_i - \{X, Y\}) \cup \{X \cup Y\}$ and check whether $< I, \hat{C} > \models P$. If NO, then label the pair $X, Y \in C_i$ as searched and go to Step 2; else go to the next step.

3. Set $C_{i+1} = \hat{C}$, and for all $Z \in C_{i+1} - \{X \cup Y\}$, label the pair $X \cup Y, Z \in C_{i+1}$ as searched if either the pair $X, Z \in C_i$ or the pair $Y, Z \in C_i$ is labeled searched; labels of all other pairs in C_{i+1} are retained from those in C_i. Set $i = i + 1$ and go to Step 2.

Remark 2 Since the number of initial "unsearched" pairs in $C_0 = C^{ge}_{\overline{\Sigma}}$ is $O(|\Sigma|)^2$, and the number of pairs as well as the number of unsearched pairs monotonically decreases, the worst case complexity of Algorithm 1 is $O(|\Sigma|^2 \times T_P)$, where T_P is the complexity of checking the property P, which we assume to be polynomial in the size of the system and any other specification involved. (For properties such as observability, normality, and diagnosability, the polynomiality of T_P is known. Thus, Algorithm 1 is polynomial in number of events and states for such properties.) It also follows that if the search for an optimum is restricted to over the set of projection masks, then the number of initial "unsearched pairs in $C_0 = C^{ge}_{\overline{\Sigma}}$ is $|\Sigma|$, and so the complexity of the algorithm is $O(|\Sigma| \times T_P)$, which is linear in the size of the events set.

It is obvious that if Algorithm 1 outputs "no minimum exists", if and only if $\mathcal{C}^P$ is empty. On the other hand, if Algorithm 1 generates a partition, the following theorem shows that the partition is a minimum in $\mathcal{C}^P$.

Theorem 3 Algorithm 1 generates a minimum in $\mathcal{C}^P$ whenever $\mathcal{C}^P \neq \emptyset$.

Proof: Since initially C_0 is a partition, it is obvious that C_1 constructed in Algorithm 1 is also a partition. From Algorithm 1, we know that if $\mathcal{C}^P \neq \emptyset$, then a solution obtained by the algorithm must be in $\mathcal{C}^P$. Thus, we only need to show that the solution obtained is a minimum in $\mathcal{C}^P$.

Now let C_{min} be a solution generated by Algorithm 1, suppose there exists another $C \in \mathcal{C}^P$ such that $C \leq_{\overline{\Sigma}} C_{min}$ and $C \neq C_{min}$, i.e., $S = \{C \in \mathcal{C}^P \mid [C \leq_{\overline{\Sigma}} C_{min}] \wedge [C \neq C_{min}]\} \neq \emptyset$. Let C_{max} be a maximum in S. Since S is non-empty and finite, a maximum does exist. From the mask-monotonicity of P, we know that C_{min} must be one step finer than C_{max}, i.e., $C_{max} \leq^1_{\overline{\Sigma}} C_{min}$. It implies that there exists X and Y in C_{min} such that $C_{max} = (C_{min} - \{X, Y\}) \cup \{X \cup Y\}$. But since C_{min} is a solution generated by Algorithm 1, there does not exist such a C_{max} after the algorithm stops at the solution C_{min}. Thus, we must have that $S = \emptyset$, i.e., C_{min} is a minimum in $\mathcal{C}^P$. ■

Next we develop the bottom-up method for finding a minimum in $\mathcal{C}^P$. For this, we need to first compute $inf\mathcal{C}^P$. We assume from now that $< I, C^{ge}_{\overline{\Sigma}} > \models P$ so that $\mathcal{C}^P \neq \emptyset$ and $inf\mathcal{C}^P$ is non-trivial. The idea behind the computation of $inf\mathcal{C}^P$ is simple: Suppose P is satisfied by making a pair $\sigma_1, \sigma_2 \in \overline{\Sigma}$ indistinguishable, i.e., $< I, C_{(1,2)} > \models P$, where

$$C_{(1,2)} := \{\{\sigma\} \mid \sigma \in \overline{\Sigma} - \{\sigma_1, \sigma_2\}\} \cup \{\sigma_1, \sigma_2\},$$

then σ_1 and σ_2 belong to the same equivalence class in $inf\mathcal{C}^P$ since $C_{(1,2)} \in \mathcal{C}^P$, and so by definition, $inf\mathcal{C}^P \leq C_{(1,2)}$. The algorithm is described below. For an illustration of the algorithm, please refer to Section 6.

Algorithm 2 Algorithm for computing $inf\mathcal{C}^P$

1. Initialization: $C_{inf} = \emptyset$, label all pairs $\sigma_1, \sigma_2 \in \overline{\Sigma}$ as unsearched, and set $Y = \overline{\Sigma}$.
2. Check if $Y = \emptyset$. If YES, then set $inf\mathcal{C}^P = C_{inf}$; otherwise pick $\sigma_0 \in Y$ and set $X = \{\sigma_0\}$.
3. Check if there exists an unsearched pair $\sigma_1 \in X$ and $\sigma_2 \in Y - X$. If NO, then set $C_{inf} = C_{inf} \cup \{X\}$, $Y = Y - X$, and go to Step 2; otherwise label the pair σ_1, σ_2 as searched, and check if $< I, C_{(1,2)} > \models P$. If YES, then set $X = X \cup \{\sigma_2\}$. Repeat Step 3.

Remark 3 Since Algorithm 2 searches each event pair at most once, its worst case complexity is $O(|\Sigma|^2 \times T_P)$, where T_P is the complexity of checking the property P. It also follows that if the search for an optimum is restricted to over the set of projection masks, then the complexity of the algorithm is $O(|\Sigma| \times T_P)$, which is linear in the size of the events set.

The following theorem shows that Algorithm 2 is correct.

Theorem 4 C_{inf} computed by Algorithm 2 equals $inf\mathcal{C}^P$.

Proof: From the construction of C_{inf}, we know that C_{inf} is a partition of $\overline{\Sigma}$. We first prove that $C_{inf} \leq_{\overline{\Sigma}} C$ for every $C \in \mathcal{C}^P$, i.e., $\forall C \in \mathcal{C}^P$ and $\forall X \in C$, there exists $Y \in C_{inf}$ such that $X \subseteq Y$. If X is a singleton, i.e., $X = \{\sigma\}$ for some $\sigma \in \overline{\Sigma}$, then obviously there exists $Y \in C_{inf}$ such that $\sigma \in Y$, and so $X \subseteq Y$. If X is not a singleton, then pick $\sigma_1 \in X$. Then for all $\sigma_2 \in X$ with $\sigma_2 \neq \sigma_1$, $C \leq_{\overline{\Sigma}} C_{(1,2)}$, where $C_{(1,2)} = \{\{\sigma\} \mid \sigma \in \overline{\Sigma} - \{\sigma_1, \sigma_2\}\} \cup \{\{\sigma_1, \sigma_2\}\}$. It follows from mask-monotonicity and the assumption $C \in \mathcal{C}^P$ that $C_{(1,2)} \in \mathcal{C}^P$. Form the construction of C_{inf} we know that $C_{inf} \leq_{\overline{\Sigma}} C_{(1,2)}$, and so there exists a $Y \in C_{inf}$ such that $X \subseteq Y$.

Next we prove that there does not exist a partition C_{Inf} such that $C_{Inf} \leq_{\overline{\Sigma}} C$ for every $C \in \mathcal{C}^P$, $C_{inf} \leq_{\overline{\Sigma}} C_{Inf}$, and $C_{Inf} \neq C_{inf}$. For contradiction, we suppose that there exists such a partition C_{Inf}. Then we must have: $\exists \sigma_1, \sigma_2 \in \overline{\Sigma}$, $\exists X \in C_{inf}$, and $\exists Y_1, Y_2 \in C_{Inf}$ such that $\sigma_1 \neq \sigma_2$, $\{\sigma_1, \sigma_2\} \subseteq X$, $\sigma_1 \in Y_1$, $\sigma_2 \in Y_2$, and $Y_1 \neq Y_2$. It further implies that $C_{(1,2)} \in \mathcal{C}^P$ but $C_{Inf} \not\leq_{\overline{\Sigma}} C_{(1,2)}$, a contradiction. Thus, no such a partition C_{Inf} exists. This completes the proof. ■

Now we present our bottom-up method for computing a minimum in $\mathcal{C}^P$ that starts from $inf\mathcal{C}^P$, and iteratively obtains a one-step refinement until a mask under which P holds is found, and finally applies the top-down method starting from that mask.

Algorithm 3 Bottom-up algorithm for finding a minimum partition in $\mathcal{C}^P$

1. Initialization: $\hat{C}_0 = inf\mathcal{C}^P$, $i = 0$.
2. Check whether $< I, \hat{C}_i > \models P$. If YES, then go to Step 3; otherwise from the test for P, a counter example consisting of a pair of indistinguishable event traces is generated, and by distinguishing certain two events σ_1 and σ_2 in some equivalence class of $\hat{C}_i$, we can make this counter example non-existent. So there exists $X \in \hat{C}_i$ with $\{\sigma_1, \sigma_2\} \subseteq X$. Set $\hat{C}_{i+1} = (\hat{C}_i - \{X\}) \cup \{X - \{\sigma_1\}\} \cup \{\{\sigma_1\}\}$, and repeat Step 2 with $i = i + 1$.
3. Apply Algorithm 1 with $C_0 = \hat{C}_i$.

Remark 4 It is easy to verify that the worst case complexity of Algorithm 3 is also $O(|\Sigma|^2 \times T_P)$, where T_P is the complexity of checking the property P. It also follows that if the search for an optimum is restricted to over the set of projection masks, then the complexity of the algorithm is $O(|\Sigma| \times T_P)$, which is linear in the size of the events set.

The following theorem guarantees the correctness of Algorithm 3.

Theorem 5 The partition generated by Algorithm 3 is a minimum of $\mathcal{C}^P$.

Proof: It is easy to verify that in Algorithm 3, $\hat{C}_i \leq_{\overline{\Sigma}} \hat{C}_{i+1}$ and $\hat{C}_i \neq \hat{C}_{i+1}$. Since $\mathcal{C}^P \neq \emptyset$, and $\mathcal{C}_{\overline{\Sigma}}$ is finite, we know that Step 2 of Algorithm 3 terminates by finding a $\hat{C}_i \in \mathcal{C}^P$. Further following the proof of Theorem 3, it can be seen that starting from any initial partition $C_0 \in \mathcal{C}^P$, Algorithm 1 generates a minimum of $\mathcal{C}^P$, and so Step 3 of Algorithm 3 generates a minimum of $\mathcal{C}^P$. This completes the proof. ■

Remark 5 In the above analysis, we have implicitly assumed that any two events can have a same masked value, i.e., any two events can share a same sensor. However, this may not be

true in practical situations. In some applications, we may need to find an optimal solution that is feasible, i.e., a solution satisfying the feasibility specification which specifies the sets of events that can share a sensor. In order to find such a feasible solution, the top-down and bottom-up algorithms are modified in the following.

We first introduce the notion of feasible partitions. Let $S_f \in \mathcal{C}_\Sigma$ be a partition of Σ that denotes the *feasibility specification*: For any $X \subseteq \Sigma$, all events in X are permitted to be indistinguishable if $\exists Y \in S_f$ such that $X \subseteq Y$. Let $X \subseteq \overline{\Sigma}$, X is said to be *feasible* with respect to S_f if either $\epsilon \in X$ or $\exists Y \in S_f$ such that $X \subseteq Y$. Let $C \in \mathcal{C}_{\overline{\Sigma}}$ be a partition of $\overline{\Sigma}$, C is said to be *feasible* with respect to S_f if $\forall X \in C$, X is feasible with respect to S_f.

To restrict search over feasible partitions of $\overline{\Sigma}$, Algorithm 1 may be modified as follows: In Step 2, only those unsearched pairs X, Y with $X \cup Y$ being feasible shall be picked. It can then be verified that if the initial partition is feasible, then whenever the Algorithm 1 can output a partition, it will output a feasible one. It is obvious that the initial partition $C_0 = C_{\overline{\Sigma}}^{ge}$ in Algorithm 1 is feasible with respect to any $S_f \in \mathcal{C}_\Sigma$. Also, it can be proved that if a partition C_i is computed by the above modified algorithm, then no feasible partition coarser than C_i exists, i.e., C_i is an optimal feasible partition.

Also, Algorithm 3 may be modified as follows: In Step 3, the modified Algorithm 1 shall be used, and further instead of choosing $C_0 = \hat{C}_i$, the partition C_0 shall be chosen as:

$$\hat{C}_i^{S_f} = \{X_\epsilon\} \cup \{X_1 \cap X_2 \mid X_1 \cap X_2 \neq \emptyset, X_1 \in \hat{C}_i - \{X_\epsilon\}, X_2 \in S_f\},$$

where $X_\epsilon \in \hat{C}_i$ with $\epsilon \in X_\epsilon$. It can be verified that $\hat{C}_i^{S_f}$ is a feasible partition finer than $\hat{C}_i$, which implies that $< I, \hat{C}_i^{S_f} > \models P$ from the monotonicity of P. From the argument about the modified Algorithm 1, we can know that whenever the modified Algorithm 3 can output a partition, it will output an optimal feasible one.

5 Sequential test of Mask-Preserving Properties

Recall that the complexity of both top-down and bottom-up algorithms is $O(|\Sigma|^2 \times T_P)$, where T_P is the complexity of checking P for a given observation mask. It is known that T_P is polynomial in the number of system states for a variety of properties of interest. Thus, the complexity of checking P can be improved when

- it is possible to reduce the state size of the system such that
- the reduced system satisfies the property P if and only if the original system satisfies the property P.

From the work in [18], it is possible to reduce the state size via a projection mask whenever the observation mask is an *observer*, and we next introduce the notion of *mask-preserving* properties such that the reduced system satisfies P if and only if the original system satisfies P.

In Algorithm 1, for each C_{i+1} we need to test whether $< I, C_{i+1} > \models P$ given that $< I, C_i > \models P$ holds. Let $M_{C_{i+1}}$ and M_{C_i} be the observation masks induced by C_{i+1} and C_i respectively, then there exists a mask M_{i+1} such that $M_{C_{i+1}} = M_{i+1} \circ M_{C_i}$ (here the

operation "$\circ$" is defined as: for any event e, $M_{C_{i+1}}(e) = M_{i+1} \circ M_{C_i}(e) = M_{i+1}(M_{C_i}(e)))$, and let $C_{M_{i+1}}$ be the associated equivalence class. We can check $< I, C_{i+1} >\models P$ by checking whether $< M_{C_i}(I), C_{M_{i+1}} >\models P$ if P has the property of "mask-preserving" which is defined as follows.

Definition 4 Given observation masks $M_1, M_2, M = M_2 \circ M_1$, we say that P is mask-preserving if it holds that

$$< I, C_{M_1} >\models P \Rightarrow [[< I, C_M >\models P] \Leftrightarrow [< M_1(I), C_{M_2} >\models P]].$$

It follows from the definition of mask-preserving properties that $< I, C_M >\models P$ can be checked by checking $< M_1(I), C_{M_2} >\models P$, where $M = M_2 \circ M_1$. So whenever M_1 is an observer for I, the state size of $M_1(I)$ will be smaller than I, lending to a computational saving in verifying $< I, C_M >\models P$. Next, we first show that the property of normality is mask-preserving, and then present an algorithm for checking whether a given observation mask is an observer. In [20], there is an algorithm for testing the observer property with a worst case complexity of $O(|Q|^5 \times |\Sigma|)$, where $|Q|$ is the number of states in the system. Our test for the observer property has a complexity of $O(|Q|^4 \times |\Sigma|^2)$.

Theorem 6 Consider a system G, an observation mask $M = M_2 \circ M_1$, and a language $K \subseteq L(G)$. Suppose that K is normal with respect to G and M_1, then K is normal with respect to G and M if and only if $M_1(K)$ is normal with respect to $M_1(G)$ and M_2.

Proof: We first prove the necessity. From [8], we know that K is normal with respect to G and M if and only if

$$M^{-1}M(pr(K)) \cap L(G) = pr(K).$$

Using $M = M_2 \circ M_1$, it follows that

$$M_1^{-1}M_2^{-1}M_2M_1(pr(K)) \cap L(G) = pr(K).$$

Applying M_1 on both sides yields

$$M_1[M_1^{-1}M_2^{-1}M_2M_1(pr(K)) \cap L(G)] = M_1(pr(K)).$$

It can be verified easily that for any languages L_1 and L_2 over the event set Σ,

$$M_1(M_1^{-1}(L_1) \cap L_2) = L_1 \cap M_1(L_2).$$

Thus we have

$$M_2^{-1}M_2M_1(pr(K)) \cap M_1(L(G)) = M_1(pr(K)).$$

Using the fact that masking and prefix-closure operations commute, the above equation can be rewritten as:

$$M_2^{-1}M_2(pr(M_1(K))) \cap L(M_1(G)) = pr(M_1(K)).$$

This establishes the necessity.

For sufficiency, suppose $M_1(K)$ is normal with respect to $M_1(G)$ and M_2, i.e.,

$$M_2^{-1}M_2(pr(M_1(K))) \cap L(M_1(G)) = pr(M_1(K)).$$

Then by applying M_1^{-1} operation on both sides and taking an intersection with $L(G)$ we have,

$$M_1^{-1}[M_2^{-1}M_2(pr(M_1(K))) \cap L(M_1(G))] \cap L(G) = M_1^{-1}[pr(M_1(K))] \cap L(G).$$

By using the facts that $M_1^{-1}(L_1 \cap L_2) = M_1^{-1}(L_1) \cap M_1^{-1}(L_2)$ and $L(M_1(G)) = M_1(L(G))$, and by applying the commutativity of masking and prefix-closure operations, we get

$$M_1^{-1}M_2^{-1}M_2M_1(pr(K)) \cap M_1^{-1}M_1(L(G)) \cap L(G) = M_1^{-1}M_1(pr(K)) \cap L(G),$$

which can be simplified as

$$M_1^{-1}M_2^{-1}M_2M_1(pr(K)) \cap L(G) = M_1^{-1}M_1(pr(K)) \cap L(G).$$

Finally since K is normal with respect to G and M_1, the right hand side equals $pr(K)$, i.e.,

$$M_1^{-1}M_1(pr(K)) \cap L(G) = pr(K).$$

So it follows that

$$M_1^{-1}M_2^{-1}M_2M_1(pr(K)) \cap L(G) = pr(K).$$

Thus K is normal with respect to G and M. ■

The following algorithm provides an $O(|Q|^4 \times |\Sigma|^2)$ test for checking the observer property.

Algorithm 4 Algorithm for testing observer-ness for system $G = (Q, \Sigma, R, q_0)$ and mask M

1. Construct $G_1 = (Q_1, \Delta, R_1, q_0^1)$ from the "masked synchronous composition" of G with itself as follows:

   - $Q_1 = Q \times Q$ is the state set;
   - Δ is the event set;
   - $R_1 \subseteq Q_1 \times (\Delta \cup \{\epsilon\}) \times Q_1$ is the state transition set that is defined as:
     for all $q_{12} = (q_1, q_2)$ and $q'_{12} = (q'_1, q'_2)$ in Q_1, and for all $\tau \in \overline{\Delta}$, $(q_{12}, \tau, q'_{12}) \in R_1$ if and only if one of the following holds
     - $\tau = \epsilon$, $q_1 = q'_1$ (resp., $q_2 = q'_2$), and $\exists \sigma \in \overline{\Sigma}$ such that $M(\sigma) = \epsilon$ and $(q_2, \sigma, q'_2) \in R$ (resp., $(q_1, \sigma, q'_1) \in R$);
     - $\tau \neq \epsilon$, and $\exists \sigma_1,\ \sigma_2 \in \Sigma$ such that $M(\sigma_1) = M(\sigma_2) = \tau$, $(q_1, \sigma_1, q'_1) \in R$, and $(q_2, \sigma_2, q'_2) \in R$.
   - $q_0^1 = (q_0, q_0)$ is the initial state.

2. Check in G_1 whether there exists a state $q_{12} = (q_1, q_2) \in Q_1$ such that it is reachable from q_0^1 and the following holds:

   Exists $\sigma \in \Sigma$ such that either $(q_1, \sigma, q'_1) \in R$ or $(q_2, \sigma, q'_2) \in R$, but there does not exist a $q'_{12} \in Q_1$ with $(q_{12}, M(\sigma), q'_{12}) \in R_1$.

If the answer to this check is YES, then M is not an observer for G; otherwise M is an observer for G.

The correctness of Algorithm 4 follows directly from the definition of an observer given in Section 2.

Theorem 7 Given a system G and an observation mask M, M is an observer for G if and only if Algorithm 4 does not answer YES.

Proof: It follows from the construction of G_1 that a state $q_{12} = (q_1, q_2) \in Q_1$ is reachable from q_0^1 if and only if there exist $s, t \in L(G)$ such that $M(s) = M(t)$ and execution of s (resp., t) in G results in state q_1 (resp., q_2). So if the Algorithm answers YES, there exists either $s\sigma \in L(G)$ but no $\sigma' \in M^{-1}M(\sigma)$ with $t\sigma' \in L(G)$, or $t\sigma \in L(G)$ but no $\sigma' \in M^{-1}M(\sigma)$ with $s\sigma' \in L(G)$. In either case, the observer property is violated. On the other hand, if the observer property is violated, we must have $s, t \in L(G)$ with $M(s) = M(t)$, and $\sigma \in \Sigma$ such that either $s\sigma \in L(G)$ but no $\sigma' \in M^{-1}M(\sigma)$ with $t\sigma' \in L(G)$, or $t\sigma \in L(G)$ but no $\sigma' \in M^{-1}M(\sigma)$ with $s\sigma' \in L(G)$. In either case, the Algorithm answers YES. ■

Remark 6 Since the number of states in G_1 is $O(|Q|^2)$ and the number of transitions is G_1 is $O(|R|^2)$, the complexity of Algorithm 4 is $O(|Q|^2+|R|^2)$. Since G may be nondeterministic, $|R|$ is bounded by $|Q|^2 \times |\Sigma|$. So it follows that the worst case complexity of Algorithm 4 is $O(|Q|^4 \times |\Sigma|^2)$.

Now we present an algorithm for the "sequential test" of a mask-preserving property. For the sake of concreteness, we illustrate it via the sequential test for normality, and discuss the resulting computational savings. Let G be a deterministic system, $M = M_2 \circ M_1$ be an observation mask, $K \subseteq L(G)$ be a language which is normal with respect to G and M_1, and H be a deterministic automaton that accepts the language K. Then we have the following algorithm to test the normality of K with respect to G and M.

Algorithm 5 Algorithm for the sequential test of normality

1. Test whether M_1 is an observer for both G and H by using Algorithm 4. If the answer is NO, then test for the normality of K with respect to G and M, and stop; otherwise continue to the next step.

2. Obtain deterministic automata G_1 and H_1 language equivalent to the nondeterministic automata $M_1(G)$ and $M_1(H)$ respectively.

3. Test for the normality of $M_1(K) = L(H_1)$ with respect to G_1 and M_2.

Remark 7 Let n_H and n_G be the number of states in H and G respectively. It can be verified that the test for normality of $L(H)$ with respect to G and $M = M_2 \circ M_1$ has a complexity of $O(n_H^4 \times n_G \times |\Sigma|^2)$. Also, the worst case complexity of Algorithm 5 is $O(n_H^4 \times |\Sigma|^2 + n_G^4 \times |\Sigma|^2 + n_H^4 \times n_G \times |\Sigma|^2)$ when M_1 is not an observer for both G and H. Thus, there is no computational saving in the worst case by the use of Algorithm 5. But if M_1 is an observer for both G and H, then the complexity of Algorithm 5 is $O(n_H^4 \times |\Sigma|^2 + n_G^4 \times |\Sigma|^2 + n_{H_1}^4 \times n_{G_1} \times |M_1(\Sigma)|^2)$. Since M_1 is an observer, from [18] we have $n_{H_1} \leq n_H$ and $n_{G_1} \leq n_G$. Since, obviously, $|M_1(\Sigma)| \leq |\Sigma|$, there is some computational saving that results from the usage of Algorithm 5.

6 Illustrative Example

In this section, we present a simple example to illustrate the concepts and algorithms developed in this paper. Consider a traffic monitoring problem of a mouse that moves around in maze of rooms, one of which is occupied by a cat. The maze, shown in Figure 3, consists of four rooms connected by various one-way passages, and all passages, except the two passages

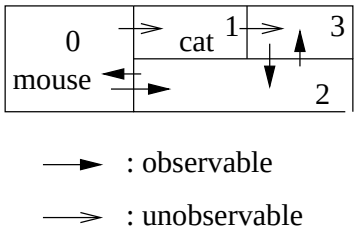

Figure 3: Mouse in a maze

connected to room 1, can have sensors installed to detect the motion of the mouse through them. The cat always stays in room 1. The mouse is initially in room 0, and it can visit other rooms by using the one way passages, and it never stays at one room forever. A failure is said to have occurred if the mouse moves to the room where the cat stays. The task is to decide what passages should have sensors installed such that by observing the sensor signals, we are able to detect (within some finite delay) the occurrence of the failure, so that the system becomes diagnosable. In the maze, two passages connecting same two rooms may share a single sensor. When two passages share one sensor, we can know whether the mouse has gone through one of them, but we cannot know exactly which passage the mouse has gone through. Due to the feasibility requirement of the sensor selection (Remark 5), we require that only those passages that connect the same two rooms may share a sensor. Thus for example a passage connecting rooms 0 and 2 can not share a sensor with a passage connecting rooms 2 and 3.

The above problem can be modeled as the following discrete event system shown in Figure 4. $G = (Q, \Sigma, R, q_0)$, where $Q = \{q_i, 0 \leq i \leq 3\}$, $\Sigma = \{o_1, o_2, o_3, o_4, u_1, u_2\}$, and

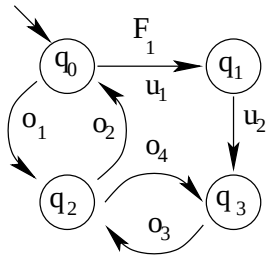

Figure 4: System model

the events u_1 and u_2 are always unobservable. Here the states represent the rooms, i.e.,

state q_i indicates room i, $i = 0, \cdots, 3$, and the events represent the passages connecting different rooms. The correspondence of events and passages can be seen clearly from Figure 4. $\mathcal{F} = \{F_1\}$ is the failure type set, and the failure assignment function ψ is given as $\psi(o_i) = \emptyset$ for $i = 1, \cdots, 4$, $\psi(u_2) = \emptyset$, and $\psi(u_1) = \{F_1\}$. The feasibility specification (Remark 5) is given as $S_f = \{\{o_1, o_2\}, \{o_3, o_4\}\}$.

Since u_1 and u_2 are always unobservable, we only need to consider the event set $\overline{\Sigma}_m = \{\epsilon,$ $o_1,$ $o_2,$ $o_3,$ $o_4\}$, and find a minimal and feasible partition of $\overline{\Sigma}_m$ so as the diagnosability holds. We use P to denote the property of the diagnosability. For a partition C of $\overline{\Sigma}_m$, the diagnosability P is said to hold for the partition C, i.e., $< I, C > \models P$, if $< I, C^u > \models P$, where C^u is the partition of $\overline{\Sigma}$ induced by C, i.e., $C^u = (C - \{X_\epsilon\}) \cup \{X_\epsilon \cup \{u_1, u_2\}\}$ with $\epsilon \in X_\epsilon \in C$. Let $\mathcal{C}^P$ be the set of all partitions of $\overline{\Sigma}_m$ under which the diagnosability P holds. From the algorithm for testing diagnosability given in [7], we know that $C^{ge}_{\overline{\Sigma}_m} = \{\{\sigma\} \mid \sigma \in \overline{\Sigma}_m\} \in \mathcal{C}^P$, i.e., $\mathcal{C}^P \neq \emptyset$.

Here we use the bottom-up method, i.e., Algorithm 3, for finding an optimal observation mask for the property of diagnosability.

We first compute $inf\mathcal{C}^P$ by using Algorithm 2. Initially, $C_{inf} = \emptyset$ and $Y = \overline{\Sigma}_m$. Next, we set $X = \{\epsilon\}$ by applying Step 2 of Algorithm 2. Then from Step 3 of Algorithm 2, we find that for the pair $(\sigma_1 = \epsilon, \sigma_2 = o_1)$, $< I, C_{(1,2)} > \models P$. Thus we set $X = \{\epsilon, o_1\}$ and repeat Step 3. After repeating Step 3 for several times, we have $X = \overline{\Sigma}_m$, $C_{inf} = \{X\}$, and $Y = \emptyset$. Finally from Step 2, we have that $inf\mathcal{C}^P = \{\overline{\Sigma}_m\}$.

Next by applying Algorithm 3 and the test for diagnosability in [7] we can get a minimum of $\mathcal{C}^P$. From Steps 1 and 2 of Algorithm 3, we first get a partition in $\mathcal{C}^P$. From the test for diagnosability in [7], we know that $< I, inf\mathcal{C}^P > \not\models P$, and we find that o_4 and ϵ should be distinguishable from each other, i.e., the partition $\hat{C}_1 = \{\{\epsilon, o_1, o_2, o_3\}, \{o_4\}\}$ is obtained first. Since $< I, \hat{C}_1 > \not\models P$, we next find that o_2 and ϵ should be distinguishable from each other, i.e., the partition $\hat{C}_2 = \{\{\epsilon, o_1, o_3\}, \{o_4\}, \{o_2\}\}$ is obtained next. Again $< I, \hat{C}_2 > \not\models P$, and we further find that o_3 and ϵ should be distinguishable from each other, i.e., the partition $\hat{C}_3 = \{\{\epsilon, o_1\}, \{o_4\}, \{o_2\}, \{o_3\}\}$ is obtained; and $< I, \hat{C}_3 > \models P$. It is easy to verify that $\hat{C}_3$ is feasible with respect to S_f.

Then by applying Algorithm 1 from Step 3 of Algorithm 3, we find that we can merge $\{\epsilon, o_1\}$ and $\{o_2\}$ in $\hat{C}_3$ and obtain a minimum of $\mathcal{C}^P$ as $\{\{\epsilon, o_1, o_2\}, \{o_4\}, \{o_3\}\}$. Note that in applying Algorithm 1, as stated in Remark 5, we should not consider the merger of $\{o_2\}$ with either $\{o_3\}$ or $\{o_4\}$. This is because both $\{o_2, o_3\}$ and $\{o_2, o_4\}$ are infeasible with respect to S_f. The final optimal observation mask M is obtained as: $M(u_1) = M(u_2) = M(o_1) = M(o_2) = \epsilon$, $M(o_3) = o_3$, and $M(o_4) = o_4$.

To illustrate the example further, suppose that the two one-way passages connecting rooms 0 and 2, as well as those connecting rooms 2 and 3, are replaced by one two-way passages. Then the solution obtained above, namely, $\{\{\epsilon, o_1, o2\}, \{o_3\}, \{o_4\}\}$ requires a sensor to detect not only the movement of the mouse through the two-way passage connecting rooms 2 and 3, but also the direction of the movement. Now suppose we are only interested in a solution that only requires movement detection sensors, i.e., we should not distinguish o_1 from o_2 and o_3 from o_4, then Algorithm 1 can be used for this purpose. In applying Algorithm 1, initially we shall set $C_0 = \{\{\epsilon\}, \{o_1, o_2\}, \{o_3, o_4\}\}$, which captures the requirement that we shall not distinguish o_1 from o_2 and o_3 from o_4. It is obvious that C_0 is feasible.

From the test of diagnosability, we know that $< I, C_0 > \models P$. From Algorithm 1, we can find that no partition coarser than C_0 and satisfying P exists. Thus, the optimal observation mask M is obtained as: $M(u_1) = M(u_2) = \epsilon$, $M(o_1) = M(o_2) \neq \epsilon$, $M(o_3) = M(o_4) \neq \epsilon$, which only requires two movement detection sensors.

7 Conclusion

For discrete event systems under partial observation, the problem of optimal sensor selection is studied in this paper. The goal is to come up with a set of sensors that provide minimal yet sufficient events observation information that is adequate for the task at hand such as estimation, diagnosis, or control. We have taken a general approach to the problem of selecting an optimal sensors so that it can be adapted to a variety of applications such as those characterized by the formal properties of (co-)observability, normality, state-observability with or without delay, diagnosability of single or repeated failures, invertibility, etc. Also, we consider the optimization over the set of general non-projection observation masks.

We show that the optimal sensor selection problem is $\mathcal{NP}$-hard in general. We identify a key property of mask-monotonicity with respect to increasing precision of the events observation mask, which lets us compute an optimal sensor set in complexity that is quadratic in the size of the events set. Two methods of such complexity for computing an optimal sensor set are presented: A bottom-up (resp. top-down) method that starts from the greatest-lower-bound (resp., least-upper-bound) and searches up (resp., down) the chain of the sensor sets. We identify the least-upper-bound as well as the greatest-lower-bound for the set of all adequate sensor sets, and provide algorithms of polynomial complexity for computing them. We end the paper by introducing the notion of mask-preserving properties, which together with the observer property allows for further computational savings. We show that the normality is mask-preserving, and also present an algorithm of polynomial complexity for checking the observer property.

References

[1] R. Cieslak, C. Desclaux, A. Fawaz, and P. Varaiya. Supervisory control of discrete event processes with partial observation. *IEEE Transactions on Automatic Control*, 33(3):249–260, 1988.

[2] R. Debouk, S. Lafortune, and D. Teneketzis. On an optimization problem in sensor selection. *Journal of Discrete Event Dynamical Systems: Theory and Application*, 1999. Submitted.

[3] A. Degani, M. Heymann, G. Meyer, and M. Shafto. Some formal aspects of human-automation interaction. Technical Report NASA/TM-2000-209600, NASA Ames Research Center, Moffett Field, CA, 2000.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[5] A. Haji-Valizadeh and K. A. Loparo. Minimizing the cardinality of an event set for supervisors of discrete-event dynamical systems. *IEEE Transactions on Automatic Control*, 41(11):1579–1593, 1996.

[6] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial algorithm for testing diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, pages 1318–1321, August 2001.

[7] S. Jiang, R. Kumar, and H. E. Garcia. Diagnosis of repeated failures in discrete event systems. *IEEE Transactions on Automatic Control*, July 2001. Submitted.

[8] R. Kumar and V. K. Garg. *Modeling and Control of Logical Discrete Event Systems*. Kluwer Academic Publishers, Boston, MA, 1995.

[9] F. Lin and W. M. Wonham. Decentralized supervisory control of discrete event systems. *Information Sciences*, 44:199–224, 1988.

[10] F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44(3):173–198, 1988.

[11] C. M. Ozveren and A. S. Willsky. Observability of discrete event dynamical systems. *IEEE Transactions on Automatic Control*, 35(7):797–806, 1990.

[12] C. M. Ozveren and A. S. Willsky. Invertibility of discrete-event dynamical systems. *Mathematics of Control, Signals and Systems*, 5:365–390, 1992.

[13] P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM Journal of Control and Optimization*, 25(1):206–230, 1987.

[14] K. Rudie and J. C. Willems. The computational complexity of decentralized discrete-event control problems. *IEEE Transactions on Automatic Control*, 40(7):1313–1319, July 1995.

[15] K. Rudie and W. M. Wonham. Think globally, act locally: decentralized supervisory control. *IEEE Transactions on Automatic Control*, 37(11):1692–1708, November 1992.

[16] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Diagonsability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, September 1995.

[17] J. N. Tsitsiklis. On the control of discrete event dynamical systems. *Mathematics of Control Signals and Systems*, 2(2):95–107, 1989.

[18] K. C. Wong. On the complexity of projections of discrete-event systems. In *IEE Workshop on Discrete Event Systems*, pages 201–208, August 1998.

[19] K. C. Wong and W. M. Wonham. Hierarchical control of discrete event systems. *Discrete Event Dynamical Systems: Theory and Applications*, 6:241–273, 1996.

[20] K. C. Wong and W. M. Wonham. On the computation of observers in discrete-event systems. In *2000 Conference on Information Sciences and Systems*, Princeton University, March 2000.

[21] T. Yoo and S. Lafortune. A generalized framework for decentralized supervisory control of discrete event systems. In *Proceedings of 2000 International Workshop on Discrete Event Systems*, Ghent, Belgium, 2000.

[22] T. Yoo and S. Lafortune. On the computational complexity of some problems arising in partially-observed discrete-event systems. In *Proceedings of 2001 American Control Conference*, Arlington, VA, 2001.