Minimization of Sensor Activation in Decentralized Discrete-Event Systems

Xiang Yin, *Member, IEEE*, and Stéphane Lafortune, *Fellow, IEEE*

***Abstract*—We investigate the problem of dynamic sensor activation for decentralized decision making in partially observed discrete-event systems, where the system is monitored by a set of agents. The sensors of each agent can be turned on/off online dynamically according to a sensor activation policy. We define a general decentralized decision-making problem called the decentralized state disambiguation problem, which covers the decentralized control problem, the decentralized fault diagnosis problem, and the decentralized fault prognosis problem. The goal is to find a *language-based* minimal sensor activation policy for each agent such that the agents can always make a correct global decision as a team. A novel approach to solve this problem is proposed. We adopt a *person-by-person* approach to decompose this decentralized minimization problem into two centralized constrained minimization problems. Each centralized constrained minimization problem is then reduced to a fully centralized sensor activation problem that is solved effectively in the literature. The solution obtained is provably language-based minimal with respect to the system language.**

***Index Terms*—Decentralized decision making, discrete-event systems (DES), dynamic sensor activation.**

I. Introduction

DECISION making under limited sensor capacities is an important problem in networked automated systems. For example, in the fault diagnosis problem, the diagnosis module needs to infer the occurrence of faults based on its observations. In this paper, we investigate the decision-making problem in discrete-event systems (DES) that operate under dynamic observations. In this context, the system makes observations online through its sensors; these sensors can be turned ON/OFF dynamically during the evolution of the system according to a sensor activation policy that depends on the system's observations. Due to energy, bandwidth, or security constraints, sensors activations are "costly." Therefore, in order to reduce sensor-related costs, it is of interest to minimize, in some technical sense, the sensor activations while maintaining some desired observational property.

Manuscript received March 16, 2017; revised August 18, 2017; accepted December 1, 2017. Date of publication December 13, 2017; date of current version October 25, 2018. This work was supported in part by the U.S. National Science Foundation under Grant CCF-1138860 (Expeditions in Computing project ExCAPE: Expeditions in Computer Augmented Program Engineering) and Grant CNS-1446298. Recommended by Associate Editor S. Takai. *(Corresponding author: Xiang Yin.)*

X. Yin is with the Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: xiangyin@sjtu.edu.cn).

S. Lafortune is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109 USA (e-mail: stephane@umich.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2017.2783048

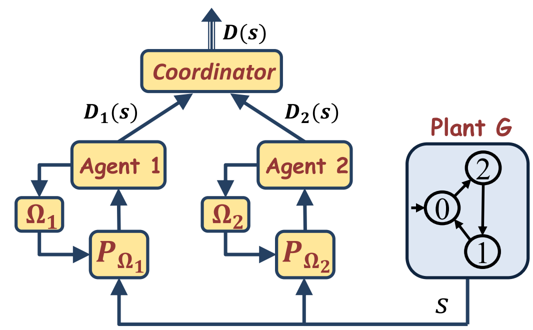


Fig. 1. Decentralized decision-making architecture with two agents, where Ω_i denotes Agent i's sensor activation policy and P_{Ω_i} denotes the information mapping induced by Ω_i.

In many large-scale systems, the information structure is decentralized due to the distributed nature of the sensors. Under the decentralized setting, the system is monitored by a set of agents that make local decisions and send them to a coordinator. Then, the coordinator makes a global decision based on the local decisions received from each local agent. This decentralized decision-making architecture is depicted in Fig. 1. In the context of DES, many different decentralized decision-making problems have already been studied. For example, in the decentralized supervisory control problem [1], [2], local supervisors need to disable/enable events dynamically in order to restrict the system such that some closed-loop behavior can be achieved. In the decentralized fault diagnosis problem considered in [3]–[5], the system is monitored by a set of local agents that work as a team in order to diagnose every occurrence of fault events. Similarly, in the decentralized fault prognosis problem [6], the local agents need to work as a team in order to predict every occurrence of fault events.

The problem of optimal sensor selection for *static observations* has been widely studied in the DES literature (see, e.g., [7]–[10]). The goal in these works was to find an optimal set of observable events that is fixed for the entire execution of the system and enforces a given DES-theoretic property. In the context of *dynamic observations*, where sensors can be turned ON/OFF dynamically, the corresponding problem of optimal sensor activation has also received a lot of attention in the literature;

see, e.g., [11]–[19] for a sample of this work and the recent survey paper [20] for an extensive bibliography. For example, in [11]–[13] and [17], the problem of centralized dynamic sensor activation for enforcement of different diagnosability properties was solved. In [11] and [14], dynamic sensor activation for the purpose of centralized control was also studied. Recently, a general framework that solves a class of centralized dynamic sensor activation problems was proposed [18].

However, for the decentralized sensor activation problem, there are very few results in the literature. In [13], the problem of dynamic sensor activation for decentralized diagnosis is studied. Specifically, a "window-based partition" approach is proposed in order to obtain a solution. The main drawback of this approach is that the solution obtained is only optimal with respect to (w.r.t.) a finite (restricted) solution space and may not be language-based optimal in general, where "language-based optimal" means that no sensor activation policy defined over the entire language domain can be strictly smaller (in terms of set inclusion) than the synthesized solution. In other words, by enlarging the solution space by refining the state space of the system model, solutions better than the solution found in [13] could be obtained in principle. Similarly, in [14], the problem of dynamic sensor activation for decentralized control is also studied, where the solution obtained is again optimal w.r.t. a finite solution space. To the best of our knowledge, the problem of *language-based* sensor optimization for *decentralized* diagnosis or control has remained an open problem, as is mentioned in the recent survey [20].

One important reason for the lack of results for the decentralized sensor activation problem is that decentralized multiplayer decision problems are conceptually much more difficult to solve than their corresponding centralized versions. In particular, to synthesize a strategy for one agent, we need to know the strategies of the other agents, which are to be synthesized and again depend on the unknown strategy of the first agent. In general, these types of problems have been discussed in the framework of team decision theory [21]. In the DES literature, it is well known that many problems that are decidable in the centralized setting become undecidable (e.g., the problem of synthesizing safe and nonblocking supervisors [22], [23]) or open (e.g., the problem of synthesizing maximally permissive safe supervisors [24]) in the decentralized case, even when only two agents are involved.

In this paper, we propose a new approach to tackle the problem of dynamic sensor activation for the purpose of decentralized decision making. The main contributions of this paper are as follows. First, we formulate a general class of decentralized decision-making problems called the *decentralized state disambiguation problem*. We propose the notion of *decentralized distinguishability*, which covers coobservability, K-codiagnosability, and coprognosability. Second, to solve the dynamic sensor activation problem, we adopt a *person-by-person* approach (see, e.g., [25] and the references therein) to decompose the decentralized minimization problem to two consecutive centralized minimization problems. We first minimize the sensor activation policy for Agent 1 by keeping the policy of Agent 2 fixed. Then, we fix Agent 1's sensor activation policy to the one obtained and solve the same minimization problem but for Agent 2. Essentially, we solve two centralized *constrained* minimization problems, since we need to take the other agent's information into account when we minimize the decisions of an agent. A novel approach is also proposed to reduce each centralized constrained minimization problem to a problem that is solved effectively by an algorithm presented in [18]. Moreover, we prove that the solution obtained by our procedure is minimal w.r.t. the system language (i.e., over an infinite set in general), in contrast to the works reviewed above where minimality was w.r.t. a finite solution space. As special cases of the proposed framework, *language-based* sensor optimizations for decentralized diagnosis and decentralized control, which were previously open, are solved. These language-based optimal decentralized solutions essentially come from the effective reduction from the decentralized problem to two fully centralized problems and the language-based optimal solution found in [18] for the fully centralized case. Finally, we show that the proposed framework is applicable to both the disjunctive architecture and the conjunctive architecture.

In general, a person-by-person approach in team decision problems may not terminate in a finite number of steps, since we may need to iterate between the two constrained minimization problems (see, e.g., [24]). However, since we consider a logical optimality criterion, our problem enjoys a *monotonicity property*, which "decouples" the minimization objective to some extent. As a consequence, we can use the person-by-person approach to effectively solve this problem and iteration is not needed. Such a monotonicity property does not hold in general for arbitrary decentralized synthesis problems, e.g., [22]–[24].

In the DES literature, the person-by-person approach has also been applied to many problems, e.g., the decentralized control problem [24] and the decentralized communication problem [26]–[28]. In particular, in [13] and [14], the person-by-person approach is also exploited for solving the decentralized sensor activation problem for the purposes of decentralized control and decentralized diagnosis, respectively. The differences between this paper and [13], [14] are as follows. First, in [31] and [32], the authors restrict the solution spaces of the minimization problems to finite domains. However, in general, the solution space of the decentralized minimization problem is infinite over the system's language. Therefore, one may find solutions that are better than those found in [13] and [14] but are not in the prespecified finite solution spaces. This infinite solution space is also the fundamental difficulty in solving the decentralized minimization problem. However, our approach does not need this restriction, and consequently, the solution obtained in this paper is *language-based* minimal. Moreover, the problem formulation in our paper is more general compared with those in [13] and [14]. Consequently, the results in this paper can be used to find language-based minimal solutions for a class of decentralized sensor activation problems under both the disjunctive and conjunctive architectures, while the results in [13] and [14] can only be applied to specific problems under the disjunctive architecture. However, due to the unrestricted solution space, our algorithm has a higher complexity than those in [13] and [14].

The remainder of this paper is organized as follows. Section II describes the model of the system under dynamic observations. In Section III, we formulate the decentralized state disambiguation problem and the decentralized minimization problem that we solve in this paper. Section IV shows how to solve the centralized constrained minimization problem by reducing it to a fully centralized problem. In Section V, we present our algorithm for synthesizing a minimal decentralized solution. In Section VI, we show how specific problems, e.g., sensor activation for decentralized diagnosis/control/prognosis, can be solved by the proposed framework. We also extend our results to the conjunctive architecture. Finally, we conclude the paper in Section VII. Preliminary and partial versions of some of the results in this paper are presented in [29]. However, Yin and Lafortune [29] only investigated K-codiagnosability, which is a special case of the general framework proposed in this paper. Moreover, the approach used in this paper to solve the constraint minimization problem is more efficient than the state-partition-automaton-based approach in [29].

II. Preliminaries

A. System Model

Let Σ be a finite set of events. A string is a finite sequence of events in Σ. We denote by Σ^* the set of all finite strings over Σ including the empty string ϵ. For any string $s \in \Sigma^*$, we denote by $|s|$ its length with $|\epsilon| = 0$. A language $L \subseteq \Sigma^*$ is a set of strings. We define $L/s := \{t \in \Sigma^* : st \in L\}$ as the set of continuations of string s in L. The prefix closure of language $L \subseteq \Sigma^*$ is $\overline{L} = \{s \in \Sigma^* : \exists w \in \Sigma^* \text{ s.t. } sw \in L\}$. We say that L is prefix-closed if $L = \overline{L}$. We say that language L is live if $\forall s \in L, \exists \sigma \in \Sigma : s\sigma \in L$, i.e., any string in L can be extended to arbitrarily long length.

We consider a DES modeled as a deterministic finite-state automaton $G = (Q, \Sigma, \delta, q_0)$, where Q is the finite set of states, Σ is the finite set of events, $\delta : Q \times \Sigma \to Q$ is the partial transition function, and q_0 is the initial state. The function δ is extended to $Q \times \Sigma^*$ in the usual way (see, e.g., [30]). The behavior of the system G starting from state $q \in Q$ is described by the prefix-closed language $\mathcal{L}(G, q) = \{s \in \Sigma^* : \delta(q, s)!\}$, where "!" means "is defined." For the sake of simplicity, we also write $\delta(q, s)$ as $\delta(s)$ and write $\mathcal{L}(G, q)$ as $\mathcal{L}(G)$ if $q = q_0$.

B. Information Mapping

We consider a general dynamic observation setting, where the observability properties of events can be controlled by a *sensor activation policy* during the evolution of the system. Let $\Sigma_o \subseteq \Sigma$ be the set of events that can become observable by activating some sensors. A sensor activation policy is defined as a pair $\Omega = (R, \Theta)$, where $R = (Q_R, \Sigma, \delta_R, q_{0,R})$ is a deterministic automaton such that $\mathcal{L}(R) = \Sigma^*$ and $\Theta : Q_R \to 2^{\Sigma_o}$ is a labeling function that specifies the current set of "observable" events within Σ_o. Specifically, for any $s \in \Sigma^*$, $\Theta(\delta_R(s))$ denotes the set of events that are *monitored* after the occurrence of s. While an event is monitored, any occurrence of it will be observed. In other words, after string s, events in $\Sigma_o \setminus \Theta(\delta_R(s))$ are currently "unobservable" (i.e., their sensors are turned OFF). Therefore, the codomain of L, i.e., Σ_o, is the set of potential observable events. To make Ω implementable, the pair (R, Θ) needs to satisfy the following conditions:

(C-1) $\mathcal{L}(R) = \Sigma^*$;
(C-2) $(\forall q, q' \in Q_R)(\forall \sigma \in \Sigma : \delta_R(q, \sigma) = q')[q \neq q' \Rightarrow \sigma \in \Theta(q)]$.

The above conditions say that the sensing decision can be updated (by updating the state of R) only when a monitored event occurs. Moreover, Ω can react to any execution of the system as $\mathcal{L}(G) \subseteq \mathcal{L}(R) = \Sigma^*$. In general, Q_R could be an infinite set. However, we will show later that the optimal sensor activation policies of interest in this paper can always be constructed with finite state spaces.

We say that the observations are *static* if the set of observable events is fixed *a priori*. We denote by Ω_{Σ_o} the corresponding sensor activation policy for the static observation with the set of observable events Σ_o. Specifically, $\Omega_{\Sigma_o} = (R, \Theta)$ is given by: 1) $Q_R = \{q_{0,R}\}$; 2) $\forall \sigma \in \Sigma_o : \delta_R(q_{0,R}, \sigma) = q_{0,R}$; and 3) $\Theta(q_{0,R}) = \Sigma_o$.

Given a sensor activation policy $\Omega = (R, \Theta)$, we define the corresponding information mapping $P_\Omega : \mathcal{L}(G) \to \Sigma_o^*$ recursively as follows:

$$P_\Omega(\epsilon) = \epsilon, \quad P_\Omega(s\sigma) = \begin{cases} P_\Omega(s)\sigma, & \text{if } \sigma \in \Theta(\delta_R(s)) \\ P_\Omega(s), & \text{if } \sigma \notin \Theta(\delta_R(s)). \end{cases}$$

That is, $P_\Omega(s)$ is the observation of string s under Ω. For any language $L \subseteq \Sigma^*$, we define $P_\Omega(L) = \{P_\Omega(s) \in \Sigma_o^* : \exists s \in L\}$.

For any two sensor activation policies $\Omega = (R, \Theta)$ and $\Omega' = (R', \Theta')$, we write that $\Omega' \subseteq \Omega$ if

$$\forall s \in \mathcal{L}(G) : \Theta'(\delta_{R'}(s)) \subseteq \Theta(\delta_R(s)) \tag{1}$$

and write that $\Omega' \subset \Omega$ if

$$[\Omega' \subseteq \Omega] \wedge [\exists s \in \mathcal{L}(G) : \Theta'(\delta_{R'}(s)) \neq \Theta(\delta_R(s))]. \tag{2}$$

C. State Estimate

Let $s \in \mathcal{L}(G)$ be a string generated by the system. We denote by $\mathcal{E}_\Omega^G(s) \subseteq Q$ the *state estimate* upon the occurrence of s w.r.t Ω and the state space of G. Specifically, for any $s \in \mathcal{L}(G)$, we have

$$\mathcal{E}_\Omega^G(s) := \{\delta(t) \in Q : \exists t \in \mathcal{L}(G) \text{ s.t. } P_\Omega(s) = P_\Omega(t)\}.$$

Clearly, if $P_\Omega(s) = P_\Omega(t)$, then $\mathcal{E}_\Omega^G(s) = \mathcal{E}_\Omega^G(t)$. To compute $\mathcal{E}_\Omega^G(s)$, we can construct the observer of G. Let $\Omega = (R, \Theta), R = (Q_R, \Sigma, \delta_R, q_{0,R})$ be a sensor activation policy. The observer for G under Ω is

$$\text{Obs}_\Omega(G) = (X, \Sigma_o, f, x_0) \tag{3}$$

where $X \subseteq 2^Q \times Q_R$ is the state space, and for any state $x \in X$, we write $x = (I(x), R(x))$, where $I(x) \in 2^Q$ and $R(x) \in Q_R$. The partial transition function of the observer $f : X \times \Sigma_o \to X$ is defined as follows: for any $x = (\imath, q), x' = (\imath', q') \in X$ and

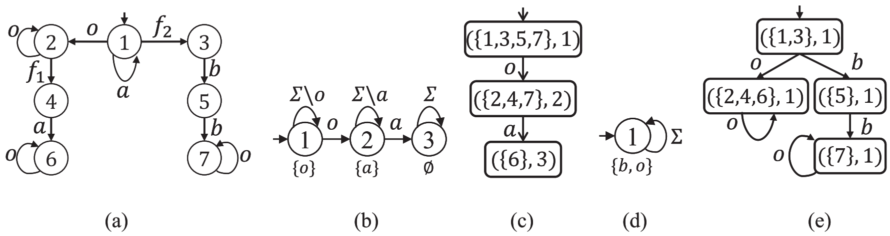


Fig. 2. Examples of sensor activation policies and observers. (a) System G. (b) Ω_1. (c) Obs$_{\Omega_1}(G)$. (d) Ω_2. (e) Obs$_{\Omega_2}(G)$.

$\sigma \in \Theta(q) \subseteq \Sigma_o$, $f(x,\sigma) = x'$ iff

$$\begin{cases} q' = \delta_R(q,\sigma) \\ \imath' = UR_{\Theta(q')}(Next_\sigma(\imath)) \end{cases} \tag{4}$$

where for any $\imath \in 2^Q$, $\sigma \in \Sigma_o$, and $\theta \in 2^{\Sigma_o}$, we have

$$\text{Next}_\sigma(\imath) = \{q_1 \in Q : \exists q_2 \in \imath \text{ s.t. } \delta(q_2,\sigma) = q_1\}$$

$$\text{UR}_\theta(\imath) = \{q_1 \in Q : \exists q_2 \in \imath, \exists s \in (\Sigma \setminus \theta)^* \text{ s.t. } \delta(q_2,s) = q_1\}.$$

Intuitively, Next$_\sigma(\imath)$ is the set of states that can be reached from some state in $\imath$ immediately after observing σ and UR$_\theta(\imath)$ is the set of states that can be reached unobservably from some state in $\imath$ under the set of monitored events θ. Finally, the initial state of Obs$_\Omega(G)$ is $x_0 = (\text{UR}_{\Theta(q_{0,R})}(\{q_0\}), q_{0,R})$. Then, the state estimate $\mathcal{E}^G_\Omega(s)$ can be computed by $I(f(P_\Omega(s))) = \mathcal{E}^G_\Omega(s)$, i.e., the state components of the observer state reached upon $P_\Omega(s)$ is the state estimator value after s. Also, if $\Omega' \subseteq \Omega$, then we have $\forall s \in \mathcal{L}(G) : \mathcal{E}^G_\Omega(s) \subseteq \mathcal{E}^G_{\Omega'}(s)$ [17].

Example 1: Consider the system G in Fig. 2(a). Let $\Sigma_{o,1} = \{o,a\}$ and $\Sigma_{o,2} = \{o,b\}$ be two sets of observable events. As shown in Fig. 2(b), Ω_1 is a sensor activation policy with the set of observable events $\Sigma_{o,1}$. The labeling function is specified by the set of events associated with each state in the figure. Initially, event o is monitored by Ω_1. Once o is observed, Ω_1 changes to monitor event a. Finally, Ω_1 turns all sensors OFF when a is observed. The corresponding observer Obs$_{\Omega_1}(G)$ is shown in Fig. 2(c). For example, for the string $oof_1a \in \mathcal{L}(G)$, we have that $P_{\Omega_1}(oof_1a) = oa$ and $I(f(oa)) = \{6\} = \mathcal{E}^G_\Omega(oof_1a)$. Similarly, Fig. 2(b) shows a sensor activation policy Ω_2 with the set of observable events $\Sigma_{o,2}$. Clearly, Ω_2 always monitors all events in $\Sigma_{o,2}$, i.e., $\Omega_2 = \Omega_{\Sigma_{o,2}}$. Therefore, the observer Obs$_{\Omega_2}(G)$ shown in Fig. 2(c) is the standard observer (see, e.g., [30]) if we ignore the second component of each state.

III. Decentralized State Disambiguation Problem

In this section, we first define the notion of decentralized distinguishability. Then, we formulate the decentralized sensor minimization problem for the purpose of state disambiguation.

A. Decentralized Distinguishability

In the decentralized decision-making problem, at each instant, each local agent sends highly compressed information, i.e., a local decision, to the coordinator based on its local (dynamic) observation. Then, the coordinator makes a global decision based on the information received from each local agent. Let $\mathcal{I}$ be the index set of local agents. For each agent $i \in \mathcal{I}$, we denote by Ω_i its sensor activation policy and by $\Sigma_{o,i}$ the set of events that can be monitored in Ω_i. For the sake of simplicity, we develop all results hereafter for the case of two agents, i.e., $\mathcal{I} = \{1,2\}$. The principle can be extended to an arbitrary number of agents. We define the pair of sensor activation policies as $\bar{\Omega} = [\Omega_1, \Omega_2]$.

In order to formulate the decentralized decision-making problem, we need to specify the following three ingredients.

1) What requirement the global decision has to fulfill?
2) What information each local agent can send to the coordinator?
3) What rule the coordinator uses to calculate a global decision based on the local decisions?

Hereafter, we refer to the first ingredient as the *specification* of the decentralized decision-making problem. The last two ingredients are referred to as the *architecture* of the decentralized decision-making problem.

Several different specifications have been studied separately in the literature for decentralized decision-making problems, e.g., to diagnose every occurrence of fault events [3], [4], to predict every occurrence of fault events [6] or to control the system [1], [2]. In this paper, we do not study a specific specification. Instead, we define a general class of specifications called *decentralized state disambiguation*. As shown later in Section VI, many existing decentralized decision-making problems are special cases of the decentralized disambiguation problem. Formally, we define a specification as a pair of state sets

$$T = Q^T_A \times Q^T_B \subseteq Q \times Q. \tag{5}$$

Intuitively, specification T is used to capture the following requirement. State set Q^T_A represents the set of states at which the global system *must* take some desired action associated to T and state set Q^T_B represents the set of states at which the global system *should not* take such an action. Then, the system must be able to distinguish between states in Q^T_A and states in Q^T_B (under certain decentralized architecture, which will be specified later) when a state in Q^T_A is reached; otherwise, the desired action associated to T cannot be taken safely.

Regarding the architecture of the decentralized decision-making problem, here, we consider the following mechanism, which is widely used in the literature for many different problems [1]–[4], [6]. We assume that communication between each agent and the coordinator is costly, and only a binary decision is allowed for each agent at each instant. That is, each local agent can only send to the coordinator a highly compressed decision "1" or "0," which correspond to "take the action" and "do not take the action," respectively. Then, the coordinator has two possible *fusion rules* to obtain a global decision from local decisions.

1) The disjunctive rule: issue "1" globally if and only if *one* local agent issues "1."
2) The conjunctive rule: issue "1" globally if and only if *all* local agents issue "1."

Hereafter, we will develop the main results based on the disjunctive rule. We will discuss how to extend our results to the conjunctive case in Section VI-D.

In general, the system may have multiple distinct objectives, i.e., it needs to distinguish different states pairs for different purposes. For the sake of generality, we consider m specifications and denote by $\mathbb{T} = \{T_1, \ldots, T_m\}$ the set of specifications, where $T_k = Q_A^{T_k} \times Q_B^{T_k} \subseteq Q \times Q, T_k \in \mathbb{T}$. Also, for the sake of generality, for each $T_k \in \mathbb{T}$, we define $\mathcal{I}_{T_k} \subseteq \mathcal{I}$ as the nonempty set of agents that can contribute to the decision associated to T_k. If $\mathcal{I}_{T_k}$ is a singleton, then the global decision will be "1" if the unique agent in $\mathcal{I}_{T_k}$ issues "1." However, in the case that $|\mathcal{I}_{T_k}| > 1$, since we consider the disjunctive architecture, the global decision will be "1" if *one* agent in $\mathcal{I}_{T_k}$ issues "1." Therefore, an agent must be able to distinguish any states pair in T_k *unambiguously* when it issues "1"; otherwise, a wrong global decision may be made. This observation leads to the following definition of *decentralized distinguishability*.

Definition 1: (Decentralized distinguishability). Let G be the system, $\mathbb{T} = \{T_1, \ldots, T_m\}$ be a set of specifications, and $\bar{\Omega} = [\Omega_1, \Omega_2]$ be a pair of sensor activation policies. We say that G is decentralized distinguishable w.r.t. $\bar{\Omega}$ and $\mathbb{T}$ if

$$(\forall T_k \in \mathbb{T})(\forall s \in \mathcal{L}(G) : \delta(s) \in Q_A^{T_k})(\exists i \in \mathcal{I}_{T_k})$$
$$[\mathcal{E}_{\Omega_i}^G(s) \cap Q_B^{T_k} = \emptyset] \tag{6}$$

Intuitively, the above definition says the following. For any specification $T_k \in \mathbb{T}$, for any string that goes to a state in $Q_A^{T_k}$, i.e., a state at which we must take the action associated to T_k, there must exist at least one local agent in $\mathcal{I}_{T_k}$ that knows *for sure* that we can take such an action. Note that, in our setting, only $Q_B^{T_k}$ are the set of states at which we cannot take the action associated to T_k. In other words, there is no harm in taking the action if the system is in $Q \setminus (Q_A^{T_k} \cup Q_B^{T_k})$. This is why, we require $\mathcal{E}_{\Omega_i}^G(s) \cap Q_B^{T_k} = \emptyset$ rather than $\mathcal{E}_{\Omega_i}^G(s) \subseteq Q_A^{T_k}$. We will show later in Section VI that K-codiagnosability, coobservability, and coprognosability are all instances of decentralized distinguishability. Note that, if $Q_A^{T_k} \cap Q_B^{T_k} \neq \emptyset$ for some $T_k \in \mathbb{T}$, then G will not be decentralized distinguishable for any sensor activation policies $\bar{\Omega}$. This phenomenon may occur in the fault prognosis problem as we will discuss later in Section VI-C.

Example 2: We still consider the system G in Fig. 2(a), and $\Sigma_{o,1} = \{o, a\}$ and $\Sigma_{o,2} = \{o, b\}$ are two sets of observable events. We assume that the observations are static, i.e., $\Omega_1 = \Omega_{\Sigma_{o,1}}$ and $\Omega_2 = \Omega_{\Sigma_{o,2}}$. Let us consider the following set of specifications: $\mathbb{T} = \{T_1, T_2\}$, where

$$T_1 = Q_A^{T_1} \times Q_B^{T_1} = \{6\} \times \{1, 2, 3, 5, 7\}$$
$$T_2 = Q_A^{T_2} \times Q_B^{T_2} = \{5, 7\} \times \{1, 2, 4, 6\}$$

and $\mathcal{I}_{T_1} = \mathcal{I}_{T_2} = \{1, 2\}$. We can verify that G is decentralized distinguishable w.r.t. $\{T_1, T_2\}$ and $[\Omega_{\Sigma_{o,1}}, \Omega_{\Sigma_{o,2}}]$. For example, for specification T_1 and string of_1a such that $\delta(of_1a) = 6 \in Q_A^{T_1}$, we have $1 \in \mathcal{I}_{T_1}$ and $\mathcal{E}_{\Omega_{\Sigma_{o,1}}}^G(of_1a) \cap Q_B^{T_1} = \{6\} \cap \{1, 2, 3, 5, 7\} = \emptyset$. However, if we add another specification $T_3 = \{4\} \times \{1, 2\}$ to $\{T_1, T_2\}$, then G will not be decentralized distinguishable. For example, for $\delta(of_1) = 4 \in Q_A^{T_3}$, we have $\mathcal{E}_{\Omega_{\Sigma_{o,1}}}^G(of_1) \cap Q_B^{T_3} = \{2, 4\} \cap \{1, 2\} \neq \emptyset$ and $\mathcal{E}_{\Omega_{\Sigma_{o,2}}}^G(of_1) \cap Q_B^{T_3} = \{2, 4, 6\} \cap \{1, 2\} \neq \emptyset$, i.e., none of the agents can distinguish specification T_3.

Remark 1: The state disambiguation problem and its sensor activation have been studied in the literature in the centralized setting (see, e.g., [17], [31], and [32]). Compared to its centralized counterpart, the decentralized disambiguation problem has the following important difference. In the centralized setting, specification $Q_A \times Q_B$ and specification $Q_B \times Q_A$ are equivalent in the sense that if the system can distinguish state q_1 from state q_2, then it can also distinguish q_2 from q_1. However, it is not the case in the decentralized setting, and we cannot swap Q_A and Q_B arbitrarily. One can easily verify that G is decentralized distinguishable w.r.t. $Q_A \times Q_B$ does not necessarily imply that it is decentralized distinguishable w.r.t. $Q_B \times Q_A$. Moreover, our procedure for solving the sensor activation problem in the decentralized setting is completely different from those in the centralized case.

B. Problem Formulation and Solution Overview

Let $\mathbb{T}$ be the set of specifications. Then, the goal of the sensor activation problem is to find an *optimal* pair of sensor activation policies $\bar{\Omega} = [\Omega_1, \Omega_2]$ such that the system is decentralized distinguishable w.r.t. $\bar{\Omega}$ and $\mathbb{T}$. In this paper, we consider the logical optimality criterion that is widely used in the literature [13], [14], [20]. Specifically, for any $\bar{\Omega} = [\Omega_1, \Omega_2]$ and $\bar{\Omega}' = [\Omega_1', \Omega_2']$, the inclusion $\bar{\Omega}' \subseteq \bar{\Omega}$ means that

$$\forall i \in \mathcal{I} : \Omega_i' \subseteq \Omega_i \tag{7}$$

and the strict inclusion $\bar{\Omega}' \subset \bar{\Omega}$ means that

$$[\bar{\Omega}' \subseteq \bar{\Omega}] \wedge [\exists i \in \mathcal{I} : \Omega_i' \subset \Omega_i]. \tag{8}$$

We are now ready to formulate the problem of minimal sensor activation for decentralized state disambiguation.

Problem 1: Let G be the system and $\mathbb{T} = \{T_1, \ldots, T_m\}$ be a set of specifications. For each agent $i \in \{1, 2\}$, let $\Sigma_{o,i} \subseteq \Sigma$ be the set of observable events. Find sensor activation policies $\bar{\Omega}^* = [\Omega_1^*, \Omega_2^*]$ such that we have the following.

C1. G is decentralized distinguishable w.r.t. $\bar{\Omega}^*$ and $\mathbb{T}$.
C2. $\bar{\Omega}^*$ is minimal, i.e., there does not exist another $\bar{\Omega}' \subset \bar{\Omega}^*$ that satisfies (C1).

Remark 2: In [13] and [14], "suboptimal" solutions to two special cases of Problem 1, the decentralized control problem and the decentralized diagnosis problem, are provided, in the sense that the solutions found therein are minimal among all solutions over *given* finite restricted solution spaces. In principle, the solutions found in [13] and [14] could be improved by employing finer partitions and repeating the optimization procedure. In this paper, we are aiming for a *language-based* minimal solution, in the sense that the notion of strict inclusion of sensor activation policies is defined in terms of the strings in $\mathcal{L}(G)$ [see (2) and (8)]. In other words, we do not impose, *a priori*, any constraints on the solution space of each Ω_i. Hence, no better solution can be obtained by refining the state space of G and repeating the solution procedure. To the best of our knowledge, such a language-based optimal solution to the decentralized sensor activation problem has never been reported in the literature. Moreover, Problem 1 is more general than the problems studied in [13] and [14].

Before we formally tackle Problem 1, let us first provide a brief overview of our solution approach. We adopt the person-by-person approach that has been widely used in decentralized optimization problems. Specifically, we decompose the decentralized minimization problem to a set of centralized constrained minimization problems, and for each such problem, we only attempt to minimize one agent's sensor activation policy, while the other one is fixed. However, the following questions arise. First, by taking the person-by-person approach, iterations involving minimization for each agent may be required in general, and such iterations may not terminate in a finite number of steps. We will show that in our particular problem, such iterations are not required. This is due to the so-called *monotonicity property* that arises in dynamic sensor activation problems. The second question of interest is how to minimize the sensor activation policy of one agent when the policy of the other agent is fixed. This problem is different from the fully centralized minimization problem, since we should not only consider the information of the agent whose sensor activation policy we are minimizing, but we must also take into account the information available to the other agent, whose sensor activation policy is fixed. Therefore, the true information state (IS) for this minimization problem consist of: 1) the knowledge of the agent whose sensor activation policy is being minimized; and 2) *this agent's inference of the other agent's potential knowledge of the system based on that agent's own information*. To resolve this information dependence, we develop a novel approach, by which we encode the second agent's knowledge into the system model. This is discussed in the next section.

IV. Constrained Minimization Problem

In this section, we tackle the problem of minimizing the sensor activation policy for one agent when the sensor activation policy of the other one is fixed. This problem is also referred to as the *centralized constrained minimization problem* herafter. Throughout this section, $i \in \{1, 2\}$ denotes the agent whose sensor activation policy is being minimized, while $j \in \{1, 2\}, j \neq i$, denotes the other agent whose sensor activation policy is fixed.

A. Constrained Minimization Problem

Problem 2: (Centralized constrained minimization problem). Let G be the system and $\mathbb{T} = \{T_1, \ldots, T_m\}$ be a set of specifications. Let $i, j \in \{1, 2\}, i \neq j$ be two agents. Suppose that the sensor activation policy Ω_j for Agent j is fixed. Find a sensor activation policy Ω_i for Agent i such that we have the following.

C1. G is decentralized distinguishable w.r.t. $[\Omega_1, \Omega_2]$ and $\mathbb{T}$.

C2. For any Ω_i' satisfying (C1), we have $\Omega_i' \not\subset \Omega_i$.

The above problem is different from both the centralized and decentralized minimization problems. In the centralized minimization problem, where only one agent is involved, to maintain distinguishability, we need to require that

$$\forall T_k \in \mathbb{T}, \forall s \in \mathcal{L}(G) : (\mathcal{E}_\Omega^G(s) \times \mathcal{E}_\Omega^G(s)) \cap T_k = \emptyset$$

where Ω is the centralized sensor activation policy. In other words, the agent should always be able to distinguish states in $Q_A^{T_k}$ from states in $Q_B^{T_k}$ for any $T_k \in \mathbb{T}$. However, in the decentralized disambiguation problem, it is possible that there exists a string $s \in \mathcal{L}(G), \delta(s) \in Q_A^{T_k}$ such that $\mathcal{E}_{\Omega_i}^G(s) \cap Q_B^{T_k} \neq \emptyset$, but $\mathcal{E}_{\Omega_j}^G(s) \cap Q_B^{T_k} = \emptyset$, where $j \in \mathcal{I}_{T_k}$. Therefore, Agent j may "help" Agent i to resolve the ambiguity. In other words, to solve the constrained minimization problem for one agent, we must take the other agent's sensor activation policy into account.

B. Problem Reduction

First, we recall a general class of fully centralized sensor activation problems that is studied in [18].

Problem 3: (Centralized sensor minimization problem for IS-based property). Let $G = (Q, \Sigma, \delta, q_0)$ be the system and $\varphi : 2^Q \to \{0, 1\}$ be a function. Find a sensor activation policy Ω such that we have the following.

C1. $\forall s \in \mathcal{L}(G) : \varphi(\mathcal{E}_\Omega^G(s)) = 1$.

C2. For any Ω' satisfying (C1), we have $\Omega' \not\subset \Omega$.

Problem 3 is a fully centralized sensor activation problem, since only one agent is involved. In particular, function $\varphi : 2^Q \to \{0, 1\}$ is referred to as an *IS-based property*. This problem is studied in more detail in [18], where an algorithm is provided that solves this problem effectively by returning a finite sensor activation policy satisfying the requirements. Note that the algorithm in [18] also guarantees by construction that the synthesized sensor activation policy satisfies the implementation conditions (C-1) and (C-2). In general, a minimal sensor activation policy does not exist for an arbitrary property, e.g., detectability or diagnosability without a prespecified delay. However, for an IS-based property, a minimal sensor activation policy does exist, and it is finitely realizable; this is because an IS-based property can be checked over the state estimates of the system. Therefore, if we can reduce Problem 2 to Problem 3, then it means that Problem 2 can also be solved effectively, and the solution will be finitely realizable. We now show that such a reduction is possible by using automata V and $\tilde{V}$, which are defined next.

Let G be the system and Ω_j be the fixed sensor activation policy, where $\Omega_j = (R_j, \Theta_j)$ and $R_j = (Q_R^j, \Sigma, \delta_R^j, q_{0,R}^j)$. We define a new automaton

$$V = (Q_V, \delta_V, \Sigma_V, q_{0,V}) \tag{9}$$

where $Q_V \subseteq Q \times Q_R^j \times Q \times Q_R^j$ is the set of states, $\Sigma_V = (\Sigma \cup \{\epsilon\}) \times (\Sigma \cup \{\epsilon\})$ is the set of events, and $q_{0,V} = (q_0, q_{0,R}^j, q_0, q_{0,R}^j)$ is the initial state.

The transition function $\delta_V : Q_V \times \Sigma_V \rightarrow Q_V$ is defined as follows: for any (q_1, q_1^R, q_2, q_2^R) and $\sigma \in \Sigma$, the following transitions are defined.

1) If $\sigma \in \Theta_j(q_1^R)$ and $\sigma \in \Theta_j(q_2^R)$, then
$$\delta_V((q_1, q_1^R, q_2, q_2^R), (\sigma, \sigma)) = (\delta(q_1, \sigma), \delta_R^j(q_1^R, \sigma), \delta(q_2, \sigma), \delta_R^j(q_2^R, \sigma)).$$

2) If $\sigma \in \Theta_j(q_1^R)$ and $\sigma \notin \Theta_j(q_2^R)$, then
$$\delta_V((q_1, q_1^R, q_2, q_2^R), (\epsilon, \sigma)) = (q_1, q_1^R, \delta(q_2, \sigma), \delta_R^j(q_2^R, \sigma)).$$

3) If $\sigma \notin \Theta_j(q_1^R)$ and $\sigma \in \Theta_j(q_2^R)$, then
$$\delta_V((q_1, q_1^R, q_2, q_2^R), (\sigma, \epsilon)) = (\delta(q_1, \sigma), \delta_R^j(q_1^R, \sigma), q_2, q_2^R).$$

4) If $\sigma \notin \Theta_j(q_1^R)$ and $\sigma \notin \Theta_j(q_2^R)$, then
$$\delta_V((q_1, q_1^R, q_2, q_2^R), (\sigma, \epsilon)) = (\delta(q_1, \sigma), \delta_R^j(q_1^R, \sigma), q_2, q_2^R)$$
$$\delta_V((q_1, q_1^R, q_2, q_2^R), (\epsilon, \sigma)) = (q_1, q_1^R, \delta(q_2, \sigma), \delta_R^j(q_2^R, \sigma)).$$

The above construction follows the well-known $\mathcal{M}$-machine (or twin-plant) construction that was originally used for the verification of (co)observability [33]–[35], but we generalize it to the dynamic observation setting. Essentially, V tracks a pair of strings that look the same for Agent j under Ω_j. Specifically, the first two components are used to track a string in the original system, and the last two components are used to track a string that looks the same as the first string. Since we are considering the dynamic observation setting, we also need to track states in the sensor activation policy in order to determine the set of monitored events, this is why the second (respectively, fourth) component always moves together with the first (respectively, third) component. Therefore, for any $(s_1, s_2) \in \mathcal{L}(V)$, we have that $P_{\Omega_j}(s_1) = P_{\Omega_j}(s_2)$. Similarly, for any $t, w \in \mathcal{L}(G)$ such that $P_{\Omega_j}(t) = P_{\Omega_j}(w)$, there exists $(s_1, s_2) \in \mathcal{L}(V)$ such that $s_1 = t$ and $s_2 = w$, i.e., state $(\delta(t), \delta_R^j(t), \delta(w), \delta_R^j(w))$ is reachable in V.

Next, we modify V as follows. For each transition in V, we have the following.

1) If the event is in the form of (σ, σ) or (σ, ϵ), then we replace the event by σ.
2) If the event is in the form of (ϵ, σ), then we replace the event by ϵ.

We denote by $\tilde{V} = (Q_{\tilde{V}}, \delta_{\tilde{V}}, \Sigma_{\tilde{V}}, q_{0,\tilde{V}})$ the modified automaton. Similar modification was also used in [36] and [37] in the static observation setting for different purpose. Intuitively, $\tilde{V}$ only keeps the first component of the event of each transition in V, since this part corresponds to the transition in the real system. Note that $\tilde{V}$ is a nondeterministic automaton, since ϵ-transition is allowed. Therefore, $\delta_{\tilde{V}}(s)$ is the *set* of states that can be reached from $q_{0,\tilde{V}}$ via s.

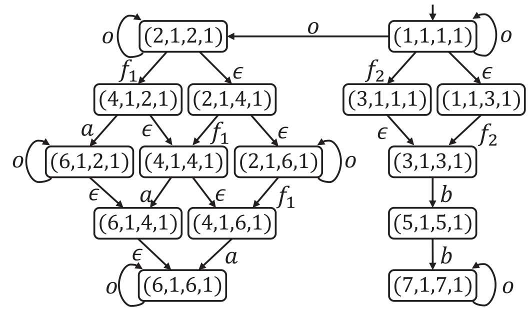


Fig. 3. Automaton $\tilde{V}$.

The modified automaton $\tilde{V}$ has the following properties. First, we have that $\mathcal{L}(\tilde{V}) = \mathcal{L}(G)$. Clearly, $\mathcal{L}(\tilde{V}) \subseteq \mathcal{L}(G)$ since a transition in $\tilde{V}$ is defined only when the corresponding transition in G is defined. Also, for any string $s \in \mathcal{L}(G)$, we know that $(s, s) \in \mathcal{L}(V)$, which implies that $s \in \mathcal{L}(\tilde{V})$. Second, for any $s \in \mathcal{L}(\tilde{V}) = \mathcal{L}(G)$, we know that

$$\begin{aligned}\delta_{\tilde{V}}(s) &= \{(\delta(s), \delta_R^j(s), \delta(t), \delta_R^j(t)) \in Q_{\tilde{V}} : (s,t) \in \mathcal{L}(V)\}\\ &= \{(\delta(s), \delta_R^j(s), \delta(t), \delta_R^j(t)) \in Q_{\tilde{V}} : t \in \mathcal{L}(G) \wedge P_{\Omega_j}(s) = P_{\Omega_j}(t)\}.\end{aligned} \tag{10}$$

Therefore, for any string $s \in \mathcal{L}(G) = \mathcal{L}(\tilde{V})$, if $(q_1, q_1^R, q_2, q_2^R) \in \delta_{\tilde{V}}(s)$, then it implies that $\delta(s) = q_1$ and state q_2 cannot be distinguished from q_1 under Ω_j. For any $x \in 2^{Q_{\tilde{V}}}$, we denote by $I_1(x) = \{q_1 \in Q : (q_1, q_1^R, q_2, q_2^R) \in x\}$ the set of states in the first component of x. Then, for any sensor activation policy Ω, by (10), we have $\mathcal{E}_\Omega^G(s) = I_1(\mathcal{E}_\Omega^{\tilde{V}}(s))$ for any $s \in \mathcal{L}(G)$.

Example 3: Let us still consider the system G shown in Fig. 2(a). Suppose that the fixed Ω_j is the sensor activation policy Ω_2 shown in Fig. 2(d), i.e., Ω_j always monitors o and b. Then, automaton $\tilde{V}$ constructed from G and Ω_j is shown in Fig. 3. Clearly, we see that $\mathcal{L}(\tilde{V}) = \mathcal{L}(G)$. For string $of_1a \in \mathcal{L}(\tilde{V}) = \mathcal{L}(G)$, we have that $\delta_{\tilde{V}}(of_1a) = \{(6,1,2,1), (6,1,4,1), (6,1,6,1)\}$ and $I_1(\mathcal{E}_{\Omega_j}^{\tilde{V}}(of_1a)) = I_1(\{(2,1,2,1), (4,1,2,1), (6,1,2,1), (2,1,4,1), (4,1,4,1), (2,1,6,1), (4,1,6,1), (6,1,6,1)\}) = \{2,4,6\} = \mathcal{E}_{\Omega_j}^G(of_1a)$.

Now, let us show how to use $\tilde{V}$ to reduce the constraint minimization problem, i.e., Problem 2, to a fully centralized minimization problem, i.e., Problem 3. First, we define the distinguishability function $DF : 2^{Q_{\tilde{V}}} \rightarrow \{0,1\}$ as follows: for each $x \in 2^{Q_{\tilde{V}}}$, we have

$$DF(x) = \begin{cases} 1, & \text{if } \forall T_k \in \mathbb{T} : \text{(c-i) or (c-ii) holds} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where conditions (c-i) and (c-ii) are defined as follows.

(c-i) $i \in \mathcal{I}_{T_k}$ and $(I_1(x) \times I_1(x)) \cap T_k = \emptyset$.
(c-ii) $j \in \mathcal{I}_{T_k}$ and $\forall q_1 \in I_1(x) \cap Q_A^{T_k}, \forall (q_1, q_1^R, q_2, q_2^R) \in x : (q_1, q_2) \notin T_k$.

Let us explain the intuition of the above two conditions in function DF. Suppose that Ω_i is the sensor activation policy to be synthesized for Agent i. Let $s \in \mathcal{L}(G)$ be a string such that $\delta(s) \in Q_A^{T_k}$, i.e., the coordinator must take the action associated to T_k when s is executed. Then, $\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)$ is the state estimate w.r.t. the state space of $\tilde{V}$ under Ω_i. Essentially, the function DF evaluates whether or not decentralized distinguishability is fulfilled by checking whether or not $x := \mathcal{E}_{\Omega_i}^{\tilde{V}}(s)$ satisfies conditions (c-i) and (c-ii), which can be interpreted as follows.

1) If (c-i) holds, then we know that Agent i can contribute to the global decision associated to T_k, since $i \in \mathcal{I}_{T_k}$. Moreover, it can contribute to the right decision since it knows for sure that the action associated to T_k has to be taken, since $(\mathcal{E}_{\Omega_i}^G(s) \times \mathcal{E}_{\Omega_i}^G(s)) \cap T_k = \emptyset$. Therefore, the disambiguation requirement is fulfilled even without looking at Agent j.
2) If (c-i) does not hold, then we know that either Agent i cannot contribute to the global decision associated to T_k or Agent i cannot make a right decision due to states ambiguity, i.e., $\exists q_1, q_2 \in \mathcal{E}_{\Omega_i}^G(s) = I_1(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) : (q_1, q_2) \in T_k$. In order to issue the right global decision, Agent j must be able to help Agent i to distinguish those ambiguous strings, i.e., condition (c-ii) needs to hold. First, Agent j should be able to contribute to the global decision associated with T_k, i.e., $j \in \mathcal{I}_{T_k}$. Then, for any string t that looks the same as s for Agent i and leads to a state in $Q_A^{T_k}$, there should not exist another string w that looks the same as t for Agent j and leads to a state in $Q_B^{T_k}$. Recall that $\tilde{V}$ is constructed by tracking all states that cannot be distinguished from q_1 by Agent j. Therefore, Agent i can *infer* which states Agent j cannot distinguish by using $\tilde{V}$. Specifically, if for any $(q_1, q_1^R, q_2, q_2^R) \in x : (q_1, q_2) \notin T_k$, then we know that there is no such a string w that can confuse Agent j for some string t, i.e., Agent j can make a right decision associated to T_k.

Finally, we would like to remark that, although specification $\mathbb{T}$ is defined over the state space of G, the distinguishability function DF is defined over the state space of $\tilde{V}$, i.e., we need to solve Problem 3 for the modified system $\tilde{V}$. However, this is not a problem, since the first component of a state $\tilde{V}$ exactly carries the same state information in G, i.e., $I_1(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) = \mathcal{E}_{\Omega_i}^G(s)$ for any Ω_i. Moreover, since $\mathcal{L}(\tilde{V}) = \mathcal{L}(G)$, we know that $\tilde{V}$ and G have the same observable behavior under any sensor activation policy. Therefore, we can first use $\tilde{V}$ to synthesize a sensor activation policy and then use it to monitor G.

We summarize the above discussions by the following theorem.

Theorem 1: Let G be the system and $\mathbb{T} = \{T_1, \dots, T_m\}$ be a set of specifications. Let $\tilde{V}$ be the automaton constructed based on Ω_j. Then, G is decentralized distinguishable w.r.t. $[\Omega_1, \Omega_2]$ and $\mathbb{T}$ if and only if

$$\forall s \in \mathcal{L}(G) : DF(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) = 1 \tag{12}$$

Proof: ($\Leftarrow$) By contraposition. Suppose that $\mathcal{L}(G)$ is not decentralized distinguishable. Then, there exists $T_k \in \mathbb{T}$, such that

$$(\exists s \in \mathcal{L}(G) : \delta(s) \in Q_A^{T_k})(\forall p \in \mathcal{I}_{T_k})[\mathcal{E}_{\Omega_p}^G(s) \cap Q_B^{T_k} \neq \emptyset]. \tag{13}$$

Let us consider the following three cases for $\mathcal{I}_{T_k}$.

Case 1: $\mathcal{I}_{T_k} = \{i\}$.

Let us consider (c-i), since (c-ii) is violated directly. By (13), since $\delta(s) \in Q_A^{T_k} \cap \mathcal{E}_{\Omega_i}^G(s)$ and $Q_B^{T_k} \cap \mathcal{E}_{\Omega_i}^G(s) \neq \emptyset$, we have

$$(I_1(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) \times I_1(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s))) \cap T_k = (\mathcal{E}_{\Omega_i}^G(s) \times \mathcal{E}_{\Omega_i}^G(s)) \cap T_k \neq \emptyset.$$

Therefore, (c-i) is also violated and $DF(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) = 0$.

Case 2: $\mathcal{I}_{T_k} = \{j\}$.

Let us consider (c-ii), since (c-i) is violated directly. We still consider string s in (13). We have $\delta(s) \in \mathcal{E}_{\Omega_i}^G(s) \cap Q_A^{T_k} = I_1(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) \cap Q_A^{T_k}$. Since $\mathcal{E}_{\Omega_j}^G(s) \cap Q_B^{T_k} \neq \emptyset$, there exists a string $t \in \mathcal{L}(G)$ such that $P_{\Omega_j}(s) = P_{\Omega_j}(t)$ and $\delta(t) \in Q_B^{T_k}$. This implies that $(\delta(s), \delta(t)) \in T_k$. Since $P_{\Omega_j}(s) = P_{\Omega_j}(t)$, by the construction of $\tilde{V}$, $(\delta(s), \delta_R^j(s), \delta(t), \delta_R^j(t)) \in \delta_{\tilde{V}}(s) \subseteq \mathcal{E}_{\Omega_i}^{\tilde{V}}(s)$. Therefore, (c-ii) is also violated and $DF(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) = 0$.

Case 3: $\mathcal{I}_{T_k} = \{1, 2\}$.

For string s in (13), since $\mathcal{E}_{\Omega_i}^G(s) \cap Q_B^{T_k} \neq \emptyset$, by the same argument as in Case 1, (c-i) does not hold. Since $\mathcal{E}_{\Omega_j}^G(s) \cap Q_B^{T_k} \neq \emptyset$, by the same argument as in Case 2, (c-ii) also does not hold. Therefore, $DF(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) = 0$.

Overall, for each case, there exists a string $s \in \mathcal{L}(G)$ such that $DF(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) = 0$, which completes the contrapositive proof.

($\Rightarrow$) Still by contrapositive. Suppose that there exists a string $s \in \mathcal{L}(G)$ such that $DF(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) = 0$. Then, there exists $T_k \in \mathbb{T}$ such that none of (c-i) and (c-ii) holds for $\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)$. Next, we still consider the following three cases for $\mathcal{I}_{T_k}$.

Case 1: $\mathcal{I}_{T_k} = \{i\}$.

Since (c-i) does not hold, we have $(\mathcal{E}_{\Omega_i}^G(s) \times \mathcal{E}_{\Omega_i}^G(s)) \cap T_k = (I_1(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s)) \times I_1(\mathcal{E}_{\Omega_i}^{\tilde{V}}(s))) \cap T_k \neq \emptyset$. This implies that $\exists w \in \mathcal{L}(G)$ such that $\delta(w) \in Q_A^{T_k}$, $P_{\Omega_i}(s) = P_{\Omega_i}(w)$ and $\mathcal{E}_{\Omega_i}^G(w) \cap Q_B^{T_k} = \mathcal{E}_{\Omega_i}^G(s) \cap Q_B^{T_k} \neq \emptyset$. Therefore, we have $(\exists w \in \mathcal{L}(G) : \delta(w) \in Q_A^{T_k})[\mathcal{E}_{\Omega_i}^G(w) \cap Q_B^{T_k} \neq \emptyset]$, i.e., G is not decentralized distinguishable.

Case 2: $\mathcal{I}_{T_k} = \{j\}$.

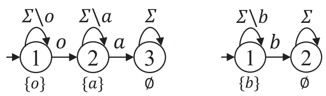
Since (c-ii) does not hold, we have

$$\exists q_1 \in \mathcal{E}_{\Omega_i}^G(s) \cap Q_A^{T_k}, \exists (q_1, q_1^R, q_2, q_2^R) \in \mathcal{E}_{\Omega_i}^{\tilde{V}}(s) : (q_1, q_2) \in T_k.$$

Since $(q_1, q_1^R, q_2, q_2^R) \in \mathcal{E}_{\Omega_i}^{\tilde{V}}(s)$, there exists a string $t \in \mathcal{L}(\tilde{V}) = \mathcal{L}(G)$, such that $P_{\Omega_i}(s) = P_{\Omega_i}(t)$ and $(q_1, q_1^R, q_2, q_2^R) \in \delta_{\tilde{V}}(t)$, which further implies that $q_1 = \delta(t)$ and there exists $w \in \mathcal{L}(G)$ such that $q_2 = \delta(w)$ and $P_{\Omega_j}(t) = P_{\Omega_j}(w)$. Therefore, $\{q_1, q_2\} \subseteq \mathcal{E}_{\Omega_j}^G(t) = \mathcal{E}_{\Omega_j}^G(w)$. Since $(q_1, q_2) \in T_k$, we know that $q_1 \in Q_A^{T_k}$ and $q_2 \in Q_B^{T_k}$. Overall, for $T_k \in \mathbb{T}$, we have $(\exists t \in \mathcal{L}(G) : \delta(t) \in Q_A^{T_k})[\mathcal{E}_{\Omega_j}^G(t) \cap Q_B^{T_k} \neq \emptyset]$, i.e., G is not decentralized distinguishable.

Case 3: $\mathcal{I}_{T_k} = \{1, 2\}$.

Since (c-ii) does not hold, by the same argument as in Case 2, there exists $\exists t \in \mathcal{L}(G)$ such that $P_{\Omega_i}(s) = P_{\Omega_i}(t)$, $\delta(t) \in Q_A^{T_k}$



(a) (b)

Fig. 4. Decentralized minimal solutions. (a) Ω_1^*. (b) Ω_2^*

and $\mathcal{E}^G_{\Omega_j}(t) \cap Q_B^{T_k} \neq \emptyset$. Since (c-i) does not hold, by the same argument as in Case 1, we know that $\mathcal{E}^G_{\Omega_i}(s) \cap Q_B^{T_k} \neq \emptyset$. Since $P_{\Omega_i}(s) = P_{\Omega_i}(t)$, we have $\mathcal{E}^G_{\Omega_i}(t) \cap Q_B^{T_k} \neq \emptyset$. Therefore

$$(\exists t \in \mathcal{L}(G) : \delta(t) \in Q_A^{T_k})(\forall p \in \mathcal{I}_{T_k})[\mathcal{E}^G_{\Omega_p}(t) \cap Q_B^{T_k} \neq \emptyset]$$

i.e., G is not decentralized distinguishable.

Overall, G is not decentralized distinguishable for each case. This completes the contrapositive proof. ■

In the above development, the essence of using $\tilde{V}$ is that we can encode Agent j's information, i.e., Ω_j, into the system model in order to reduce the constrained minimization problem for Agent i to a fully centralized minimization problem. That is, $\tilde{V}$ is a nondeterministic refinement of G that carries both the original state information in G and some useful information of Ω_j. Once $\tilde{V}$ is constructed, we will not use Ω_j anymore, since all useful information, i.e., which pairs of states Agent j cannot distinguish, has been encoded in $\tilde{V}$. Finally, using Theorem 1, we have the following result.

Corollary 1: Problem 2 is decidable.

Proof: By Theorem 1, it is clear that Problem 2 is a special case of Problem 3 by considering system $\tilde{V}$ and setting φ to be $DF : 2^{Q_{\tilde{V}}} \to \{0, 1\}$. Since Problem 3 can be effectively solved, Problem 2 can also be effectively solved. ■

Example 4: We return to the system G in Fig. 2(a) with $\Sigma_{o,1} = \{o, a\}$ and $\Sigma_{o,2} = \{o, b\}$. We still consider specifications $\mathbb{T} = \{T_1, T_2\}$ defined in Example 2. We assume that the sensor activation policy Ω_2 shown in Fig. 2(d) is fixed for Agent 2 and the corresponding automaton $\tilde{V}$ has been shown in Fig. 3. Now, we want to synthesize sensor activation policy Ω_1 such that G is decentralized distinguishable. By defining function DF for $\tilde{V}$ and applying the synthesis algorithm in [18], we obtain a minimal sensor activation policy Ω_1^* shown in Fig. 4(a). Since the main purpose of this paper is to show how to solve the decentralized minimization problem, the reader is referred to [18] for more details about the solution approach to Problem 3. Here, instead of showing how to find Ω_1^*, let us *verify* that Ω_1^* satisfies function DF.

For example, for specification T_1, we consider string of_1a such that $\delta(of_1a) = 6 \in Q_A^{T_k}$. Then we have $x = \mathcal{E}^{\tilde{V}}_{\Omega_1^*}(of_1a) = \{(6, 1, 6, 1)\}$, i.e., $I_1(x) = \{6\}$. Therefore, condition (c-i) holds for x, and we have $DF(x) = 1$. For specification T_2, let us consider string f_2b such that $\delta(f_2b) = 5 \in Q_A^{T_2}$. Then, we have $x = \mathcal{E}^{\tilde{V}}_{\Omega_1^*}(f_2b) = \{(1, 1, 1, 1), (3, 1, 1, 1,), (1, 1, 3, 1), (3, 1, 3, 1), (5, 1, 5, 1), (7, 1, 7, 1)\}$, i.e., $I_1(x) = \{1, 3, 5, 7\}$. For this case, condition (c-i) does not hold for x since $1 \in I_1(x) \cap Q_B^{T_2}$. However, for $5 \in I_1(x) \cap Q_A^{T_2} = \{5, 7\}$, $(5, 1, 5, 1)$ is the only state in x whose first component is 5 and $(5, 5) \notin T_2$. Similarly, for $7 \in I_1(x) \cap Q_A^{T_2}$, $(7, 1, 7, 1)$ is the only state in x whose first component is 7 and $(7, 7) \notin T_2$. Therefore, condition (c-ii) holds, and we still have $DF(x) = 1$.

Algorithm 1: D-MIN-ACT.

input : $G, \mathbb{T}, \Sigma_{o,1}$ and $\Sigma_{o,2}$
output: $\bar{\Omega}^*$

$\Omega_1^* \leftarrow \Omega_{\Sigma_{o,1}}$ and $\Omega_2^* \leftarrow \Omega_{\Sigma_{o,2}}$
for $i \in \{1, 2\}$ **do**
  $j \in \{1, 2\} \setminus \{i\}$
  Fix Ω_j^*. Construct automaton $\tilde{V}$ w.r.t. Ω_j^* and define function DF.
  Obtain minimal Ω_i' by solving Problem 3 w.r.t. system $\tilde{V}$ and function DF.
  $\Omega_i^* \leftarrow \Omega_i'$.
$\bar{\Omega}^* \leftarrow [\Omega_1^*, \Omega_2^*]$

V. Synthesis Algorithm

In this section, we first present an algorithm that solves the decentralized sensor activation problem by using the results we developed so far. Then, we prove the correctness of the algorithm.

Our synthesis algorithm is formally presented in Algorithm 1. Essentially, Algorithm 1 solves two centralized constrained minimization problems. First, we set Agent 2's sensor activation policy to be $\Omega_{\Sigma_{o,2}}$, i.e., the most conservative one, and solve the constrained minimization problem for Agent 1. Then, we fix the obtained sensor activation policy for Agent 1 and solve the constrained minimization problem for Agent 2. However, the following question arises: *"After the above procedure, do we need to fix Agent 2's new sensor activation policy and go back to minimize Agent 1's sensor activation policy again?"* In other words, we need to answer whether or not iterations between two centralized constrained minimization problems are required in order to obtain a decentralized minimal solution. Hereafter, we show that such iterations are not necessary for our problem, and Algorithm 1 indeed yields a decentralized minimal solution in the above two steps. This is because of the following monotonicity property, which generalizes the results in [13] and [14].

Lemma 1: (Monotonicity property). Let G be the system, $\mathbb{T}$ be a set of specifications, and $\bar{\Omega} = [\Omega_1, \Omega_2]$ and $\bar{\Omega}' = [\Omega_1', \Omega_2']$ be two sensor activation policies such that $\bar{\Omega}' \subseteq \bar{\Omega}$. Then, G is decentralized distinguishable w.r.t. $\bar{\Omega}'$ and $\mathbb{T}$ implies that G is decentralized distinguishable w.r.t. $\bar{\Omega}$ and $\mathbb{T}$.

Proof: Since G is decentralized distinguishable w.r.t. $\bar{\Omega}'$ and $\mathbb{T}$, then $\forall T_k \in \mathbb{T}, \forall s \in \mathcal{L}(G) : \delta(s) \in Q_A^{T_k}$, we have $\exists i \in \{1, 2\} : \mathcal{E}^G_{\Omega_i'}(s) \cap Q_B^{T_k} = \emptyset$. Since $\bar{\Omega}' \subseteq \bar{\Omega}$, we know that $\forall i \in \{1, 2\} : \Omega_i' \subseteq \Omega_i$, which implies that $\mathcal{E}^G_{\Omega_i}(s) \subseteq \mathcal{E}^G_{\Omega_i'}(s)$ for any $s \in \mathcal{L}(G)$. Therefore, $\forall T_k \in \mathbb{T}, \forall s \in \mathcal{L}(G) : \delta(s) \in Q_A^{T_k}$, we have $\exists i \in \{1, 2\} : \mathcal{E}^G_{\Omega_i}(s) \cap Q_B^{T_k} = \emptyset$, i.e., G is also decentralized distinguishable w.r.t. $\bar{\Omega}$. ■

We are now ready to prove the correctness of Algorithm 1.

Theorem 2: Let $\bar{\Omega}^*$ be the output of Algorithm 1. Then, $\bar{\Omega}^*$ solves Problem 1.

Proof: It is clear that G is decentralized distinguishable w.r.t. $\bar{\Omega}^*$ and $\mathbb{T}$, since decentralized distinguishability is guaranteed in each centralized constrained minimization problem. It remains to show that $\bar{\Omega}^*$ is minimal; we proceed by contradiction. Let us assume that there exists another sensor activation policy $\bar{\Omega}' = [\Omega_1', \Omega_2']$ such that G is decentralized distinguishable w.r.t. $\bar{\Omega}'$ and $\mathbb{T}$ and $\bar{\Omega}' \subset \bar{\Omega}^*$. The second condition means that $\exists i, j \in \{1, 2\}, i \neq j$, such that $\Omega_i' \subset \Omega_i^*$ and $\Omega_j' \subseteq \Omega_j^*$. Suppose that $i = 1$ and $j = 2$. Then, we know that Ω_1^* is obtained by fixing Agent 2's sensor activation policy to be $\Omega_{\Sigma_{o,2}}$, where $\Omega_2' \subseteq \Omega_2^* \subseteq \Omega_{\Sigma_{o,2}}$. By Lemma 1, we know that "G is decentralized distinguishable w.r.t. $[\Omega_1', \Omega_2']$" implies that "G is decentralized distinguishable w.r.t. $[\Omega_1', \Omega_{\Sigma_{o,2}}]$." However, since $\Omega_1' \subset \Omega_1^*$, this contradicts to the fact that Ω_1^* is a solution to Problem 2. Similarly, suppose that $i = 2$ and $j = 1$. Then, we know that Ω_2^* is obtained by fixing Agent 1's sensor activation policy to be Ω_1^*, where $\Omega_1' \subseteq \Omega_1^*$. By Lemma 1, we know that "G is decentralized distinguishable w.r.t. $[\Omega_1', \Omega_2']$" implies that "G is decentralized distinguishable w.r.t. $[\Omega_1^*, \Omega_2']$." However, since $\Omega_2' \subset \Omega_2^*$, it again contradicts the fact that Ω_2^* is a solution to Problem 2. ■

Remark 3: Recall that in the synthesized minimal decentralized policy $\Omega^* = [\Omega_1^*, \Omega_2^*]$, each Ω_i^* is a pair. Therefore, to implement Ω^*, for each agent i, one can first store the offline computed Ω_i^* at local site i. To run Ω_i^* online, we just need to remember the current state in Ω_i^* and the current sensing decision is the output of this state. Whenever a new event is observed, we just update the current state based on the transition function of Ω_i^*, move to a new state of Ω_i^* and update the sensing decision to be the output of this new state, and so forth. This is also the same way for implementing a supervisor (see, e.g., [30]).

Remark 4: In general, the minimal solution to Problem 1 is not unique due to the following reasons. First, for each centralized constraint minimization problem involved in Algorithm 1, the minimal solution is not unique in general [18]. There may exist two incomparable centralized minimal solutions to Problem 2 or 3. Second, the decentralized minimal solution obtained by Algorithm 1 also depends on the order of the centralized constraint minimization problems. To implement Algorithm 1, we can randomly select an order for each agent. In general, fixing Agent 1 first and fixing Agent 2 first may result in different minimal solutions. However, in any case, solution $\bar{\Omega}^*$ returned by Algorithm 1 is *guaranteed* to be minimal in the sense that other minimal solutions must be *incomparable* with $\bar{\Omega}^*$.

We illustrate Algorithm 1 by an example.

Example 5: Again, consider the system G in Fig. 2(a) and specifications $\mathbb{T} = \{T_1, T_2\}$ defined in Example 2. Let $\Sigma_{o,1} = \{o, a\}$ and $\Sigma_{o,2} = \{o, b\}$, respectively, be the set of observable events for Agent 1 and Agent 2. Initially, we set $\Omega_2 = \Omega_{\Sigma_{o,2}}$ and solve the constrained minimization problem for Agent 1; this has been solved in Example 4, and we obtained Ω_1^* shown in Fig. 4(a). Next, we fix Ω_1^* for Agent 1 and solve the constrained minimization problem for Agent 2. Then, we obtain the sensor activation policy Ω_2^* as shown in Fig. 4(b). We see that Ω_2^* turns all sensors OFF after b is observed, since once b occurs, Agent 2 will know for sure that the system is in state 5 or 7, and there is no need to monitor any event. Therefore, $[\Omega_1^*, \Omega_2^*]$ is a minimal pair of sensor activation policies that ensure decentralized distinguishability.

Remark 5: We conclude this section by discussing the complexity of synthesis algorithm. Suppose that we first fix Agent 2. Initially, $\Omega_2 = \Omega_{\Sigma_{o,2}}$ and its automaton only contains a single state. To solve the constraint optimization problem when Ω_2 is fixed, first, we need to construct $\tilde{V}$, which is polynomial in the size of G and Ω_2. However, since an observer-like constructed is exploited, the algorithm in [18] requires exponential complexity w.r.t. the size of the system, i.e., $\tilde{V}$, and the size of the solution Ω_1^* is also exponential in the size of $\tilde{V}$. Again, constructing $\tilde{V}$ when Agent 1 is fixed only requires polynomial complexity w.r.t. Ω_1^*, but synthesizing Ω_2^* requires exponential complexity again. Therefore, the overall complexity is doubly exponential w.r.t. the size of G. Such a doubly exponential complexity arises in many synthesis problems, where two *incomparable* observations are involved (see, e.g., [24] and [38]).

Remark 6: In this paper, we adopt a logical optimality criterion that has been widely used in the literature. One possible future direction is to consider a numerical cost function that introduces a quantitative objective. This numerical setting is much more challenging to deal with, in particular in the decentralized setting. Specifically, using the person-by-person approach for this setting may have the following problems. First, how to solve the constrained optimization problem for a quantitative objective may be very different from the approach developed in this paper. Second, iterations are needed in general and the convergence may not be guaranteed as the domain of languages is infinite. Moreover, even if the person-by-person iteration converges, it may only converge to a local optimal solution. These questions are very interesting future directions but are already beyond the scope of this paper.

VI. Application of the Decentralized State Disambiguation Problem

In this section, we show that the notions of K-codiagnosability, coobservability, and coprognosability are instances of decentralized distinguishability. Therefore, the proposed framework is applicable for solving the dynamic sensor activation problems for the purposes of decentralized fault diagnosis, decentralized control, and decentralized fault prognosis.

A. Decentralized Fault Diagnosis

In the decentralized fault diagnosis problem, the local agents need to work as a team such that any fault be diagnosed within a bounded number of steps. Formally, we denote by $\Sigma_F \subseteq \Sigma_{uo}$ the set of fault events. We assume that Σ_F is partitioned into m *fault types*: $\Sigma_F = \Sigma_{F_1} \dot{\cup} \ldots \dot{\cup} \Sigma_{F_m}$; we denote by Π the partition and by $\mathcal{F} = \{1, \ldots, m\}$ the index set of the fault types. For any $k \in \mathcal{F}$, we define $\Psi(E_{F_k}) = \{sf \in \mathcal{L}(G) : f \in E_{F_k}\}$ to be the set of strings that end with a fault event of type k. We write $E_{F_k} \in s$, if $\overline{\{s\}} \cap \Psi(E_{F_k}) \neq \emptyset$. The notion of K-codiagnosability was

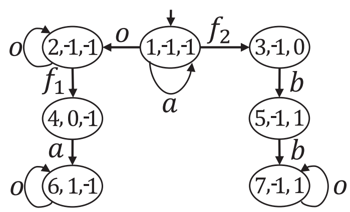


Fig. 5. Augmented system $\tilde{G}$.

proposed in the literature to capture whether or not any fault can be diagnosed within K steps [3], [4].

Definition 2: (K-codiagnosability). Let $K \in \mathbb{N}$. We say that live language $\mathcal{L}(G)$ is K-codiagnosable w.r.t. $\bar{\Omega}$, Σ_F, and Π if

$$(\forall k \in \mathcal{F})(\forall s \in \Psi(\Sigma_{F_k}))(\forall t \in \mathcal{L}(G)/s : |t| \geq K)$$
$$(\exists i \in \{1,2\})(\forall w \in \mathcal{L}(G))[P_{\Omega_i}(w) = P_{\Omega_i}(st) \Rightarrow \Sigma_{F_k} \in w]. \quad (14)$$

To show that K-codiagnosability can be formulated as decentralized distinguishability, following similar constructions in [17] and [19], we first refine the state space of G by defining a new automaton $\tilde{G} = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{q}_0)$, where $\tilde{Q} \subseteq Q \times \{-1, 0, 1 \ldots, K\}^m$, $\tilde{q}_0 = (q_0, -1, \ldots, -1)$ and the partial transition function $\tilde{\delta} : \tilde{Q} \times \Sigma \rightarrow \tilde{Q}$ is defined by: for any $(q, n_1, \ldots, n_m) \in \tilde{Q}$ and $\sigma \in \Sigma$, we have

$$\tilde{\delta}((q, n_1, \ldots, n_m), \sigma) = (\delta(q, \sigma), n_1 + \Delta_1, \ldots, n_m + \Delta_m)$$

where for each $i \in \{1, \ldots, m\}$, Δ_i is defined by

$$\Delta_i = \begin{cases} 0, & \text{if } [n_i = K] \text{ or } [n_i = -1 \wedge \sigma \notin \Sigma_{F_i}] \\ 1, & \text{if } [0 \leq n_i < K] \text{ or } [n_i = -1 \wedge \sigma \in \Sigma_{F_i}]. \end{cases}$$

Intuitively, $\tilde{G}$ simply unfolds G by "counting" the number of steps since each type of fault has occurred. Since $\mathcal{L}(\tilde{G}) = \mathcal{L}(G)$, we can synthesize a sensor activation policy for G based on $\tilde{G}$. For any state $\tilde{q} = (q, n_1, \ldots, n_m) \in \tilde{G}$, we denote by $[\tilde{q}]_i$ its $(i+1)$th component, i.e., n_i.

Based on $\tilde{G}$, we define a set of specifications $\mathbb{T}_{\text{diag}} = \{T_1, T_2, \ldots, T_m\}$ as follows: for each $T_k \in \mathbb{T}$, we have

$$Q_A^{T_k} = \{q \in \tilde{Q} : [q]_k = K\} \text{ and } Q_B^{T_k} = \{q \in \tilde{Q} : [q]_k = -1\}.$$

The following result reveals that, to enforce K-codiagnosability, it suffices to enforce decentralized distinguishability for $\mathbb{T}_{\text{diag}}$.

Theorem 3: A live language $\mathcal{L}(G)$ is K-codiagnosable w.r.t. $\bar{\Omega}$, Σ_F and Π if and only if $\tilde{G}$ is decentralized distinguishable w.r.t. $\bar{\Omega}$ and $\mathbb{T}_{\text{diag}}$.

Example 6: Let us consider again system G shown in Fig. 2(a). Suppose that $\Sigma_F = \Sigma_{F_1} \dot{\cup} \Sigma_{F_2} = \{f_1\} \dot{\cup} \{f_2\}$. Let us consider $K = 1$. Then, the refined automaton $\tilde{G}$ is shown in Fig. 5 . For example, state $q = (6, 1, -1)$ means that 1)the system is at state 6 in G, 2)f_1 has occurred for more than one step (since $[q]_1 = K$), and 3) f_2 has not occurred (since $[q]_2 = -1$). Then, $\mathbb{T}_{\text{diag}} = \{T_1, T_2\}$ is defined by $T_1 = \{(6, 1, -1)\} \times \{(1, -1, -1), (2, -1, -1), (3, -1, 0), (5, -1, 1), (7, -1, 1)\}$ and $T_2 = \{(5, -1, 1), (7, -1, 1)\} \times \{(1, -1, -1), (2, -1, -1), (4, 0, -1), (6, 1, -1)\}$. Since $\tilde{G}$ and G are isomorphic for this specific example, we see that $\mathbb{T}_{\text{diag}}$ is indeed the same specification $\mathbb{T}$ defined in Example 2. Therefore, the solution we obtained in Example 5 has solved the sensor activation problem for 1-codiagnosability.

B. Decentralized Supervisory Control

Another important decentralized decision-making problem is the decentralized supervisory control problem [1], [2]. In this problem, each local agent $i \in \mathcal{I}$ can disable events in $\Sigma_{c,i} \subseteq \Sigma$ dynamically based on its local observation Ω_i. We define $\Sigma_c = \cup_{i \in \mathcal{I}} \Sigma_{c,i}$ as the set of all controllable events, and for each $\sigma \in \Sigma_c$, we define $\mathcal{I}^c(\sigma) = \{i \in \mathcal{I} : \sigma \in \Sigma_{c,i}\}$ as the set of agents that can disable σ. The control objective is to make sure that the closed-loop system achieves a desired language $\mathcal{L}(H) \subseteq \mathcal{L}(G)$. The key property regarding the decentralized information in this problem is the notion of *coobservability*; it together with the notion of controllability provides the necessary and sufficient conditions for exactly achieving a given specification language. We recall its definition from [1].

Definition 3: (Coobservability). We say that $\mathcal{L}(G)$ is coobservable w.r.t. $\mathcal{L}(H)$, $\Sigma_{c,1}, \Sigma_{c,2}$, and $\bar{\Omega}$ if

$$(\forall s \in \mathcal{L}(H))(\forall \sigma \in \Sigma_c : s\sigma \in \mathcal{L}(G) \setminus \mathcal{L}(H))$$
$$(\exists i \in \mathcal{I}^c(\sigma_k))[P_{\Omega_i}^{-1}(P_{\Omega_i}(s))\{\sigma\} \cap \mathcal{L}(H) = \emptyset]. \quad (15)$$

Hereafter, we assume that $H = (Q_H, \Sigma, \delta_H, q_{0,H})$ is a strict subautomaton of G, i.e.:

1) $Q_H \subseteq Q$;
2) $\forall s \in \mathcal{L}(H) : \delta_H(s) = \delta(s)$;
3) $\forall s \in \mathcal{L}(G) \setminus \mathcal{L}(H) : \delta(s) \in X \setminus X_H$.

This assumption is without loss of generality, since we can always refine H and G such that it holds [39]. The refinement essentially takes the product of H of G, and the resulting system contains at most $|Q_H| \times |Q|$ states. Now, suppose that $\Sigma_c = \{\sigma_1, \ldots, \sigma_m\}$ is the set of controllable events. We define a set of specifications $\mathbb{T}_{\text{cont}} = \{T_1, T_2, \ldots, T_m\}$ as follows: for each $T_k \in \mathbb{T}$, we have

$$Q_A^{T_k} = \{q \in Q_H : \delta(q, \sigma_k)! \wedge \delta_H(q, \sigma_k)\neg!\}$$
$$Q_B^{T_k} = \{q \in Q_H : \delta_H(q, \sigma_k)!\}$$

with $\mathcal{I}_{T_k} = \mathcal{I}^c(\sigma_k)$, where "$\neg!$" means "is not defined."

Intuitively, for each controllable event $\sigma_k \in \Sigma_c$, $Q_A^{T_k}$ is the set of states at which σ_k must be disabled for safety purposes, while $Q_A^{T_k}$ is the set of states at which σ_k must be enabled to achieve $\mathcal{L}(H)$. The following result reveals that coobservability is also a special case of decentralized distinguishability with $\mathbb{T}_{\text{cont}}$.

Theorem 4: Let G be the system and H be the specification automaton. Then, $\mathcal{L}(G)$ is coobservable w.r.t. $\mathcal{L}(H)$, $\Sigma_{c,1}, \Sigma_{c,2}$, and $\bar{\Omega}$ if and only if G is decentralized distinguishable w.r.t. $\bar{\Omega}$ and $\mathbb{T}_{\text{cont}}$.

C. Decentralized Fault Prediction

In some safety-critical systems, we may not only want to diagnose any fault after its occurrence, but also want to *predict* any fault before it occurs [40]. In [6], the notion of coprognosability was proposed to capture whether or not any fault occurrence can be predicted in a decentralized system. The definition is reviewed as follows.

Definition 4: (Coprognosability). We say that language $\mathcal{L}(G)$ is coprognosable w.r.t. $\bar{\Omega}$ and Σ_F if

$$(\forall s \in \Psi(\Sigma_F))(\exists t \in \overline{\{s\}} : \Sigma_F \notin t)$$
$$(\exists i \in \{1,2\})(\forall u \in P_{\Omega_i}^{-1}(P_{\Omega_i}(t)) : \Sigma_F \notin u)$$
$$(\exists K \in \mathbb{N})(\forall v \in \mathcal{L}(G)/u)[|v| \geq K \Rightarrow \Sigma_F \in uv]. \quad (16)$$

To proceed further, we assume that the state space of G is partitioned as $Q = Q_N \dot{\cup} Q_F$ such that we have the following:

1) $\forall s \in \mathcal{L}(G) : \delta(s) \in Q_N \Leftrightarrow \Sigma_F \notin s$;
2) $\forall s \in \mathcal{L}(G) : \delta(s) \in Q_F \Leftrightarrow \Sigma_F \in s$.

Note that, this assumption is without loss of generality, since we can simply refine the state space of G such that this assumption holds.

In order to formulate coprognosability as an instance of decentralized distinguishability, we need the notions of nonindicator states and boundary states, which are initially introduced in [6]. We say that a state $q \in Q$ is:

1) a nonindicator state, if $q \in Q_N$ and $\forall K \in \mathbb{N}, \exists s \in \mathcal{L}(G, q) : |s| \geq K \wedge \Sigma_F \notin s$;
2) a boundary state, if $\exists f \in \Sigma_F : \delta(q, f)!$.

We denote by $\mathcal{N}_Q$ and ∂_Q the set of nonindicator states and the set of boundary states, respectively.

With these notions, we define a simple specification $\mathbb{T}_{\text{pre}} := \{T_1\}$, where $Q_A^{T_1} = \partial_Q$ and $Q_B^{T_1} = \mathcal{N}_Q$ with $\mathcal{I}_{T_1} = \mathcal{I}$. The following result reveals that, to enforce coprognosability, it suffices to enforce decentralized distinguishability with $\mathbb{T}_{\text{pre}}$.

Theorem 5: $\mathcal{L}(G)$ is coprognosable w.r.t. $\bar{\Omega}$ and Σ_F if and only if G is decentralized distinguishable w.r.t. $\bar{\Omega}$ and $\mathbb{T}_{\text{pre}}$.

Remark 7: Note that ∂_Q and $\mathcal{N}_Q$ need not be disjoint. By the above theorem, the system will not be coprognosable under any sensor activation policies if $\partial_Q \cap \mathcal{N}_Q \neq \emptyset$.

D. Extension to the Conjunctive Architecture

So far, we have shown that K-codiagnosability, coobservability, and coprognosability are special cases of decentralized distinguishability. As we mentioned earlier, all results in this paper are developed based on the disjunctive architecture, i.e., the coordinator issues "1" globally if and only if *one* local agent issues "1." Alternatively, one may also use the *conjunctive* rule to obtain a global decision, i.e., the coordinator issues "0" globally if and only if *one* local agent issues "0." In this case, suppose that a string leading to a state in $Q_B^{T_k}$ is executed and a global decision "0" has to be made. Then, a local agent must know that the system is not in $Q_A^{T_k}$ *unambiguously* when it issues "0"; otherwise, a wrong global decision may be made at some state in $Q_A^{T_k}$. Therefore, we need to require that

$$(\forall s \in \mathcal{L}(G) : \delta(s) \in Q_B^{T_k})(\exists i \in \mathcal{I}_{T_k})[\mathcal{E}_{\Omega_i}^{G}(s) \cap Q_A^{T_k} = \emptyset].$$

By comparing the above requirement with decentralized distinguishability, which is defined in terms of the disjunctive architecture, we see that this requirement is indeed the same as decentralized distinguishability by swapping $Q_A^{T_k}$ and $Q_B^{T_k}$. Therefore, there is no need to define a conjunctive version of decentralized distinguishability; it is just a matter of how the specification T_k is defined.

For example, in [2], the notion of D&A-coobservability was proposed as a complement of coobservability.[1] We recall its definition.

Definition 5: (D&A-coobservability). We say that $\mathcal{L}(G)$ is D&A-coobservable w.r.t. $\mathcal{L}(H)$, $\Sigma_{c,1}$, $\Sigma_{c,2}$, and $\bar{\Omega}$ if

$$(\forall s \in \mathcal{L}(H))(\forall \sigma \in \Sigma_c : s\sigma \in \mathcal{L}(H))$$
$$(\exists i \in \mathcal{I}^c(\sigma_k))[P_{\Omega_i}^{-1}(P_{\Omega_i}(s))\{\sigma\} \cap \mathcal{L}(G) \subseteq \mathcal{L}(H)]. \quad (17)$$

Intuitively, D&A-coobservability requires that for any string for which σ has to be enabled, there exists at least one agent that knows for sure that σ should not be disabled. We can also formulate D&A-coobservability as an instance of decentralized distinguishability by defining $\mathbb{T}_{\text{cont}}^{CJ} = \{T_1^{CJ}, T_2^{CJ}, \ldots, T_m^{CJ}\}$, where for each $T_k^{CJ} \in \mathbb{T}$, we have

$$Q_A^{T_k^{CJ}} = \{q \in Q_H : \delta_H(q, \sigma_k)!\}$$
$$Q_B^{T_k^{CJ}} = \{q \in Q_H : \delta(q, \sigma_k)! \wedge \delta_H(q, \sigma_k)\neg!\}$$

with $\mathcal{I}_{T_k} = \mathcal{I}^c(\sigma_k)$. The proof of the correctness of $\mathbb{T}_{\text{cont}}^{CJ}$ is omitted, since it is similar to the proof of Theorem 4.

Similarly, one can also show that conjunctive K-codiagnosability [5], [41] and conjunctive coprognosability [42], [43] are instances of decentralized distinguishabiity; we just need to define new specifications $\mathbb{T}_{\text{diag}}^{CJ}$ and $\mathbb{T}_{\text{pre}}^{CJ}$ by swapping each $Q_A^{T_k}$ and $Q_B^{T_k}$ in $\mathbb{T}_{\text{diag}}$ and $\mathbb{T}_{\text{pre}}$, respectively.

VII. Conclusion

We presented a novel approach for solving the problem of decentralized sensor activation for a class of properties. We proposed the notion of decentralized distinguishability, which covers coobservability, K-codiagnosability, and coprognosability. To enforce decentralized distinguishability, we first adopted a person-by-person approach to decompose the decentralized minimization problem to two consecutive centralized constrained minimization problems. Then, a novel approach was proposed to reduce each centralized constrained minimization problem to a fully centralized sensor activation that is solved effectively in the literature. Finally, we showed that the decentralized solution obtained by our methodology is language-based minimal.

Appendix

A. Proofs Not Contained in the Main Body

Proof of Theorem 3

Proof: ($\Rightarrow$) By contraposition. Suppose that $\tilde{G}$ is not decentralized distinguishable. Then, we know that there exist

[1] Here, "D&A" stands for "disjunctive & antipermissive." Also, coobservability in Definition 3 is referred to as C&P-coobservability, where "C&A" standards for "conjunctive & permissive." The reason why C&P-coobservability corresponds to decentralized distinguishability in the disjunctive architecture is that Rudie and Wonham [1] consider the conjunction of enablements, while $\mathbb{T}_{\text{cont}}$ captures the disjunction of disablements; they are essentially equivalent.

$k \in \{1,\ldots,m\}$ and a string $s \in \mathcal{L}(G)$ such that $q := \tilde{\delta}(s) \in Q_A^{T_k}$, and for each $i \in \{1,2\}$, there exists $q_i \in \mathcal{E}_{\Omega_i}^{\tilde{G}}(s)$ such that $q_i \in Q_B^{T_k}$. Then, we know that, for each $i \in \{1,2\}$, there exists a string $s_i \in \mathcal{L}(G)$ such that $\tilde{\delta}(s_i) = q_i$ and $P_{\Omega_i}(s) = P_{\Omega_i}(s_i)$. By the definition of T_k, $q \in Q_A^{T_k}$ implies that $[q]_k = K$. According to the construction of $\tilde{G}$, $\tilde{\delta}(s) = q$ implies that we can write $s = uv$ such that $u \in \Psi(\Sigma_{F_k})$ and $|v| \geq K$. For each $i \in \{1,2\}$, since $q_i \in Q_B^{T_k}$, we know that $[q_i]_k = -1$, which implies that $\Sigma_{F_k} \notin s_i$. Overall, we know that

$$(\exists k \in \mathcal{F})(\exists u \in \Psi(\Sigma_{F_k}))(\exists v \in \mathcal{L}(G)/u : |v| \geq K)$$
$$(\forall i \in \{1,2\})(\exists s_i \in \mathcal{L}(G))[P_{\Omega_i}(uv) = P_{\Omega_i}(s_i) \wedge \Sigma_{F_i} \notin s_i] \quad (18)$$

i.e., $\mathcal{L}(G)$ is not K-codiagnosable.

($\Leftarrow$) Still by contraposition. Suppose that $\tilde{G}$ is not K-codiagnosable, i.e., (18) holds. Let $q := \tilde{\delta}(uv)$, $q_1 := \tilde{\delta}(s_1)$ and $q_2 := \tilde{\delta}(s_2)$. Then, according to the definition of $\tilde{G}$, we know that $[q]_k = K$, $[q_1]_k = [q_2]_k = -1$, which implies that $q \in Q_A^{T_k}$ and $q_1, q_2 \in Q_B^{T_k}$. Moreover, since for each $i = 1,2$, $P_{\Omega_i}(uv) = P_{\Omega_i}(s_i)$, we know that $q_i \in \mathcal{E}_{\Omega_i}^{\tilde{G}}(s_i) = \mathcal{E}_{\Omega_i}^{\tilde{G}}(uv)$, i.e., $\mathcal{E}_{\Omega_i}^{\tilde{G}}(uv) \cap Q_B^{T_k} \neq \emptyset$. Overall, we know that $(\exists T_k \in \mathbb{T})(\exists uv \in \mathcal{L}(\tilde{G}) : \tilde{\delta}(uv) \in Q_A^{T_k})(\forall i \in \{1,2\})[\mathcal{E}_{\Omega_i}^{\tilde{G}}(uv) \cap Q_B^{T_k} \neq \emptyset]$, i.e., $\tilde{G}$ is not decentralized distinguishable w.r.t. $\mathbb{T}_{\text{diag}}$. ∎

Proof of Theorem 4

Proof: ($\Rightarrow$) By contraposition. Suppose that G is not decentralized distinguishable. Then, we know that there exist $T_k \in \mathbb{T}, s \in \mathcal{L}(G) : \delta(s) \in Q_A^{T_k}$ such that for each $i \in \mathcal{I}_{T_k}$, there exists $t_i \in \mathcal{L}(G)$ such that $P_{\Omega_i}(s) = P_{\Omega_i}(t_i)$ and $\delta(t_i) \in Q_B^{T_k}$. Let $\sigma_k \in \Sigma_c$ be the controllable event associated with T_k. Then, $\delta(s) \in Q_A^{T_k}$ implies that $s \in \mathcal{L}(H)$, $s\sigma_k \in \mathcal{L}(G) \setminus \mathcal{L}(H)$ and $\delta(t_i) \in Q_B^{T_k}$ implies that $t_i\sigma_k \in \mathcal{L}(H)$. Moreover, $\mathcal{I}_{T_k} = \mathcal{I}^c(\sigma_k)$. Overall, we know that $\exists s \in \mathcal{L}(H), \sigma_k \in \Sigma_c$ such that $s\sigma_k \in \mathcal{L}(G) \setminus \mathcal{L}(H)$ and for each $i \in \mathcal{I}^c(\sigma_k)$, $t_i\sigma_k \in P_{\Omega_i}^{-1}(P_{\Omega_i}(s))\{\sigma_k\} \cap \mathcal{L}(H) \neq \emptyset$, i.e., $\mathcal{L}(G)$ is not coobservable.

($\Leftarrow$) By contraposition. Suppose that $\mathcal{L}(G)$ is not coobservable. Then, we know that $\exists s \in \mathcal{L}(H), \sigma_k \in \Sigma_c : s\sigma_k \in \mathcal{L}(G) \setminus \mathcal{L}(H)$ such that for each $i \in \mathcal{I}^c(\sigma_k)$, there exists $t_i \in \mathcal{L}(G)$ such that $t_i\sigma_k \in \mathcal{L}(H)$ and $P_{\Omega_i}(s) = P_{\Omega_i}(t_i)$. For the above s and t_i, we know that $\delta(s) \in Q_A^{T_k}$ and $\delta(t_i) \in Q_B^{T_k}$. Therefore, for s and σ_k, we know that for each $i \in \mathcal{I}_{T_k} = \mathcal{I}^c(\sigma_k)$, $\delta(t_i) \in \mathcal{E}_{\Omega_i}^G(t_i) \cap Q_B^{T_k} = \mathcal{E}_{\Omega_i}^G(s) \cap Q_B^{T_k} \neq \emptyset$, i.e., G is not decentralized distinguishable. ∎

Proof of Theorem 5

Proof: ($\Rightarrow$) By contraposition. Suppose that G is not decentralized distinguishable. Then, we know that there exists $s \in \mathcal{L}(G)$ such that $q := \delta(s) \in \partial_Q$ and for each $i \in \{1,2\}$, there exists $q_i \in \mathcal{E}_{\Omega_i}^G(s)$ such that $q_i \in \mathcal{N}_Q$, i.e., there exists a string $s_i \in \mathcal{L}(G)$ such that $\Sigma_F \notin s_i$, $\delta(s_i) = q_i$ and $P_{\Omega_i}(s) = P_{\Omega_i}(s_i)$. Since $q \in \partial_Q$, we know that $\exists f \in \Sigma_F : sf \in \Psi(\Sigma_F)$. Let $t \in \overline{\{s\}}$ be an arbitrary prefix of s such that $\Sigma_F \notin t$. Then, for each $i \in \{1,2\}$, since $P_{\Omega_i}(s) = P_{\Omega_i}(s_i)$, we know that

$$\forall t \in \overline{\{s\}}, \exists t_i \in \overline{\{s_i\}} : P_{\Omega_i}(t) = P_{\Omega_i}(t_i) \wedge \Sigma_F \notin t_i. \quad (19)$$

Moreover, since $q_i \in \mathcal{N}_Q$ which is reachable from $\delta(t_i)$, we know that, for any $K \in \mathbb{N}$, there exists a string w_i such that $t_iw_i \in \mathcal{L}(G)$, $\Sigma_F \notin t_iw_i$ and $|w_i| \geq K$. Overall, we know that

$$(\exists sf \in \Psi(\Sigma_F))(\forall t \in \overline{\{sf\}} : \Sigma_F \notin t)$$
$$(\forall i \in \{1,2\})(\exists t_i \in P_{\Omega_i}^{-1}(P_{\Omega_i}(t)) : \Sigma_F \notin t_i)$$
$$(\forall K \in \mathbb{N})(\exists w_i \in \mathcal{L}(G)/t_i)[|w_i| \geq K \wedge \Sigma_F \notin t_iw_i] \quad (20)$$

i.e., G is not coprognosable w.r.t. $\bar{\Omega}$ and Σ_F.

($\Leftarrow$) Suppose that G is not coprognosable, i.e., (20) holds. Let sf be a string satisfying (20). Let t be a prefix of s such that $\Sigma_F \notin t$ and $tf' \in \overline{\{s\}}$ for some $f' \in \Sigma_F$. Then, we know that $q := \delta(t) \in \partial_Q$. According to (20), we know that, for each agent $i \in \{1,2\}$, there exists a string $t_i \in \mathcal{L}(G)$ such that 1) $\Sigma_F \notin t_i$; 2) $(\forall K \in \mathbb{N})(\exists w_i \in \mathcal{L}(G)/t_i)[|w_i| \geq K \wedge \Sigma_F \notin t_iw_i]$; and 3) $P_{\Omega_i}(t_i) = P_{\Omega_i}(t)$. The first two conditions imply that $q_i := \delta(t_i) \in \mathcal{N}_Q$. Moreover, the last condition implies that $\{q, q_i\} \subseteq \mathcal{E}_{\Omega_i}^G(t)$. Overall, we know that $(\exists t \in \mathcal{L}(G) : \delta(t) \in \partial_Q)(\forall i \in \{1,2\})[\mathcal{E}_{\Omega_i}^G(t) \cap \mathcal{N}_Q = \emptyset]$, i.e., G is not decentralized distinguishable. ∎

References

[1] K. Rudie and W. Wonham, "Think globally, act locally: Decentralized supervisory control," *IEEE Trans. Automat. Control*, vol. 37, no. 11, pp. 1692–1708, Nov. 1992.

[2] T.-S. Yoo and S. Lafortune, "A general architecture for decentralized supervisory control of discrete-event systems," *Discrete Event Dyn. Syst., Theory Appl.*, vol. 12, no. 3, pp. 335–377, 2002.

[3] R. Debouk, S. Lafortune, and D. Teneketzis, "Coordinated decentralized protocols for failure diagnosis of discrete event systems," *Discrete Event Dyn. Syst., Theory Appl.*, vol. 10, no. 1–2, pp. 33–86, 2000.

[4] W. Qiu and R. Kumar, "Decentralized failure diagnosis of discrete event systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 36, no. 2, pp. 384–395, Mar. 2006.

[5] Y. Wang, T.-S. Yoo, and S. Lafortune, "Diagnosis of discrete event systems using decentralized architectures," *Discrete Event Dyn. Syst.: Theory Appl.*, vol. 17, no. 2, pp. 233–263, 2007.

[6] R. Kumar and S. Takai, "Decentralized prognosis of failures in discrete event systems," *IEEE Trans. Automat. Control*, vol. 55, no. 1, pp. 48–59, Jan. 2010.

[7] A. Haji-Valizadeh and K. Loparo, "Minimizing the cardinality of an events set for supervisors of discrete-event dynamical systems," *IEEE Trans. Automat. Control*, vol. 41, no. 11, pp. 1579–1593, Nov. 1996.

[8] T.-S. Yoo and S. Lafortune, "NP-completeness of sensor selection problems arising in partially observed discrete-event systems," *IEEE Trans. Automat. Control*, vol. 47, no. 9, pp. 1495–1499, Sep. 2002.

[9] S. Jiang, R. Kumar, and H. Garcia, "Optimal sensor selection for discrete-event systems with partial observation," *IEEE Trans. Automat. Control*, vol. 48, no. 3, pp. 369–381, Mar. 2003.

[10] J. Basilio, S. Lima, S. Lafortune, and M. Moreira, "Computation of minimal event bases that ensure diagnosability," *Discrete Event Dyn. Syst., Theory Appl.*, vol. 22, no. 3, pp. 249–292, 2012.

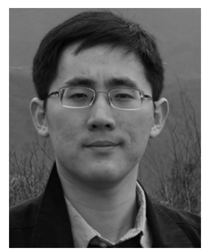
[11] D. Thorsley and D. Teneketzis, "Active acquisition of information for diagnosis and supervisory control of discrete event systems," *Discrete Event Dyn. Syst., Theory Appl.*, vol. 17, no. 4, pp. 531–583, 2007.

[12] F. Cassez and S. Tripakis, "Fault diagnosis with static and dynamic observers," *Fundam. Informat.*, vol. 88, no. 4, pp. 497–540, 2008.

[13] W. Wang, S. Lafortune, A. R. Girard, and F. Lin, "Optimal sensor activation for diagnosing discrete event systems," *Automatica*, vol. 46, no. 7, pp. 1165–1175, 2010.

[14] W. Wang, S. Lafortune, F. Lin, and A. R. Girard, "Minimization of dynamic sensor activation in discrete event systems for the purpose of control," *IEEE Trans. Automat. Control*, vol. 55, no. 11, pp. 2447–2461, Nov. 2010.

[15] S. Shu, Z. Huang, and F. Lin, "Online sensor activation for detectability of discrete event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 2, pp. 457–461, Apr. 2013.
[16] D. Sears and K. Rudie, "Efficient computation of sensor activation decisions in discrete-event systems," in *Proc. 52nd IEEE Conf. Decision Control*, 2013, pp. 6966–6971.
[17] E. Dallal and S. Lafortune, "On most permissive observers in dynamic sensor activation problems," *IEEE Trans. Automat. Control*, vol. 59, no. 4, pp. 966–981, Apr. 2014.
[18] X. Yin and S. Lafortune, "A general approach for solving dynamic sensor activation problems for a class of properties," in *Proc. 54th IEEE Conf. Decision Control*, 2015, pp. 3610–3615.
[19] X. Yin and S. Lafortune, "Codiagnosability and coobservability under dynamic observations: Transformation and verification," *Automatica*, vol. 61, pp. 241–252, 2015.
[20] D. Sears and K. Rudie, "Minimal sensor activation and minimal communication in discrete-event systems," *Discrete Event Dyn. Syst., Theory Appl.*, vol. 26, no. 2, pp. 295–349, 2016.
[21] Y.-C. Ho, "Team decision theory and information structures," *Proc. IEEE*, vol. 68, no. 6, pp. 644–654, Jun. 1980.
[22] S. Tripakis, "Undecidable problems of decentralized observation and control on regular languages," *Inf. Process. Lett.*, vol. 90, no. 1, pp. 21–28, 2004.
[23] J. Thistle, "Undecidability in decentralized supervision," *Syst. Control Lett.*, vol. 54, no. 5, pp. 503–509, 2005.
[24] A. Overkamp and J. H. van Schuppen, "Maximal solutions in decentralized supervisory control," *SIAM J. Control Optim.*, vol. 39, no. 2, pp. 492–511, 2000.
[25] J. H. van Schuppen, "Control of distributed stochastic systems-introduction, problems, and approaches," in *Proc. 18th IFAC World Congr.*, 2011, pp. 4446–4452.
[26] R. Boel and J. H. van Schuppen, "Decentralized failure diagnosis for discrete-event systems with costly communication between diagnosers," in *Proc. 6th Int. Workshop Discrete Event Syst.*, 2002, pp. 175–181.
[27] K. Rudie, S. Lafortune, and F. Lin, "Minimal communication in a distributed discrete-event system," *IEEE Trans. Automat. Control*, vol. 48, no. 6, pp. 957–975, Jun. 2003.
[28] W. Sadid, S. Ricker, and S. Hashtrudi-Zad, "Nash equilibrium for communication protocols in decentralized discrete-event systems," in *Proc. IEEE Amer. Control Conf.*, 2010, pp. 3384–3389.
[29] X. Yin and S. Lafortune, "Minimization of sensor activation in decentralized fault diagnosis of discrete event systems," in *Proc. 54th IEEE Conf. Decision Control*, 2015, pp. 1014–1019.
[30] C. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, 2nd ed. Berlin, Germany: Springer, 2008.
[31] W. Wang, S. Lafortune, and F. Lin, "An algorithm for calculating indistinguishable states and clusters in finite-state automata with partially observable transitions," *Syst. Control Lett.*, vol. 56, no. 9, pp. 656–661, 2007.
[32] D. Sears and K. Rudie, "On computing indistinguishable states of nondeterministic finite automata with partially observable transitions," in *Proc. 53rd IEEE Conf. Decision Control*, 2014, pp. 6731–6736.
[33] J. Tsitsiklis, "On the control of discrete-event dynamical systems," *Math. Control, Signals Syst.*, vol. 2, no. 2, pp. 95–107, 1989.
[34] K. Rudie and J. Willems, "The computational complexity of decentralized discrete-event control problems," *IEEE Trans. Automat. Control*, vol. 40, no. 7, pp. 1313–1319, Jul. 1995.
[35] Y. Huang, K. Rudie, and F. Lin, "Decentralized control of discrete-event systems when supervisors observe particular event occurrences," *IEEE Trans. Automat. Control*, vol. 53, no. 1, pp. 384–388, Feb. 2008.
[36] T.-S. Yoo and S. Lafortune, "Decentralized supervisory control with conditional decisions: Supervisor realization," *IEEE Trans. Automat. Control*, vol. 50, no. 8, pp. 1205–1211, Aug. 2005.
[37] S. Yokota, T. Yamamoto, and S. Takai, "Computation of the delay bounds and synthesis of diagnosers for decentralized diagnosis with conditional decisions," *Discrete Event Dyn. Syst., Theory Appl.*, vol. 27, no. 1, pp. 45–84, 2017.
[38] J. Dubreil, P. Darondeau, and H. Marchand, "Supervisory control for opacity," *IEEE Trans. Automat. Control*, vol. 55, no. 5, pp. 1089–1100, May 2010.
[39] H. Cho and S. Marcus, "On supremal languages of classes of sublanguages that arise in supervisor synthesis problems with partial observation," *Math. Control Signal. Syst.*, vol. 2, no. 1, pp. 47–69, 1989.
[40] S. Genc and S. Lafortune, "Predictability of event occurrences in partially-observed discrete-event systems," *Automatica*, vol. 45, no. 2, pp. 301–311, 2009.
[41] T. Yamamoto and S. Takai, "Conjunctive decentralized diagnosis of discrete event systems," in *Proc. 4th IFAC Workshop Dependable Control Discrete Syst.*, 2013, pp. 67–72.
[42] A. Khoumsi and H. Chakib, "Conjunctive and disjunctive architectures for decentralized prognosis of failures in discrete-event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 412–417, Apr. 2012.
[43] X. Yin and Z. Li, "Decentralized fault prognosis of discrete event systems with guaranteed performance bound," *Automatica*, vol. 69, pp. 375–379, 2016.



Xiang Yin (M'17) was born in Anhui, China, in 1991. He received the B.Eng. degree from Zhejiang University, Hangzhou, China, in 2012, and the M.S. and Ph.D. degrees from the University of Michigan, Ann Arbor, MI, USA, in 2013 and 2017, respectively, all in electrical engineering.

Since 2017, he has been with Shanghai Jiao Tong University, Shanghai, China, where he is currently an Associate Professor with the Department of Automation. His research interests include supervisory control of discrete-event systems, model-based fault diagnosis, formal methods, security, and their applications to cyber and cyber-physical systems.

Dr. Yin was a recipient of the Outstanding Reviewer Award from *Automatica* in 2016, the Outstanding Reviewer Award from the IEEE TRANSACTIONS ON AUTOMATIC CONTROL in 2017, and the IEEE Conference on Decision and Control (CDC) Best Student Paper Award Finalist in 2016. He is the co-Chair of the IEEE CSS Technical Committee on Discrete-Event Systems.

Stéphane Lafortune (F'99) received the B.Eng. degree from the Ecole Polytechnique de Montréal, Montréal, QC, Canada, in 1980, the M.Eng. degree from McGill University, Montréal, in 1982, and the Ph.D. degree from the University of California, Berkeley, CA, USA, in 1986, all in electrical engineering.

Since September 1986, he has been with the University of Michigan, Ann Arbor, MI, USA, where he currently is a Professor of electrical engineering and computer science. He is the lead developer of the software package UMDES and co-developer of DESUMA with L. Ricker. He coauthored, with C. Cassandras, the textbook entitled *Introduction to Discrete Event Systems* (2nd ed. New York, NY, USA: Springer, 2008). His research interests include discrete-event systems including multiple problem domains: modeling, diagnosis, control, optimization, and applications to computer and software systems.

Dr. Lafortune was a recipient of the Presidential Young Investigator Award from the National Science Foundation in 1990 and the George S. Axelby Outstanding Paper Award from the IEEE Control Systems Society in 1994 (for a paper coauthored with S.-L. Chung and F. Lin) and in 2001 (for a paper coauthored with G. Barrett). He is the Editor-in-Chief of the *Journal of Discrete Event Dynamic Systems: Theory and Applications*.