2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Constrained Sensor Selection for Discrete Event Systems Modeled by Petri Nets

Yu Ru and Christoforos N. Hadjicostis

***Abstract*— This paper studies how to place a minimum number of sensors in discrete event systems modeled by partially observed Petri nets while maintaining structural observability. When the sensors are constrained to be associated with specific sets of transitions (which could be the result of physical or geographical constraints), the resulting constrained optimal sensor selection problems are shown to be reducible to the optimal place sensor selection (OPSS) problem introduced in previous work. These reductions establish the central role that the OPSS problem plays in our sensor selection problem formulation. Therefore, in order to obtain a solution with known performance guarantees (that we precisely characterize), we propose in this paper a heuristic method based on a reduction from the OPSS problem to the set cover problem.**

I. INTRODUCTION

A discrete event system is a dynamic system that evolves in accordance with the abrupt occurrence, at possibly unknown and irregular intervals, of events [1]. Such systems arise in a variety of contexts, ranging from energy distribution networks and automated manufacturing systems to communication networks and air traffic control systems. Applications that involve monitoring and controlling of such systems rely on information conveyed by various types of sensors that are available in the system. Usually it is unnecessary/impossible to place sensors everywhere because sensors may be unavailable or prohibitively expensive for certain state transitions and/or partial system states. Therefore, selecting a minimum number of sensors (or a set of sensors of minimal cost) that meet the system design requirements is critical and often mandatory.

Optimal sensor selection problems have been studied extensively in discrete event systems that can be modeled as finite state machines, in which only partial state transitions can be observed (e.g., [2]). In contrast, there is limited previous work on sensor selection problems when the underlying model is a Petri net [3], [4]. In [3], observability notions based on inputs and outputs (namely, partial state information) are used as criteria when optimizing the selection of sensors in bounded interpreted Petri net models. In [4], both partial state and partial transition information can be observed, and structural observability (refer to Definition 3) is considered a necessary requirement when optimizing the selection of sensors in general Petri net models. The general sensor selection problem (in which both place and transition sensors can be selected) is difficult; therefore, the optimal place sensor selection (OPSS) problem with fixed transition sensors, and the optimal transition sensor selection problem with fixed place sensors, are considered in [4] to gain a better understanding of sensor selection problems.

In this paper we further study the optimal transition sensor selection problem and the general sensor selection problem by considering constraints that are imposed on the way transitions can share sensors. We show that both constrained problems can be converted to an OPSS problem, which establishes the central role that the OPSS problem plays in these sensor selection problems. Then, we propose a heuristic method by establishing a reduction from the OPSS problem to the set cover problem (SCP) and by utilizing a well-known greedy algorithm for SCP [5]. The advantage of this method over the top-down and bottom-up methods introduced in [4] is that it offers performance guarantees. In addition, we establish a reduction from SCP to the OPSS problem, which proves the $\mathcal{NP}$-completeness of the OPSS problem (following a route different from the one in [4]).

This material is based upon work supported in part by the National Science Foundation, under NSF CNS Award 0834409. The research leading to these results has also received funding from the European Commission (EC) Seventh Framework Programme (FP7/2007-2013), under grant agreements INFSO-ICT-223844 and PIRG02-GA-2007-224877. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of NSF or EC.

Yu Ru was with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign (e-mail: yuru2@illinois.edu); he is now with the Department of Mechanical and Aerospace Engineering, University of California, San Diego. C. N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus, and also with the Coordinated Science Laboratory, and the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign (e-mail: chadjic@ucy.ac.cy).

II. PRELIMINARIES

Definition 1 [6] A *Petri net structure* is a 4-tuple $N = (P, T, F, W)$, where $P = \{p_1, p_2, ..., p_n\}$ is a finite set of n places; $T = \{t_1, t_2, ..., t_m\}$ is a finite set of m transitions; $F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs; $W : F \to \{1, 2, 3, ...\}$ is a weight function; $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

A *marking* is a function $M : P \to \mathcal{N}_0$ that assigns to each place a nonnegative integer number of tokens ($\mathcal{N}_0$ denotes the set of nonnegative integers); $M(p)$ denotes the number of tokens in place p. Pictorially, places are represented by circles, transitions by bars, and tokens by black dots, as shown in Fig. 1. A *Petri net* $G = \langle N, M_0 \rangle$ is a Petri net structure N with an initial marking M_0.

A transition t is said to be *enabled* at marking M if each input place p of t (i.e., each place p such that $(p, t) \in F$) is marked with at least $W(p, t)$ tokens; this is denoted by $M[t\rangle$. The firing of an enabled transition t removes $W(p, t)$ tokens

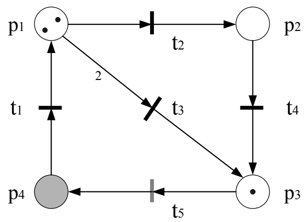

Fig. 1. A partially observed Petri net Q [4].

from each input place p and adds $W(t, p')$ tokens to each output place p' (i.e., each place p' such that $(t, p') \in F$), resulting in a marking M'; this is denoted by $M[t\rangle M'$. In this paper, we assume that at most one transition can fire at any instant. Notation $S = t_{s_1} t_{s_2} \cdots t_{s_k}$ denotes a k-length *firing sequence* from marking M if $t_{s_i} \in T$ and $M[t_{s_1}\rangle M_1[t_{s_2}\rangle M_2 \cdots [t_{s_k}\rangle M'$; this is denoted by $M[S\rangle M'$. Marking M' can also be written as $M' = M + D\sigma$, where (i) D is the $n \times m$ *incidence matrix* of N satisfying $D(i,j) = -W(p_i, t_j) + W(t_j, p_i)$ (if $W(p_i, t_j)$ or $W(t_j, p_i)$ is not defined for a specific place p_i and transition t_j, it is taken to be 0), and (ii) σ is the $m \times 1$ *firing vector* of S with its ith entry being the number of times transition t_i appears in S. In this paper, we assume that Petri nets do not have self-loops (however, this assumption is not essential since a Petri net with self-loops can be transformed into a Petri net without self-loops [6]).

Definition 2 [4] A *partially observed Petri net* Q is a 3-tuple (N, P_o, T_o), where

- $N = (P, T, F, W)$ with $|P| = n$;
- $P_o \subseteq P$, is the set of observable places; without loss of generality, P_o is taken to be $\{p_1, p_2, ..., p_{n_1}\}$ with $0 \leq n_1 \leq n$;
- $T_o \subseteq T$, is the set of observable transitions.

Observable places can have sensors (e.g., vision sensors) that indicate the number of tokens in a particular place, but unobservable places (denoted by $P_{uo} = P \backslash P_o$) cannot. The association between sensors and places is captured by the *place sensor configuration* $V = (v_1 \ v_2 \ \ldots \ v_{n_1})^T$, where $v_i = 1$ if a place sensor is selected for place p_i and $v_i = 0$ otherwise. $||V|| := \sum_{i=1}^{n_1} v_i \leq n_1$ denotes the total number of sensors in the place sensor configuration V. Given a place sensor configuration V, the $||V|| \times m$ matrix D_V is constructed by keeping the rows of D that correspond to observable places with sensors.

Similarly, $T_{uo} = T \backslash T_o$ denotes the set of unobservable transitions. Observable transitions can have sensors (e.g., motion sensors) that indicate when a transition in a given subset of transitions has fired, but unobservable transitions cannot. In general, the association between sensors and transitions is captured by the *labeling function* $L : T \rightarrow \Sigma \cup \{\varepsilon\}$, which assigns a label to each transition and satisfies $L(t) = \varepsilon$ for any $t \in T_{uo}$. Here, Σ is the set of labels and ε is the empty label. We define Σ so that, for each $e \in \Sigma$ there exists $t \in T_o$ satisfying $L(t) = e$. Therefore, $|\Sigma|$ is the total number of transition sensors in use and could be zero if no transition sensor is used. When an observable transition t with a sensor fires, the label $L(t)$ is observed. If $L(t) = \varepsilon$, then the firing of transition t is not observed at all. We define $T_e := \{t \in T : L(t) = e\}$ for any $e \in \Sigma \cup \{\varepsilon\}$.

Example 1 The net in Fig. 1 is a partially observed Petri net with $P_o = \{p_1, p_2, p_3\}$ and $T_o = \{t_1, t_2, t_3, t_4\}$. Suppose $V = (0 \ 0 \ 1)^T$, L is defined as $L(t_1) = L(t_2) = a$, $L(t_3) = L(t_4) = b$, and $L(t_5) = \varepsilon$. If $M_0 = (2 \ 0 \ 1 \ 0)^T$ and $t_3 t_5$ occurs, then the system trajectory is $M_0[t_3\rangle M_1[t_5\rangle M_2$, where $M_1 = (0 \ 0 \ 2 \ 0)^T$ and $M_2 = (0 \ 0 \ 1 \ 1)^T$. The available sensing information is $1 \rightarrow b \rightarrow 2 \rightarrow 1$, where $\rightarrow$ denotes the temporal order of observations. In general, the sensing information of state transition $M_i[t\rangle M_{i+1}$ is $M_i^V \rightarrow L(t) \rightarrow M_{i+1}^V$, where M_i^V is obtained by keeping any jth entry of M_i for which $V(j) = 1$. Note that if $L(t) = \varepsilon$, $M_i^V \rightarrow L(t) \rightarrow M_{i+1}^V$ reduces to $M_i^V \rightarrow M_{i+1}^V$ (or M_i^V) if $M_i^V \neq M_{i+1}^V$ (or $M_i^V = M_{i+1}^V$). In particular, note that unobservable transitions that do not cause token changes in places with sensors go unrecorded. ■

III. Problem Formulation

Definition 3 [4] Given a place sensor configuration V and a labeling function L, a partially observed Petri net Q is *structurally observable* if for an *arbitrary* but known initial state M_0 and *any* firing sequence from M_0, the system state M at any given time step can be determined uniquely based on observations from place sensors and transition sensors up to that time step.

Structural observability requires the accurate determination of the current system state at any given time step, and is motivated by applications where it is necessary to accurately represent the underlying system state (for details, refer to [4]).

We assume without loss of generality that there are *no identically behaving transitions* (namely, transitions that have identical columns in the incidence matrix D) in the Petri net, so that structural observability is essentially equivalent to transition distinguishability as introduced in [4]. The following proposition can be derived from Propositions 1 and 2 in [4], and is used to determine structural observability.

Proposition 1 Given a place sensor configuration V and a labeling function L, a partially observed Petri net Q is structurally observable if and only if i) for each label $e \in \Sigma$, all columns of D_V^e are pairwise different, and ii) for ε, all columns of D_V^ε are nonzero and pairwise different, where D_V^e for $e \in \Sigma \cup \{\varepsilon\}$ is obtained by keeping the columns in D_V that correspond to transitions in T_e.

Given a partially observed Petri net, the general sensor selection problem consists of choosing a place sensor configuration V and a labeling function L such that $||V|| + |\Sigma|$ is minimized (or, more generally, the total cost of all sensors in use is minimized) *and* the system is structurally observable under V and L. To gain a better understanding of the general sensor selection problem, we have studied the following subproblem.

Problem 1 [4] (Optimal Place Sensor Selection (OPSS)) Given a partially observed Petri net Q and a fixed labeling function L, find V such that i) the system is structurally observable under V and L, and ii) V minimizes the number of place sensors $||V||$.

Checking the existence of a feasible solution to the OPSS problem is provided in Theorem 1 of [4]. Naturally, one can obtain the other subproblem called optimal transition sensor selection by fixing a place sensor configuration V. In the optimal transition sensor selection problem, we implicitly assume that a nonempty label can be associated to any subset of observable transitions. However, this assumption may not be realistic in certain applications due to topological, or other constraints; for instance, it might be the case that only physically close transitions (in distributed systems) can share the same label. To capture such requirements, we introduce the following constraints on transition sensors: i) there are d types of transition sensors $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_d$; and ii) each type $\mathcal{T}_i$ covers a subset of observable transitions while some transitions may not be covered and some transitions may be covered by more than one type of sensors. If a transition t is covered by a type $\mathcal{T}_i$ transition sensor, then the label $e_{\mathcal{T}_i}$ will be observed if t fires; if t is covered by more than one type of transition sensors (e.g., covered by both type $\mathcal{T}_i$ and type $\mathcal{T}_j$ transition sensors), then all associated labels will be simultaneously observed if t fires (e.g., labels $e_{\mathcal{T}_i}$ and $e_{\mathcal{T}_j}$ will be observed simultaneously, or equivalently, a single label $e_{\mathcal{T}_i\mathcal{T}_j}$ will be observed). Now, we define a *transition sensor configuration* $W = (w_1\ w_2\ ...\ w_d)^T$, where $w_i = 0$ if no type $\mathcal{T}_i$ transition sensor exists for transitions in $\mathcal{T}_i$ and $w_i = 1$ otherwise. We use $||W|| := \sum_{i=1}^{d} w_i \leq d$ to denote the total number of transition sensors in the configuration W. Given a transition sensor configuration W, we can construct an equivalent labeling function L_W as shown in the following example.

Example 2 For the partially observed Petri net shown in Fig. 1, suppose there are two types of transition sensors: $\mathcal{T}_1 = \{t_1, t_2\}$ (which means the sensor covers transitions t_1 and t_2) and $\mathcal{T}_2 = \{t_2, t_3\}$. If $W = (1\ 1)^T$, then the equivalent labeling function L_W is $L_W(t_1) = e_{\mathcal{T}_1}$, $L_W(t_2) = e_{\mathcal{T}_1\mathcal{T}_2}$, $L_W(t_3) = e_{\mathcal{T}_2}$, $L_W(t_4) = L_W(t_5) = \varepsilon$. The labeling function is equivalent to the transition sensor configuration in the sense that the outputs from both are essentially the same given the same system activities. It is straightforward to generalize the construction of L_W to an arbitrary transition sensor configuration W. ■

Problem 2 (Constrained Optimal Transition Sensor Selection (COTSS)) Given a partially observed Petri net Q, a fixed place sensor configuration V and d types of transition sensors $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_d$, find W such that i) the system is structurally observable under V and L_W, and ii) W minimizes the number of transition sensors $||W||$.

Problem 3 (Constrained General Sensor Selection (CGSS)) Given a partially observed Petri net Q, and d types of transition sensors $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_d$, find V and W such that i) the system is structurally observable under V and L_W, and ii) V and W minimize the number of sensors $||V|| + ||W||$.

IV. Constrained Sensor Selection

A. Constrained Optimal Transition Sensor Selection

In the COTSS problem, we can construct a labeling function L_V which provides essentially the same sensing information as the place sensor configuration V. Before introducing the construction, we look at the partition of T generated by V.

Definition 4 Given a partially observed Petri net Q and a fixed place sensor configuration V, the *partition of* T *generated by* V is defined to be $\Omega(V) = \{S_0, S_1, S_2, ..., S_k\}$, where i) k is equal to the number of distinct nonzero columns in the matrix D_V, ii) $S_0 \cup S_1 \cup S_2 \cup \cdots \cup S_k = T$ and $S_i \cap S_j = \emptyset$ if $i \neq j$, iii) S_0 is a (possibly empty) set with all transitions $t_j, \cdots, t_l \in T$ that satisfy $D_V(:, j) = \cdots = D_V(:, l) = \mathbf{0}_{||V||}$, where $D_V(:, j)$ denotes the jth column of matrix D_V and $\mathbf{0}_{||V||}$ is an all 0 vector with dimension $||V||$; iv) S_i for $i = 1, 2, ..., k$ is a non-empty set with the maximal number of transitions $\{t_j, ..., t_l\}$, $t_j, \cdots, t_l \in T$, that satisfy $D_V(:, j) = \cdots = D_V(:, l) \neq \mathbf{0}_{||V||}$.

The firings of transitions in S_0 cannot generate any place sensor output; equivalently, we could assign the empty label to all transitions in S_0, i.e., $L_V(t) = \varepsilon$ for $t \in S_0$. For each S_i, $i = 1, 2, ..., k$, the firings of transitions in S_i generate a unique combination of token changes among all places with sensors in V (but this combination can be generated by any of these transitions). Equivalently, we could assign the label e_{S_i} to all transitions in S_i, i.e., $L_V(t) = e_{S_i}$ for $t \in S_i$.

Once we have the equivalent labeling function, we can construct an instance of the OPSS problem given a COTSS instance, as illustrated in the following example.

Example 3 Consider the partially observed Petri net Q in Fig. 1 with $V = (0\ 0\ 1)^T$, and two types of transition sensors with $\mathcal{T}_1 = \{t_1, t_2\}$ and $\mathcal{T}_2 = \{t_2, t_3\}$. The partition of T generated by V is $S_0 = \{t_1, t_2\}$, $S_1 = \{t_3, t_4\}$ and $S_2 = \{t_5\}$, as can be readily obtained from matrix $D_V = (0\ 0\ 1\ 1\ -1)$. The equivalent labeling function L_V is $L_V(t_1) = L_V(t_2) = \varepsilon$, $L_V(t_3) = L_V(t_4) = e_{S_1}$, and $L_V(t_5) = e_{S_2}$. Clearly, with this change, the COTSS problem can be interpreted as follows: given a labeling function L_V and two types of transition sensors $\mathcal{T}_1$ and $\mathcal{T}_2$, find a transition sensor configuration $W = (w_1\ w_2)^T$ such that all transitions are distinguished (refer to Proposition 1) and $||W||$ is minimized.

Now we construct an instance of the OPSS problem from this COTSS problem. Consider a partially observed Petri net Q' with 5 observable transitions $t'_1, ..., t'_5$ corresponding to $t_1, ..., t_5$, 2 observable places p'_1, p'_2 corresponding to the two types of transition sensors, and labeling function defined as $L'(t'_1) = L'(t'_2) = \varepsilon$, $L'(t'_3) = L'(t'_4) = e_{S_1}$, $L'(t'_5) = e_{S_2}$, which is essentially L_V. The incidence matrix D' is

$$D' = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

and is constructed based on the coverage of different types of transition sensors; for example, as sensor type $\mathcal{T}_1$ covers t_1, t_2, the place p'_1 corresponding to $\mathcal{T}_1$ can distinguish t'_1, t'_2 from other transitions and[1] $D'(p'_1, :) = (1\ 1\ 0\ 0\ 0)$. Since transitions t'_4 and t'_5 are identically behaving transitions, one could add "dummy" unobservable place p'_3 with[2] $D'(p'_3, :) = (0\ 0\ 0\ 1\ {-1})$ to resolve this issue. The goal in the constructed OPSS problem is to find V' with a minimum number of sensors such that all transitions can be distinguished. For this example, it is easy to see that Q being structurally observable under V and L_W is equivalent to Q' being structurally observable under $V' := W$ and L'. ■

In general, given an instance of the COTSS problem, one can construct an instance of the OPSS problem as shown in Algorithm 1. Now we briefly argue the correctness of the reduction. Note that for any transition sensor configuration W in Q, there is a place sensor configuration $V' := W$ in Q' because observable place p'_i in Q' corresponds to the type of transition sensors $\mathcal{T}_i$ in Q, and vice versa. Also, Q being structurally observable under V and L_W is equivalent to Q' being structurally observable under L' and $V' := W$ because the construction of L' provides the same distinguishability on transitions as the given V (also note that L' is constructed from L_V and that L_V is equivalent to V). It can be verified that Algorithm 1 has complexity $\mathcal{O}(nm^2 + dm)$, where d is the number of different types of transition sensors.

Algorithm 1 Reduction from COTSS to OPSS

Input: An instance of Problem 2

Output: An instance of Problem 1

1: Calculate $\Omega(V) = \{S_0, S_1, ..., S_k\}$ based on Definition 4, and construct the labeling function L_V satisfying $L_V(t) = \varepsilon$ if $t \in S_0$ and $L_V(t) = e_{S_i}$ if $t \in S_i$.

2: Construct Q': $T' = T'_o = \{t'_1, t'_2, ..., t'_m\}$, $P'_o = \{p'_1, p'_2, ..., p'_d\}$, $L'(t'_i) = L_V(t_i)$, and D' satisfying $D'(p'_i, t'_j) = 1$ (for $i = 1, ..., d$ and $j = 1, ..., m$) if t_j is covered by $\mathcal{T}_i$ and $D'(p'_i, t'_j) = 0$ otherwise.

3: Check if there are identically behaving transitions. If so, then add unobservable place p'_{d+1} (i.e., $P' = P'_o \cup \{p'_{d+1}\}$) and assign $D'(p'_{d+1}, :)$ so that identically behaving transitions are eliminated; otherwise, $P' = P'_o$.

4: Output the OPSS problem instance with Q' and L'.

B. Constrained General Sensor Selection

Example 4 Consider the partially observed Petri net Q in Fig. 1 with 2 types of transition sensors satisfying $\mathcal{T}_1 = \{t_1, t_2\}$ and $\mathcal{T}_2 = \{t_2, t_3\}$. The CGSS problem asks to obtain a place sensor configuration V and a transition sensor configuration W such that Q is structurally observable under V and L_W, and $||V|| + ||W||$ is minimized.

[1]The constraint we have regarding $D'(p'_1, :)$ is that the first two entries should be the same and nonzero, and the last three entries should all be zero. Therefore, other choices are also possible (e.g., $D'(p'_1, :) = (-1\ {-1}\ 0\ 0\ 0)$).

[2]The only constraint we have regarding $D'(p'_3, :)$ is that $D'(p'_3, t'_4)$ and $D'(p'_3, t'_5)$ should be different.

Now we can construct an instance of the OPSS problem from this CGSS problem. Consider a partially observed Petri net Q' with 5 observable transitions $t'_1, ..., t'_5$ corresponding to $t_1, ..., t_5$; 6 places $p''_1, p''_2, p'_1, p'_2, p'_3, p'_4$ in which p''_1, p''_2 correspond to the two types of transition sensors, p'_1, p'_2, p'_3, p'_4 correspond to p_1, p_2, p_3, p_4 in Q, and only p'_4 is unobservable; labeling function defined as $L'(t'_1) = L'(t'_2) = L'(t'_3) = L'(t'_4) = L'(t'_5) = \varepsilon$. The incidence matrix D' is $(U^T\ D^T)^T$, where D is the incidence matrix of Q and U is the same as D' in Example 3. Since there are no identically behaving transitions in Q, neither are there in Q'. The goal in the constructed OPSS problem is to find V' such that all transitions can be distinguished. For this example, it is easy to see that Q being structurally observable under V and L_W is equivalent to Q' being structurally observable under $V' := (W^T\ V^T)^T$ and L'. ■

In general, given an instance of the CGSS problem, one can construct an instance of the OPSS problem as shown in Algorithm 2. Now we briefly argue the correctness of the reduction. Note that for any place sensor configuration V and any transition sensor configuration W in Q, there is a place sensor configuration $V' := (W^T\ V^T)^T$ in Q' because observable place p''_i in Q' (for $i = 1, ..., d$) corresponds to the type of transition sensors $\mathcal{T}_i$ in Q and observable place p'_j in Q' (for $j = 1, ..., n_1$) corresponds to observable place p_j in Q, and vice versa. Also, Q being structurally observable under V and L_W is equivalent to Q' being structurally observable under L' and $V' = (W^T\ V^T)^T$ because the labeling function L' essentially outputs nothing, and the construction of V' provides the same distinguishability on transitions as the combination of V and W. It can be verified that Algorithm 2 has complexity $\mathcal{O}(dm + nm)$.

Algorithm 2 Reduction from CGSS to OPSS

Input: An instance of Problem 3

Output: An instance of Problem 1

1: Construct Q': $T' = T'_o = \{t'_1, t'_2, ..., t'_m\}$, $P' = \{p''_1, p''_2, ..., p''_d, p'_1, p'_2, ..., p'_n\}$ of which only $p'_{n_1+1}, ..., p'_n$ are unobservable, $L'(t'_i) = \varepsilon$ for any $t'_i \in T'$, and $D' = (U^T\ D^T)^T$ where D is the incidence matrix of Q, and $U(p''_i, t'_j) = 1$ (for $i = 1, ..., d$ and $j = 1, ..., m$) if t_j is covered by $\mathcal{T}_i$ and $U(p''_i, t'_j) = 0$ otherwise.

2: Output the OPSS problem instance with Q' and L'.

V. OPSS: Approximation Algorithm

A. Reduction from OPSS to SCP

Problem 4 [5] (Set Cover Problem (SCP)) Given a universe $\mathcal{U}$ of q elements, and a collection of subsets of $\mathcal{U}$, $\mathcal{S} = \{S_1, ..., S_k\}$, find a minimum number of subsets of $\mathcal{S}$ that cover all elements of $\mathcal{U}$.

Example 5 Consider the partially observed Petri net Q in Fig. 1 with L satisfying $L(t_1) = a$, $L(t_2) = L(t_3) = b$, $L(t_4) = c$ and $L(t_5) = \varepsilon$. The OPSS problem requires a place sensor configuration V such that Q is structurally observable under V and the given L, and $||V||$ is minimized. An equivalent binary integer programming (BIP) problem can be constructed as follows [4]:

$$\min \quad c^T x \qquad s.t. \;\; Ax \geq b$$

where $c = (1\ 1\ 1)^T$, $x = V = (v_1\ v_2\ v_3)^T$ (as p_1, p_2 and p_3 are all observable), $b = (1\ 1)^T$, and[3]

$$A = \begin{bmatrix} -1 \neq -2 & 1 \neq 0 & 0 \neq 1 \\ 0 \neq 0 & 0 \neq 0 & -1 \neq 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}.$$

The first constraint captures the requirement that V distinguish t_2 from t_3, while the second constraint captures the requirement that V detect unobservable transition t_5.

Let $k = n_1$ which is 3, $q = 2$ which is the number of constraints in the BIP problem. Set $\mathcal{U} = \{1, 2\}$, and $S_1 = \{1\}$, $S_2 = \{1\}$ and $S_3 = \{1, 2\}$ (these subsets are obtained by reading the nonzero entries in each column of the matrix A; for example, S_1 contains only the element 1 because only the 1st entry of $A(:, 1)$ is 1). With this construction, each constraint in the BIP problem is equivalent to the requirement on these subsets to cover the corresponding element in $\mathcal{U}$. Since the objective function of the BIP formulation is equivalent to minimizing the number of subsets in SCP, the constructed SCP is equivalent to the OPSS problem. ■

Given an instance of the OPSS problem, one could construct an instance of SCP as shown in Algorithm 3. Now we briefly show the correctness of the reduction. Note that for any place sensor configuration V, there is a unique combination of subsets of $\mathcal{U}$ (namely $\{S_i \mid V(i) \text{ is } 1\}$), and vice versa. Also, Q being structurally observable under V and L is equivalent to satisfying $AV \geq \mathbf{1}_q$ (as shown in [4]), which is equivalent to the requirement that the combination of subsets of $\mathcal{U}$ (corresponding to V) covers each element of the universe $\mathcal{U}$. It can be verified that Algorithm 3 has complexity $\mathcal{O}(nm^2)$.

Algorithm 3 Reduction from OPSS to SCP

Input: An instance of Problem 1

Output: An instance of Problem 4

1: Set $q = |T_\varepsilon| + \sum_{e \in \Sigma \cup \{\varepsilon\}, |T_e| \geq 2} \frac{|T_e|(|T_e|-1)}{2}$, and $k = n_1$.
2: Set $\mathcal{U} = \{1, 2, ..., q\}$.
3: Set A to be a $q \times n_1$ binary matrix with two kinds of rows: a) $\forall e \in \Sigma \cup \{\varepsilon\}$ with $|T_e| \geq 2$, for each pair $t_j, t_k \in T_e$ $(j \neq k)$, there is a row of the form $(D(1,j) \neq D(1,k)\ \ D(2,j) \neq D(2,k)\ \ \cdots\ \ D(n_1,j) \neq D(n_1,k))$; b) for each $t_j \in T_\varepsilon$, there is a row of the form $(D(1,j) \neq 0\ \ D(2,j) \neq 0\ \ \cdots\ \ D(n_1,j) \neq 0)$.
4: Set $S_i = \{j \mid j \in \mathcal{U} \text{ and } A(j,i) \text{ is } 1\}$ for $i = 1, 2, ..., n_1$.
5: Output the set cover problem.

One can also reduce SCP to OPSS. Given an instance of Problem 4, here is a sketch of a procedure to construct an instance of Problem 1: i) construct a partially observed Petri net with $T = \{t_1, t_2, ..., t_{2q}\}$ and k observable places, ii) let $\Sigma = \{e_1, e_2, ..., e_q\}$ and define L as $L(t_{2i-1}) = L(t_{2i}) = e_i$ for $i = 1, ..., q$, iii) define the incidence matrix D as follows: $D(j, 2i-1) \neq D(j, 2i)$ if S_j covers the element i in $\mathcal{U}$, and $D(j, 2i-1) = D(j, 2i)$ otherwise, for $j = 1, ..., k$ and $i = 1, ..., q$, iv) add one unobservable place p_{k+1} if there are identically behaving transitions and set $P = \{p_1, p_2, ..., p_{k+1}\}$; otherwise, set $P = \{p_1, p_2, ..., p_k\}$. The correctness of this reduction can be verified based on the following observations: i) the subset S_j covers the element i if and only if a sensor on place p_j distinguishes transition t_{2i-1} from t_{2i}; and ii) minimizing the number of subsets that cover all elements of $\mathcal{U}$ is equivalent to minimizing the number of place sensors in a place sensor configuration under which the system is structurally observable. It can be verified that this is a polynomial reduction. Since the set cover problem is $\mathcal{NP}$-complete [5], this reduction also establishes the $\mathcal{NP}$-completeness of the OPSS problem (following a route different from reducing the vertex cover problem to OPSS in [4]).

B. Greedy Algorithm for OPSS

A well-known greedy algorithm for the set cover problem selects, at each time, the subset S_i that can cover the most elements in the universe that have not been covered so far, and terminates when all elements are covered. The algorithm is guaranteed to provide a solution within $OPT * H_q$ [5], where OPT is the optimal (smallest) number of subsets and $H_q = 1 + \frac{1}{2} + ... + \frac{1}{q}$ (note that H_q is $\mathcal{O}(\ln q)$).

Once we reduce the OPSS problem to the set cover problem and utilize the known greedy algorithm, the method guarantees a place sensor configuration with the number of place sensors within $OPT * H_q$, where OPT is the optimal number of place sensors and q is $|T_\varepsilon| + \sum_{e \in \Sigma \cup \{\varepsilon\}, |T_e| \geq 2} \binom{|T_e|}{2}$ as given in Algorithm 3. Note that the factor H_q is roughly $\mathcal{O}(\ln m)$ because q is $\mathcal{O}(m^2)$.

VI. Example

In this section, we consider the OPSS problem in a flexible manufacturing cell, which includes three workstations, two part-receiving stations and one completed part station. The Petri net model (with 64 places and 53 transitions) of the cell is shown in Fig. 2 of [7] (for our simulations on sensor selection, we do not need the control places in that figure). We use this example to compare heuristic methods, i.e., the top-down method, the bottom-up method, the combined method (applying the top-down method after the bottom-up method), and the method based on the reduction to the set cover problem (called the SCP based method). The first three heuristic methods are proposed in [4] but the performance of the combined method was not studied in [4].

As in [4], we model the cell as a partially observed Petri net and assume that all places and transitions are observable so that there is at least one solution to the constructed OPSS problems. To test the effectiveness of our approximation methods, we generate labeling functions in the following way: i) we first specify the number of transition labels i, ii) then, we let i take values $10, 13, 16, 20, 24, 30$, and for

[3]Here, for integers a and b, $a \neq b$ has value 1 if a is not equal to b, and 0 otherwise.

each value of i, we randomly generate 5 labeling functions by allowing each transition t to have any of the i labels with equal probability. In total, we have 30 randomly generated labeling functions; then we solve the 30 OPSS problems using the four heuristic methods and the BIP based method as introduced in [4]. Simulation programs were written in Matlab and were run on a 1.4Ghz laptop. The results are shown in Table I, in which i refers to the number of transition labels, q is defined in Algorithm 3 (note that q captures the number of constraints in the BIP formulation), and $OPT*H_q$ is the performance guarantee for the SCP based method as mentioned in Section V-B. The results suggest that the four heuristic methods run much faster than the BIP based method especially when there are less transition sensors. Among these four heuristic methods, the SCP based method is the fastest one. The number of sensors generated by the SCP based method indeed satisfies the bound $OPT*H_q$ as shown in the table.

We compare the four heuristic methods with the BIP-based method and Table II shows the comparison results when considering the difference Δ between the number of sensors given by heuristic methods and the number given by the BIP-based method. The combined method has the best performance among all heuristic methods: 28 out of 30 simulations give a very close solution (namely, $\Delta \leq 1$). In contrast, the SCP based method performs even worse than the top-down method (but comes with performance guarantees).

VII. Conclusion

In this paper, we studied the constrained sensor selection problems in discrete event systems modeled by partially observed Petri nets. We showed that constrained sensor selection problems can be converted to the OPSS problem, and proposed a heuristic method for the OPSS problem with known performance guarantees. In future work, we would like to directly establish the performance guarantees for the bottom-up method we proposed earlier. We would also like to relax the notion of structural observability to incorporate the initial state of the Petri net, take possible permanent or intermittent faulty sensors into account, and associate different costs with place sensors and transitions sensors.

References

[1] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, pp. 81–98, Jan. 1989.

[2] T. Yoo and S. Lafortune, "NP-completeness of sensor selection problems arising in partially observed discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 47, pp. 1495–1499, Sep. 2002.

[3] L. Aguirre-Salas, "Sensor selection for observability in interpreted Petri nets: a genetic approach," in *Proc. of 42nd IEEE Conf. on Decision and Control*, Dec. 2003, pp. 3760–3765.

[4] Y. Ru and C. N. Hadjicostis, "Sensor selection for structural observability in discrete event systems modeled by Petri nets," *IEEE Transactions on Automatic Control*, vol. 55, pp. 1751–1764, Aug. 2010.

[5] V. V. Vazirani, *Approximation Algorithms*. Berlin, Germany: Springer-Verlag, 2004.

[6] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, pp. 541–580, Apr. 1989.

[7] L. E. Holloway and B. H. Krogh, "Synthesis of feedback control logic for a class of controlled Petri nets," *IEEE Transactions on Automatic Control*, vol. 35, pp. 514–523, May 1990.

TABLE I

Simulation Results of 4 Heuristic Methods and BIP Based Method

	Top-down		Bottom-up		Combined	
i	time (s)	# sensors	time (s)	# sensors	time (s)	# sensors
	0.343	14	1.688	15	1.751	15
30	0.328	17	1.922	16	1.984	16
	0.359	15	1.735	15	1.814	15
	0.344	18	1.812	15	1.874	15
	0.359	17	1.844	16	1.907	15
	0.406	18	2.203	18	2.297	18
24	0.391	16	2.125	16	2.219	16
	0.390	17	2.156	17	2.250	17
	0.359	17	2.063	16	2.141	16
	0.391	18	2.343	17	2.421	17
	0.422	20	2.547	18	2.640	18
20	0.406	19	2.266	17	2.360	17
	0.407	18	2.235	17	2.328	17
	0.438	18	2.703	18	2.813	18
	0.406	18	2.250	18	2.344	18
	0.406	20	2.437	21	2.546	19
16	0.406	21	2.890	20	3.030	20
	0.406	20	2.219	20	2.297	20
	0.391	19	2.219	20	2.297	20
	0.359	21	2.281	19	2.375	19
	0.390	22	2.453	21	2.562	21
13	0.406	22	2.234	21	2.343	21
	0.406	22	2.282	21	2.376	21
	0.390	21	2.562	21	2.656	21
	0.390	21	2.266	22	2.376	21
	0.390	23	2.360	23	2.470	22
10	0.438	23	2.219	22	2.313	22
	0.406	23	2.469	22	2.563	22
	0.391	22	2.297	22	2.406	22
	0.422	23	2.375	22	2.485	22

	SCP based Method		BIP based Method			
i	time (s)	# sensors	time (s)	# sensors	q	$OPT*H_q$
	0.046	33	0.203	13	38	55.0
30	0.047	44	0.609	15	52	68.1
	0.047	27	0.156	14	41	60.2
	0.047	25	0.640	15	44	65.6
	0.047	29	0.266	14	42	60.6
	0.094	40	8.266	17	70	82.2
24	0.047	36	7.000	16	49	71.7
	0.062	22	0.812	15	49	67.2
	0.062	28	1.703	16	51	72.3
	0.063	31	1.562	16	56	73.8
	0.078	27	30.984	18	69	86.7
20	0.078	36	6.781	17	61	79.8
	0.078	39	1.562	17	65	80.9
	0.063	29	36.297	18	59	83.9
	0.156	50	256.328	18	70	87.0
	0.125	34	296.719	19	87	96.0
16	0.109	32	280.437	19	74	92.9
	0.141	39	127.359	19	91	96.8
	0.093	30	67.390	19	79	94.1
	0.094	36	49.875	19	93	97.2
	0.094	31	75.406	20	97	103.1
13	0.109	39	136.547	20	115	106.5
	0.125	54	313.656	20	113	106.2
	0.093	32	419.547	20	89	101.4
	0.125	40	275.390	20	110	105.6
	0.157	28	1703.9	22	143	122.0
10	0.188	52	5351.2	22	143	122.0
	0.125	24	4410.2	22	116	117.4
	0.171	52	3898.8	22	132	120.2
	0.156	50	2238.3	22	143	122.0

TABLE II

Comparison of Heuristic Methods with BIP based Method Over 30 Simulations

Δ	Top-down	Bottom-up	Combined	SCP
0	5	13	15	0
1	13	12	13	0
2	10	5	2	1
≥ 3	2	0	0	29