Centrum voor Wiskunde en Informatica

REPORTRAPPORT

MAS

Modelling, Analysis and Simulation

Modelling, Analysis and Simulation

Approximating the minimal-cost sensor-selection for discrete-event systems

K.R. Rohloff, J.H. van Schuppen

REPORT MAS-R0404 DECEMBER 2004

CWI is the National Research Institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organization for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

Approximating the minimal-cost sensor-selection for discrete-event systems

ABSTRACT

This paper discusses the approximation of solutions to several NP-complete optimization problems related to the supervisory control of discrete-event systems. Approximation calculations for the minimal-cost sensor-selection problem in a partial observation, centralized control setting is first discussed. It is shown that approximate solutions to this problem cannot always be calculated with a given degree of accuracy in polynomial time. An efficient construction method is shown to convert this sensor selection problem into a novel type of graph cutting problem. Several heuristic algorithms are then shown to approximate solutions to this problem. Approximation methods for computationally difficult communicating decentralized controller problems and actuator selection problems are also discussed. It is shown how to convert these problems into graph cutting problems.

2000 Mathematics Subject Classification: 93C65, 90C27.

Keywords and Phrases: Control, sensor selection, discrete-event systems, combinatorial optimization, minimal cut set.

Note: This work was carried out under project MAS2 - CC and financially sponsored in part by the European Commission via project EU.IST.CC (IST-2001-33520).

Approximating the Minimal-Cost Sensor-Selection for Discrete-Event Systems

Kurt R. Rohloff

UIUC — Coordinated Science Laboratory,
The University of Illinois at Urbana-Champaign
1308 West Main Street, Urbana, IL 61801, U.S.A.
krohloff@uiuc.edu

Jan H. van Schuppen

CWI — Centrum voor Wiskunde en Informatica
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
J.H.van.Schuppen@cwi.nl

This paper discusses the approximation of solutions to several NP-complete optimization problems related to the supervisory control of discrete-event systems. Approximation calculations for the minimal-cost sensor-selection problem in a partial observation, centralized control setting is first discussed. It is shown that approximate solutions to this problem cannot always be calculated with a given degree of accuracy in polynomial time. An efficient construction method is shown to convert this sensor selection problem into a novel type of graph cutting problem. Several heuristic algorithms are then shown to approximate solutions to this problem. Approximation methods for computationally difficult communicating decentralized controller problems and actuator selection problems are also discussed. It is shown how to convert these problems into graph cutting problems.

2000 Mathematics Subject Classification: 93C65 90C27.

Keywords and Phrases: Control, sensor selection, discrete-event systems, combinatorial optimization, minimal cut set.

1 Introduction

When synthesizing a controller for a system to achieve a specification, a control engineer may commonly have a choice of sensors for the controller to use. The controller would naturally need a sufficient set of sensors to satisfy the specification, but due to reasons of economy the engineer may want to minimize the cost of purchasing and installing the sensors. This prompts an important problem in control theory that is the focus of discussion in this paper: the minimum-cost sensor-selection problem. The sensor selection problem is closely related to two other types of important control problems also discussed in this paper: the actuator selection problem, and in the case of communicating decentralized control, the communication minimization problem. For the actuator selection problem the task is to find the minimum-cost amount of control actuation required for a controller to achieve a specification. For the communication minimization problem in the setting of decentralized control, decentralized controllers operating on a system may observe different parts of system behavior and may not be able to control a system to achieve a specification. However, if the controllers were to pool their observations of system behavior, the specification may be achievable. The communication minimization problem is to find the minimal cardinality set of events whose observation occurrences the controllers should to share in order to achieve some specification

These cost minimization problems are discussed using the framework of supervisory control theory and discrete-event systems introduced in the seminal works [13, 14]. In this control setting, a controller may have sufficient actuation to disable some events but not others. Similarly, a controller may not be able to observe all system events. The framework presented in [11] is used to model a controller's observation behavior where system events are partitioned into events whose occurrence is always observed by the supervisor and events that are never observed by the supervisor. This framework is justified in the context of the problem of sensor selection in that it is assumed the sensors are deterministic and report all occurrences of the events they are designated to sense.

Variations of the sensor selection problem using frameworks similar to the one used in this paper have been investigated in [3, 5, 7, 24, 26]. The problem of designing an observation function that is as coarse as possible is discussed in [3]. A projection mapping is assumed in [3] that is different from the natural projection operation used as the observation function in this paper, and optimization and approximation methods are not discussed in [3]. The optimization of the observable event set is discussed in [5] for achieving both observability and normality for a problem setting very similar to that discussed here. An exponential-time algorithm is shown in [5] for giving an optimal observable set. An algorithm is given in [26] for optimizing the sensor selection set in exponential time along with a polynomial time algorithm for finding exactly one locally minimum sensor selection. It is shown in [24] that the decision problem version of the sensor selection problem investigated here is NP-complete, so most likely polynomial time methods to solve this problem cannot be developed. An optimal sensor selection problem is also discussed in [8], except the observation function is different from the one assumed in this paper.

Variations of the actuator selection problem have been discussed in the supervisory controls literature. In [18, 19], cost is put on the actions of the controller and behavior of the system. A related problem is also discussed in [26].

The decentralized control systems in this paper are discussed in the framework of [11, 17] where controllers control and observe possibly different subsets of system behavior. The two-way communication minimization problem discussed here is a special case of the open problem presented in [21] for the setting of decentralized control. It is hypothesized in [21] that more general versions of the communication minimization problem are undecidable, so it is worthwhile to investigating solutions to simpler versions of the problem. The communication minimization problem in the setting discussed here is shown to be NP-complete.

Fortunately, despite the computational difficulty of many of these problems, the absolute minimum-cost solutions to the problems may not always be necessary for many practical applications. Approximate solutions for the minimization problems might be found in a more time-efficient manner, and they may commonly be sufficient for practical use. Therefore, an interesting compromise to finding the minimum-cost solution would be to develop methods to approximate the minimal sensor selection and hopefully put some bounds on the closeness of the approximations found this way as many NP-complete problems have accurate polynomial-time approximation methods. However, there has been little investigation into the computation of approximate solutions to many computationally difficult supervisory control problems.

Unfortunately, one of the main results of this paper is that the approximation of minimal-cost sensor selections for supervisory control is computation difficult and can most likely not be done in polynomial time with reasonable accuracy. However, it is shown how to convert the sensor cost minimization problem into a type of directed graph minimum-cost st-cut that has not previously been investigated in the literature. This alternative formulation of the selection problems allows the more intuitive development of approximation methods. It is also shown how to convert these graph cutting problems into an integer programming problem. Several heuristic polynomial time approximation methods are shown that are based on this problem conversion. It is also shown that the actuator selection problem and communication selection problem are computationally similar to the sensor selection problem. Therefore, the heuristic approximation methods can also be used with these other problems.

The next section of this paper formally introduces the system framework and notation used in this paper's investigations of the cost minimization problems. Section 3 introduces the sensor selection problem and Section 4 discusses the approximation complexity of this problem. Section 5 discusses randomized methods for solving the sensor selection approximation problem, and Section 6 shows the conversion of the sensor selection problem into an edge-colored directed graph st-cut problem. Section 7 analyzes a deterministic greedy and a randomized method for finding approximate solutions to the sensor selection problem. Section 8 shows how to convert the graph cutting problem into an integer programming problem. Section 9 discusses the decentralized control communication minimization problem, and Section 10 discusses the actuator cost minimization problem. Section 12 closes the paper with a discussion of the shown results.

2 Notational Review

To aid the reader, this section reviews the notation of supervisory control theory. First, the discrete-event system automata models of [13, 14] are presented and then the supervisory control model in the decentralized setting of [10] is introduced. This section closes with a discussion of computational approximation.

The basic automaton system model G used in this paper is a 5-tuple $(X^G, x_o^G, \Sigma^G, \delta^G, X_m^G)$ where X^G is the set of states, x_o^G is the initial state, Σ^G is the automaton alphabet, $\delta^G : X^G \times \Sigma^G \to X^G$ is the (possibly partial) state transition function, and X_m^G is the set of "marked" states. Deterministic systems and specification automata are exclusively used in this paper as comparisons on nondeterministic automata are generally PSPACE-complete [4].

The language *generated* by G is the set of strings $\mathcal{L}(G) = \{s \in \Sigma^{G*} | \delta^G(x_0^G, s)!\}$ that are defined in G from the initial state. Note that the unary operator ! for $f(\alpha)!$ returns true if $f(\cdot)$ is defined for input α, false otherwise. The language *marked* by an automaton G is the set of strings that lead to a marked state from the initial state. That is, $\mathcal{L}_m(G) = \{s \in \Sigma^{G*} | \delta^G(x_0^G, s) \in X_m^G\}$. The language generated $\mathcal{L}(G)$ is a prefix-closed language, i.e., it contains all the prefixes of all its strings. The marked language $\mathcal{L}_m(G)$ is not prefix-closed in general. For a language K, $\overline{K}$ denotes the set of all the prefixes of all the strings in K. An automaton that accepts a prefix-closed language is called a *prefix-closed automaton.* An automaton is said to be *nonblocking* if the prefix-closure of its marked language is equal to its generated language, i.e., $\overline{\mathcal{L}_m(G)} = \mathcal{L}(G)$.

2.1 Supervisory Control

Following the modelling formalisms of [13, 23], systems are modelled as finite-state automata with external controllers. Control actions are enforced by selectively disabling controllable events. Controllers can be realized as finite-state automata that observe some events and control a potentially different set of events. Controllers should not be able to disable uncontrollable events and control actions should not update on the occurrence of locally unobservable events in the absence of external communication.

Given a centralized controller S and a system G, the composed system of S controlling G is denoted as the controlled system S/G. Controller S is said to be nonblocking for system G if S/G is nonblocking, i.e., if $\overline{\mathcal{L}_m(S/G)} = \mathcal{L}(S/G)$.

As stated above, a controller may only observe a subset of the system events $\Sigma_o \subseteq \Sigma$. In this discussion a natural projection operation $P : \Sigma \to \Sigma_o$ is used to model a controller's observations of system behavior. For the empty event ϵ, $P(\epsilon) = \epsilon$, and for a string of events s and an event σ, $P(s\sigma) = P(s)\sigma$ if $\sigma \in \Sigma_o$ and $P(s\sigma) = P(s)$ otherwise. The inverse function $P^{-1} : \Sigma_o^* \to 2^{\Sigma^*}$ is defined such that $P^{-1}(s)$ is the set of strings with s as their common projection. As system behavior progresses and a string of events s is generated by the system, a controller would observe $P(s)$. The controller would then use observation projection $P(s)$ to estimate the current system state and determine its control action. A controller is said to be *admissible* if it only attempts to disable controllable events and updates its control action on the occurrence of observable events.

For the case of multiple controllers (i.e., standard decentralized control) in the framework of [10] an event is disabled if it is disabled by at least one controller. (Hence the actions local controllers are combined globally by a "fusion by intersection" policy.) See Figure 1 for a schematic of a system with a set decentralized controllers.

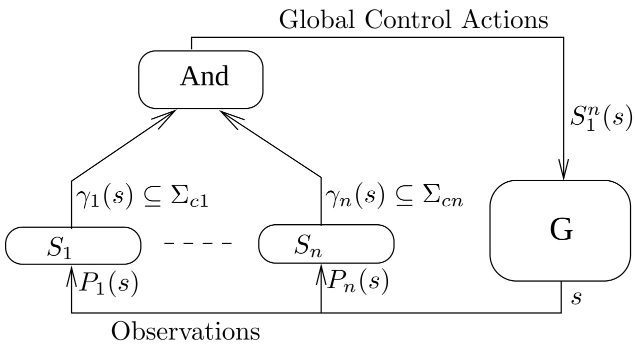

Figure 1: Schematic of a conjunctive decentralized control system

In Figure 1, the monolithic system G is controlled by n controllers, $\{S_1, \dots, S_n\}$. The controllers $\{S_1, \dots, S_n\}$ observe system events $\{\Sigma_{o1}, \dots, \Sigma_{on}\}$ and control events $\{\Sigma_{c1}, \dots, \Sigma_{cn}\}$ respectively. For every controller i there is also a defined local observation projection $P_i : \Sigma \to \Sigma_{oi}$. Decentralized controller S_i makes observations of the system behavior s and generates a control action $\gamma_i(s)$. The local control actions $\{\gamma_1(s), \dots, \gamma_n(s)\}$ are combined globally using an intersection operation to form the global control action that is enforced on the system G. Therefore, for this decentralized control system, if an event is locally disabled by a controller S_i, then it is globally disabled due to the global intersections of control actions. Given a decentralized controller system $\{S_1, \dots, S_n\}$ and a system G, $\{S_1, \dots, S_n\}$ controlling G is denoted as $S_1 \wedge \cdots \wedge S_n/G$. Also in this setting let $\Sigma_c = \cup_{i=1}^n \Sigma_{ci}$ and $\Sigma_{uc} = \Sigma \setminus \Sigma_c$.

Three important properties related to decentralized controller existence are controllability, co-observability and $\mathcal{L}_m(G)$-closure.

Definition 1 *[13] Consider the languages K and M such that $M = \overline{M}$ and the set of uncontrollable events Σ_{uc}. The language K is* controllable *with respect to M and Σ_{uc} if $\overline{K}\Sigma_{uc} \cap M \subseteq \overline{K}$.*

Definition 2 *Consider the sets of languages K and M. The set K is M-*closed *if $K = \overline{K} \cap M$.*

Definition 3 *[17] Consider the sets of languages K and M such that $M = \overline{M}$ and the sets of locally controllable, Σ_{ci}, and observable Σ_{oi} events such that $i \in \{1, ..., s\}$. The set of languages K is* co-observable *with respect to M, P_i and Σ_{ci}, $i \in \{1, ..., s\}$ if for all $t \in \overline{K}$ and for all $\sigma \in \Sigma_c$,*

$$\left(t\sigma \notin \overline{K}\right) \text{ and } (t\sigma \in M) \Rightarrow$$
$$\exists i \in \{1, ..., s\} \text{ such that } P_i^{-1}\left[P_i(t)\right]\sigma \cap \overline{K} = \emptyset \text{ and } \sigma \in \Sigma_{ci}.$$

The concept of co-observability captures the notion that one supervisor always knows to disable an event when needed. In the case of centralized control as discussed in [11], co-observability is called observability. The above definitions on controllability, M-closure and co-observability are central in the following controller existence theorem called the controllability and co-observability theorem.

Theorem 1 *[17] For a finite-state automaton system G and a finite-state automaton specification H such that $\mathcal{L}_m(H) \subseteq \mathcal{L}_m(G)$, sets of controllable events $\{\Sigma_{c1}, ..., \Sigma_{cs}\}$ and sets of observable events $\{\Sigma_{o1}, ..., \Sigma_{os}\}$ there exists a set of partial observation controllers $\{S_1, S_2, ...S_s\}$ such that $\mathcal{L}_m(S_1 \wedge \cdots \wedge S_s/G) = \mathcal{L}_m(H)$ and $\mathcal{L}(S_1 \wedge \cdots \wedge S_s/G) = \overline{\mathcal{L}_m(H)}$ if and only if the following three conditions hold:*

1. *$\mathcal{L}_m(H)$ is controllable with respect to $\mathcal{L}(G)$ and Σ_{uc}.*
2. *$\mathcal{L}_m(H)$ is co-observable with respect to $\mathcal{L}(G)$, Σ_{o1},...,Σ_{os} and Σ_{c1},...,Σ_{cs}.*
3. *$\mathcal{L}_m(H)$ is $\mathcal{L}_m(G)$-closed.*

For languages generated by deterministic automata, controllability and $\mathcal{L}_m(G)$-closure can be decided in polynomial time using standard automata manipulation operations. There is a construction presented in [16] for deciding the co-observability of languages generated by deterministic automata. A modified version of this construction is used several times later in this paper. The essence of this method is that a automaton $\mathcal{M}$ is constructed from a system G, a specification automata H, sets of controllable events $\Sigma_{c1}, \ldots, \Sigma_{cs}$ and sets of observable events $\Sigma_{o1}, \ldots, \Sigma_{os}$, such that $\mathcal{L}_m(\mathcal{M}) = \emptyset$ if and only if the $\mathcal{L}_m(H)$ is co-observable with respect to $\mathcal{L}(G)$, $\Sigma_{o1}, \ldots, \Sigma_{os}$ and $\Sigma_{c1}, \ldots, \Sigma_{cs}$.

The construction of $\mathcal{M}$ takes polynomial time if the number of controllers is bounded. Therefore, for deterministic monolithic system problems with a bounded number of controllers, controller existence can be decided in polynomial time.

A less restrictive version of Theorem 1 also holds for the case of generated language specifications (and hence prefix-closed specifications) where the $\mathcal{L}_m(G)$-closure condition is disregarded. Also, in the case of centralized control (i.e., one controller), co-observability is called observability. In this paper the terminology of control theory is used even though it may be counter to naming conventions currently used in computer science theory. (That is, co-observability is not non-observability.)

2.2 Background on Approximation Theory

In the field of computation theory a problem instance is said to be a "decision problem" if all problem instances are mapped to be either "true" or "false". In the set of decision problems, a problem is said to be in class P if it can be decided in polynomial time using a *deterministic* computation device and it is said to be in NP if it can be decided in polynomial time using a *nondeterministic* computation device. Although it is not known for sure, it is generally believed that the class NP is distinct from the class P. Therefore, it is believed that not all decision problems can be solved efficiently in time.

Similar to decision problems there is a set of problems called optimization problems where a specific optimization problem has a set of problem instances $\mathcal{P}$, a set of feasible solutions $\mathcal{S}_p$ for a problem instance $p \in \mathcal{P}$ and a cost function $cost_p : \mathcal{S}_p \rightarrow \Re$ that maps the set of feasible solutions of a problem instance to a real value that is a measure of the desirability of that solution. The solution to an instance of an optimization problem is the minimal-cost solution for that problem instance, i.e., $S(p) \in \mathcal{S}_p$ such that $\forall s \in \mathcal{S}_p, cost_p(S(p)) \leq cost_p(s)$.

The set of feasible solutions to the optimization problem may be finite, countably infinite or a subset of the real numbers. Similar to decision problems, there are the PO and NPO optimization problem classes where an optimization problem is said to be in PO if an optimal solution can be calculated in polynomial time using a *deterministic* computation device and it is said to be in NPO if an optimal solution can be computed in polynomial time using a *nondeterministic* computation device.

There are several important optimization problems in NPO that are believed to not be in PO. Please see the compendium in [1] for an extensive listing of problems such as these. However, even though some optimization problems probably cannot be solved in polynomial time, it may still be possible to calculate accurate approximate solutions to these problems efficiently in time. Naturally, the solutions to some problems may be more difficult to approximate than the solutions for other problems. The concept of an *r-approximation*, shown in Definition 4 below captures this property. First, however, some necessary notation needs to be introduced.

Definition 4 r-approximate algorithm: *[1] For a problem instance $p \in \mathcal{P}$ of an optimization problem let $\mathcal{A}$ be an algorithm that such that when given p as input, $\mathcal{A}$ returns an approximate solution $\mathcal{A}(p) \in \mathcal{S}_p$ for that problem instance. The approximation algorithm $\mathcal{A}(p)$ is r-approximate if*

$$\forall p \in \mathcal{P} \left(\frac{cost(\mathcal{A}(p))}{cost(S(p))} \leq r \right). \tag{1}$$

Conceptually, the r in an r-approximation is the maximum ratio between the cost of the optimal solution $cost(S(p))$ and the approximation found by an algorithm, $\mathcal{A}(x)$. The ratio r may be some function on the size of the problem instance. An important problem class in the hierarchy of approximation problems is the APX problem class formally defined below.

Definition 5 The APX Problem Class: *[1] APX is the class of all NP-complete optimization problems such that there is a polynomial time r-approximate for a constant $r \in \Re, r \geq 1$,*

3 The Sensor Selection Problem

In the framework of supervisory control, a set $\Sigma_o \subseteq \Sigma$ is called a *sufficient sensor selection* with respect to G, H and Σ_c if $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c. A sufficient sensor selection can also be called an observability set. Therefore, if Σ_o is a sufficient sensor selection and $\mathcal{L}(H)$ is controllable with respect to $\mathcal{L}(G)$ and Σ_c, then there exists an admissible controller S such that $\mathcal{L}(S/G) = \mathcal{L}(H)$.

It may be possible that there is a positive cost $cost : \Sigma \to \Re^+$ associated with observing an event. Due to economic reasons the total cost of sensors used by a controller may want to be kept to a minimum as long as the sensor selection is sufficient. The problem of finding a minimal-cost sufficient sensor selection is called the *minimal-cost sensor-selection problem.* For a set of events $\Sigma_o \subseteq \Sigma$ define $cost(\Sigma_o) = \Sigma_{\sigma \in \Sigma_o} cost(\sigma)$.

Problem 1 Minimal-Cost Sensor-Selection: *Given G, H and $\Sigma_c \subseteq \Sigma$, find a sufficient sensor selection Σ_o^{min} such that for any other sufficient sensor selection Σ_o, $cost(\Sigma_o^{min}) \leq cost(\Sigma_o)$.*

There is a special case of Problem 1 where the sensors have uniform cost. Therefore, the cost minimization problem becomes a cardinality minimization problem.

Problem 2 Minimal-Cardinality Sensor-Selection: *Given G, H and $\Sigma_c \subseteq \Sigma$, find a sufficient sensor selection Σ_o^{min} such that for any other sufficient sensor selection Σ_o, $|\Sigma_o^{min}| \leq |\Sigma_o|$.*

Because of the NP-completeness of Problem 2 the minimal-cardinality sensor-selection can not always be found in a computationally efficient manner [25]. This also shows that Problem 1 is similarly computationally difficult. However, a sufficient sensor selection Σ_o may still need to be found reasonably efficiently such that the cost of this sensor selection $cost(\Sigma_o)$ is as close to the minimal-cost sensor-selection $cost(\Sigma_o^{min})$ as possible. Fortunately, as mentioned above, some NP-complete minimization problems have fairly accurate polynomial time approximation algorithms [1, 22]. This means sufficient and approximate solutions can be found for many computationally difficult problems in a reasonable amount of time. However, not all NP-complete minimization problems are believed to have this property [1, 22]. This prompts the investigations into the approximation difficulty of minimal-cost sensor-selection problem in the following subsection.

4 The Complexity of Minimal-Cost Sensor-Selection Approximations

It is now shown that the minimal-cardinality sensor-selection problem is not in APX using a minimal set cover problem reduction. This implies that the minimal-cost sensor-selection problem is also not in APX.

The minimal set covering problem is a fundamental problem in computer science used to show the computational difficulty of many other problems. For this problem a set $S = \{\gamma_1, \dots, \gamma_n\}$ is given along with a set of subsets $\mathcal{C} = \{C_1, \dots, C_m\} \subseteq 2^S$ and the problem is to find a set covering $\mathcal{C}_{min} = \{C_{i_1}, \dots, C_{i_m}\} \subseteq \mathcal{C}$ such that $C_{i_1} \cup \cdots \cup C_{i_m} = S$ and for any other covering subset $\mathcal{C}' = \{C_{k_1}, \dots, C_{k_l}\} \subseteq \mathcal{C}$ such that $C_{k_1} \cup \cdots \cup C_{k_l} = S$, $|\mathcal{C}_{min}| \leq |\mathcal{C}'|$. The set $\mathcal{C}_{min}$ is called the minimal set covering. It is known that the minimal set covering problem is NP-complete [4] and not in APX [1]. This result can be used to show that minimal-cardinality sensor-selection problem is also not in APX.

Theorem 2 *The minimal-cardinality sensor-selection problem is not in APX.*

Proof: This theorem is demonstrated using a proof by contradiction. Suppose the minimal-cardinality sensor-selection problem is in APX. Then there is an algorithm $\mathcal{A}_o$ that when given an instance of the minimal-cardinality sensor-selection problem, returns an approximation of the minimal-cardinality sensor-selection whose cardinality is within a constant ratio r of the cardinality of the minimal-cardinality sensor-selection in polynomial time. It is now shown how algorithm $\mathcal{A}_o$ can be used to construct an algorithm $\mathcal{A}_{sc}$ that when given an instance of the set cover problem, returns an approximation of the minimal set cover whose cardinality is within a constant ratio r of the cardinality of the minimal set cover in polynomial time.

Given an instance of the set cover problem, i.e., a set $S = \{\gamma_1, \ldots, \gamma_n\}$ and a set of subsets $\mathcal{C} = \{C_1, \ldots, C_m\} \subseteq 2^S$, assume without loss of generality that $C_1 \cup \cdots \cup C_m = S$. Put an arbitrary ordering on the subsets of S such that $C_1 < \ldots < C_m$. For an element γ_i let $\mathcal{C}_i = \{C_1^i, \ldots, C_{j_i}^i\}$ represent the subsets that contain γ_i. That is, $\forall C_k^i \in \mathcal{C}_i, \gamma_i \in C_k^i$. Furthermore, assume that $C_1^i \leq \ldots \leq C_{j_i}^i$. Now, for the sets $\{\mathcal{C}_1, \ldots, \mathcal{C}_n\}$ construct the automaton G seen in Figure 2 where the ith branch of the initial state represents the ordered list of sets that contain γ_i and α is a symbol not already used.

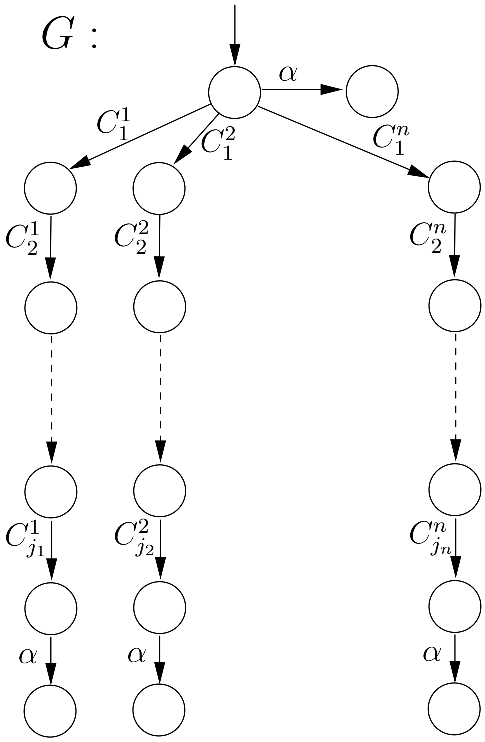

Figure 2: G-automaton used in proof of Theorem 2.

The G automaton can be constructed in polynomial time with respect to the size of the encoding of the set cover problem, and therefore G also has a polynomial number of states. Note that the automaton G may be nondeterministic, but it can be converted to a deterministic automaton accepting the same language by iteratively merging state transitions with the same label at the same parent node until the automaton is deterministic. That is, if $\delta^G(x_1, C_j^k) = x_2$ and $\delta^G(x_1, C_j^l) = x_3$ such that $C_j^k = C_j^l$, then merge the states x_2 and x_3 and remove the $\delta^G(x_1, C_j^l) = x_3$ transition. Because the number of states is bounded by a polynomial, this determinization procedure will halt in a polynomial amount of time.

Let H be a copy of automaton G with all transitions labelled by α removed except for the α transition at the initial state. Let G be the system automaton, let H be the specification automaton and let $\Sigma_c = \{\alpha\}$.

Suppose there exists a string of events $C^i_1 C^i_2 \dots C^i_{j_i}$ such that no events in this string are observed. Then the system is unobservable because a controller would not know to disable the α event after $C^i_1 C^i_2 \dots C^i_{j_i}$ occurred. Hence, if $\mathcal{L}(H)$ is not observable with respect to $\mathcal{L}(G)$, Σ_c and Σ_o where $\Sigma_o = \{C^1_o, \dots, C^p_o\}$, then there exists some event γ_i such that for all $C^q_o \in \Sigma_o$, $\gamma_i \notin C^q_o$. Then $\{C^1_o, \dots, C^p_o\} = \mathcal{C}$ does not form set cover for S.

Similarly, if every string of events $C^i_1 C^i_2 \dots C^i_{j_i}$ contains at least one event that is observed, then a controller would always know to disable α after the string of events $C^i_1 C^i_2 \dots C^i_{j_i}$ occurs. Therefore, for every event γ_i, there must be an event in $C^i_1 C^i_2 \dots C^i_{j_i}$ that is observable. Hence, if $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_c and Σ_o where $\Sigma_o = \{C^1_o, \dots, C^p_o\}$, then $\{C^1_o, \dots, C^p_o\} = \mathcal{C}'$ forms a set cover for S. This is because for any γ_i, there exists a C^q_o such that $\gamma_i \in C^q_o$.

Then, for the given construction of G and H, a set of event $\Sigma_o \in 2^{\Sigma}$ is a sufficient sensor selection with respect to G, H and Σ_c if and only if the corresponding set $\mathcal{C}'$ is a set cover for S. Furthermore, the cardinality of the minimal sensor selection is equal to the cardinality of the corresponding minimal set cover. Therefore, $|\Sigma^{min}_o| = |\mathcal{C}_{min}|$.

Suppose algorithm $\mathcal{A}_o$ is run with the construction of G, H and Σ_c and the observability set Σ'_o is returned. It is known that $|\Sigma'_o|/|\Sigma^{min}_o| \leq r$ because of the assumption on $\mathcal{A}_o$. The set Σ'_o can then be used to calculate a set $\mathcal{C}'$ using the construction above such that $|\mathcal{C}'|/|\mathcal{C}_{min}| \leq r$. The problem instance G, H and Σ_c can be constructed in polynomial time and it was assumed the algorithm $\mathcal{A}_o$ can be run in polynomial time. This implies there exists a polynomial time algorithm to find an approximation to the minimal set cover such that the ratio of the cardinality of the approximation to the cardinality of the minimal set cover is bound by a constant r. This implies by definition that the minimal set cover problem is in APX which forms a contradiction. Therefore, there does not exist an algorithm $\mathcal{A}_o$ that when given an instance of the minimal-cardinality sensor-selection problem, returns an approximation of the minimal-cardinality sensor-selection whose cardinality is within a constant ratio r of the cardinality of the minimal-cardinality sensor-selection in polynomial time unless P=NP. Finally, the minimal-cardinality sensor-selection problem is not in APX.

■

Theorem 2 can be easily extended to the non-uniform cost case.

Corollary 1 *The minimal-cost sensor-selection problem is not in APX.*

Theorem 2 and Corollary 1 show that sensor selection problems are difficult to approximate in a time efficient manner. It was also shown in [9] that the sensor selection problem admits no $2^{\log^{(1-\epsilon)} n}$ approximation for any $\epsilon > 0$ unless $NP \subseteq DTIME(n^{\text{polylog } n})$. It follows from this result that if solutions to the sensor selection problems can be found with better than a $2^{\log^{(1-\epsilon)} n}$-approximation, then a method has been found for solving NP-complete problems in quasi-polynomial time. This lower bound on the ability to approximate minimal sensor selections is generally considered to be a very poor lower bound in the computer science community because as ϵ approaches 0, then $2^{\log^{(1-\epsilon)} n}$ approaches n. However, because of the fundamental importance of this problem, usable methods need to be developed to approximate the minimal-cost sensor-selections. This prompts the algorithms in the later sections of this paper for approximating solutions to sensor selection problems.

5 A Randomized Descent Approximation Algorithm

A randomized descent algorithm for approximating minimal-cost sensor-selections is now shown. Consider the set of system events Σ and its power set 2^{Σ}. The process of finding the minimum-cost sufficient sensor selection $\Sigma_o^{min} \subseteq \Sigma$ is effectively a search over the power set of Σ, 2^{Σ}. An interesting property of observable systems is that for any set of observable events $\Sigma_o \subseteq \Sigma$ such that $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c, then for any Σ_o' such that $\Sigma_o \subseteq \Sigma_o' \subseteq \Sigma$, $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o' and Σ_c and $cost(\Sigma_o) \leq cost(\Sigma_o')$. That is, all supersets of sufficient sensor selections are also sufficient sensor selections.

Given a set of events Σ_o such that $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c, it may be possible that there does not exist an event $\sigma \in \Sigma_o$ such that $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, $\Sigma_o \setminus \{\sigma\}$ and Σ_c. In this case Σ_o is called a locally minimal sufficient sensor selection. For a given system there may possibly be many locally minimal sufficient sensor selections, but these locally minimal sufficient sensor selections may not all be minimal-cost sufficient sensor selections. Consider the following example.

Example 1 *Suppose* $\mathcal{L}(G) = \overline{\{\alpha, \lambda\beta\alpha, \lambda\gamma\alpha\}}$, $\mathcal{L}(H) = \overline{\{\alpha, \lambda\beta, \lambda\gamma\}}$ *and* $\Sigma_c = \{\alpha\}$. *Assume the events have uniform cost. The set* $\Sigma_o = \{\beta, \gamma\}$ *is a sufficient sensor selection and it is locally minimal, but the minimal-cardinality sensor-selection is* $\Sigma_o = \{\alpha\}$.

Consider how the power set 2^{Σ} forms a lattice with respect to the partial ordering of the subsets of Σ. It is assumed that $\mathcal{L}(H)$ is never observable with respect to $\mathcal{L}(G)$, $\emptyset$ and Σ_c (i.e. the trivial case). Let $n = |\Sigma|$. Therefore, for every path $\Sigma\Sigma_1\Sigma_2\cdots\Sigma_n$ on the lattice formed by 2^{Σ} such that $\Sigma \supseteq \Sigma_1 \supseteq \cdots \supseteq \Sigma_n \supseteq \emptyset$, there is a boundary observability set Σ_i such that for any $\Sigma_i' \supseteq \Sigma_i$, $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_i' and Σ_c and for any $\Sigma_{i+1}' \subseteq \Sigma_{i+1}$, $\mathcal{L}(H)$ is not observable with respect to $\mathcal{L}(G)$, Σ_{i+1}' and Σ_c.

Therefore, for the sets of all paths $\Sigma \supseteq \Sigma_1 \supseteq \cdots \supseteq \Sigma_n \supseteq \emptyset$ in the lattice formed by 2^{Σ}, the set of boundary sets of these paths forms a frontier between sufficient sensor selections and sets of observable events that would make the system unobservable. The minimum-cost observability set is somewhere on this boundary. Note that not all members of this set of boundaries are locally minimal sensor selections.

Finding a locally minimal sensor selection for a system is fairly easy. This could be done by initializing the set of observable events to be Σ, and events could iteratively be removed from the set of observable events as long as the specification is still observable. This is exactly what is done in the following randomized algorithm.

Algorithm 1 Randomized Local Minima Search Algorithm (RanLocMin):

Input: G, H, Σ_c.
  $\Sigma_{test} \Leftarrow \Sigma$;
  $\Sigma_o \Leftarrow \Sigma$;
  Repeat:
   {
    Randomly remove $\sigma \in \Sigma_{test}$ *from* Σ_{test}.
    If $\mathcal{L}(G)$ *is not observable w.r.t.* $\mathcal{L}(H)$, Σ_c *and* $\Sigma_o \setminus \{\sigma\}$, *then:*
     {
      $\Sigma_o \Leftarrow \Sigma_o \setminus \{\sigma\}$;
     }
   }
  Until $\Sigma_{test} = \emptyset$.
Return: Σ_o.

During the operation of Algorithm 1, events are iteratively tested to be removed from the set of observable events. The set Σ_{test} is the set of events which have not been tested for removal from the set of observable events. Note that the algorithm is greedy in that once an event is removed from the set of observable events, it is never returned to that set. Consequently, if an event σ is tested for removal from Σ_o and it is found that $\mathcal{L}(G)$ is observable with respect to $\mathcal{L}(H)$, Σ_c and $\Sigma_o \setminus \{\sigma\}$, then σ can be removed from Σ_o and never needs to be tested for removal again. On the other hand, if at any iteration of the algorithm an event σ is tested for removal and it is found that $\mathcal{L}(G)$ is not observable with respect to $\mathcal{L}(H)$, Σ_c and $\Sigma_o \setminus \{\sigma\}$, then any subset $\Sigma_o' \subseteq \Sigma_o \setminus \{\sigma\}$, $\mathcal{L}(G)$ would not observable with respect to $\mathcal{L}(H)$, Σ_c and Σ_o'. Consequently, such an event σ, removed from Σ_{test} but retained in Σ_o should never need to be tested again for removal from Σ_o. By removing events iteratively from Σ_{test} as they are tested for removal from Σ_o it is ensured that all events are tested exactly once for removal from Σ_o. This guarantees that the final set of observable events Σ_o returned by Algorithm 1 is a local minimum.

Suppose Algorithm 1 finds the global minimum-cost observability set with probability p, which may be quite low, but it is desired to find the global minimum with probability $r \in (p, 1)$. The probability of finding the global minimum using Algorithm 1 could be boosted through iteration as in Algorithm 2 below.

Algorithm 2 Iterated Randomized Local Minima Search Algorithm (ItRanLocMin):

Input: G, H, Σ_c.

$\Sigma_o \Leftarrow \Sigma$

Repeat k *times:*

{

$\Sigma_f \Leftarrow RanLocMin(G, H, \Sigma_c)$

if $cost(\Sigma_f) \leq cost(\Sigma_o)$, *then* $\Sigma_o \Leftarrow \Sigma_f$

}

Return: Σ_o.

Algorithm 2 makes k calls to Algorithm 1 and hence takes k times as long as Algorithm 1. This prompts the question of what value of k should be chosen such that a global minima is found with probability at least r using Algorithm 2? It would be helpful to have k as small as possible such that a minimal-cost sensor-selection is found with high probability.

If Algorithm 1 finds a global minimum with probability p, this algorithm does not find a global minimum with probability $(1 - p)$. Hence, over the k trials of Algorithm 2, the probability that a global minimum is not found is $(1 - p)^k$. It is well known that $(1 - p)^k \leq \exp(-pk)$ based on the convex analysis that $\exp(-x) - 1 + x \geq 0$ $\forall x \in [0, \inf)$. So, if it is desired that the iterated randomized local minima search algorithm returns a non-global minimum with probability at least $(1 - p)^k \leq (1 - r)$, then k needs to be found such that $\exp(-pk) = (1 - r)$, or $k = \frac{\ln\left(\frac{1}{(1-r)}\right)}{p}$.

Unfortunately, p may be very small in the worst case. Consider the case when $\Sigma = n$. It is assumed for the sake of discussion that n is even and that the events have uniform cost of being observed. The lattice of potential observability sets formed by 2^Σ could have a frontier containing at most $\binom{n}{n/2}$ local minima. This corresponds to a level frontier on the lattice such that for all sets Σ_a such $|\Sigma_a| \geq n/2$, Σ_a is an observability set and not otherwise. Now suppose that from this frontier, there is another frontier with exactly one global minimum observability set Σ_o^{min} such that $|\Sigma_o^{min}| = n/2 - 1$ and for all other possible observability sets Σ_a, Σ_a is a local minimum if and only if $|\Sigma_a| = n/2$ and $\Sigma_o^{min} \not\subseteq \Sigma_a$.

With this construction the probability that Σ_o^{min} is found by the randomized local minimum search algorithm is $\frac{n}{\binom{n}{n/2}}$. This is because as the randomized algorithm removes an event from the observability set, there are then $n/2$ events remaining, n possible combinations of n events taken $n/2$ at a time leads to the global minimum, Σ_o^{min}.

Therefore, for n reasonably large:

$$\begin{aligned}
p &= \frac{n}{\binom{n}{n/2}} \\
&= \frac{n}{\left(\frac{n!}{(n/2)!(n/2)!}\right)} \\
&= \frac{n}{n\left(\frac{n-1}{n/2}\cdots\frac{n-i}{(n/2+1)-i}\cdots\frac{n/2+1}{2}\right)} \\
&= \left(\frac{n/2}{n-1}\cdots\frac{(n/2+1)-i}{n-i}\cdots\frac{2}{n/2+1}\right) \\
&\leq (1/2)^{n/2-1}
\end{aligned}$$

Therefore, p is exponential in $-n$ in the worst case and k would therefore have to be exponentially large in order to obtain an arbitrarily high probability of finding a minimum-cost sensor-selection with Algorithm 2. However, worst-case scenarios are rather unusual and this algorithm helps us gain insight into the problem. The more iterations that are taken in Algorithm 2 the closer of an approximation is obtained of the global minimum. In the hypothetical example above with the nearly flat frontier, a very close approximation to the global minimum is obtained after one iteration of the randomized search algorithm.

6 The Graph Cutting Problem

A method is outlined in [20] for testing observability in polynomial time with an $\mathcal{M}$-automaton construction. (However, this construction is not explicitly given in [20]), A modified version of this test is presented here that can be used to convert sensor selection problems into a novel type of graph cutting problem called an edge colored directed-graph *st*-cut problem.

For this edge colored directed-graph *st*-cut problem, assume an edge-colored directed graph $D = (V, A, C)$ where V is a set of vertices, $A \subseteq V \times V$ are directed edges and $C = \{c_1, \ldots, c_p\}$ is the set of colors. Each edge is assigned a color in C and let A_i be the edges having color c_i. Given $I \subseteq C$, let $A_I = \cup_{c_i \in I} A_i$. For two nodes $s, t \in V$ such that there is a path of directed edges from s to t, then I is a colored *st*-cut if $(V, (A \setminus A_I), C)$ has no path from s to t. For each edge color $c_i \in C$, associated a positive cost $cost(c_i) \in \Re^+$ and for a set of colors $I \subseteq C$, define the cost of I, $cost(I)$ to be $\Sigma_{c_i \in I} cost(c_i)$. This prompts us to define the colored cut problem.

Problem 3 Minimal-Cost Colored Cut: *Given an edge colored directed graph $D = (V, A, C)$ and two vertices, $s, t \in V$, find a colored st-cut $I^{min} \subseteq C$ such that for any other colored st-cut $I \subseteq C$, $cost(I^{min}) \leq cost(I)$.*

It is now shown how to convert an instance of a colored cut problem into an instance of a sensor selection problem.

Proposition 1 *An instance of the minimal-cost colored cut problem can be reduced to an instance of the minimal-cost sensor-selection problem in polynomial time.*

Suppose an edge-colored directed graph $D = (V, A, C)$ and two vertices s, t are given. A system G, specification H and controllable event set Σ_c are now constructed from D. For the colors $C = \{c_1, \ldots, c_p\}$, let the event set Σ include a corresponding set of events $\{\sigma_1, \ldots, \sigma_p\}$ such that color c_i is paired with event σ_i. Let γ be another event and define $\Sigma = \{\sigma_1, \ldots, \sigma_p, \gamma\}$. Suppose that for all c_i, σ_i pairs that $cost(c_i) = cost(\sigma_i)$ and let $cost(\gamma)$ be any value in $\Re^+$. Also define $X^G = V \cup \{s', s'', t'\}$ where s', s'', t' are states not in V. Let $x_0^G = s$. To define the state transition function, let v_1, v_2 be any vertices except s. If $(v_1, v_2) \in A_i$, then $\delta^G(v_1, \sigma_i) = v_2$. If $(s, v_2) \in A_i$, then $\delta^G(s, \sigma_i) = v_2$ and $\delta^G(s'', \sigma_i) = v_2$. If $(v_1, s) \in A_i$, then $\delta^G(v_1, \sigma_i) = s''$. For simplicity it is assumed that $(s, s) \notin A$. Also, transitions are added to G such that $\delta^G(s, \gamma) = s'$ and $\delta^G(t, \gamma) = t'$. Let H be a copy of G except that $\delta^H(t, \gamma)$ is undefined. Let $\Sigma_c = \{\gamma\}$.

An example of such a system construction for converting a directed graph D to a system and specification G and H is given in Figure 3.

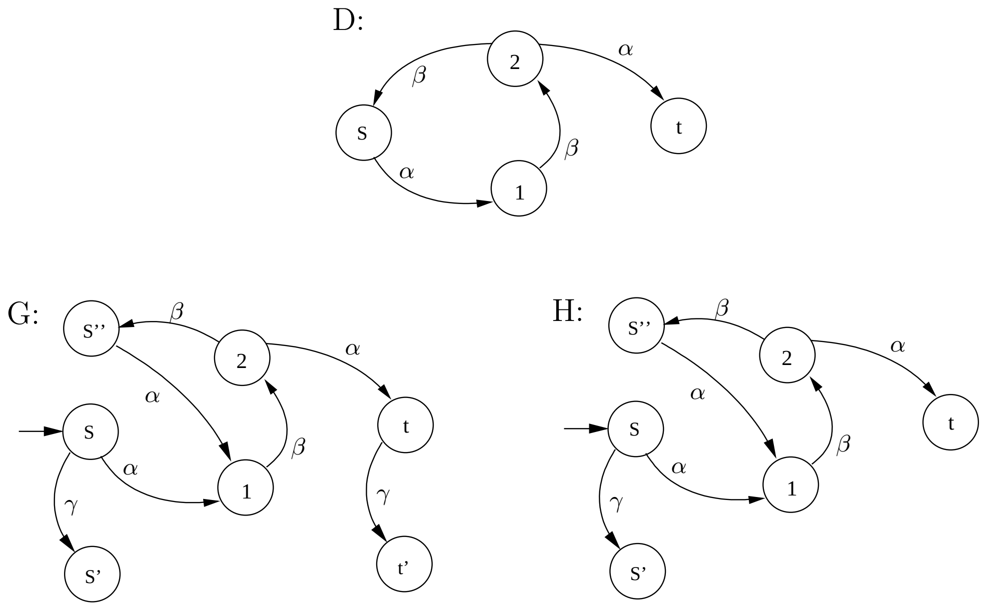

Figure 3: A directed graph D and the systems G and H constructed from it.

In the system constructed above, γ must be enabled at s and be disabled at t. There is a control conflict if there is a path in G from s to t where no event is observed. Therefore, as system behavior progresses, if any event is observed, then γ can be disabled. Hence, a set of colors $I = \{c_a, \ldots, c_z\}$ is a colored cut for D if and only if selecting the sensors $\{\sigma_a, \ldots, \sigma_z\}$ corresponding to I makes the system observable. Therefore any approximation algorithm for the minimal-cost sensor-selection problem can also be used with the same absolute effectiveness for the minimal-cost colored cut problem.

The converse construction is now shown to convert an instance of Problem 3 to an instance of Problem 1.

Proposition 2 *An instance of the minimal-cost sensor-selection problem can be reduced to an instance of the minimal-cost colored cut problem in polynomial time.*

The reduction construction relies on a modified $\mathcal{M}$-automaton method for testing observability where a automaton $\mathcal{M}_{\Sigma_o} = (X^{\mathcal{M}_{\Sigma_o}}, x_0^{\mathcal{M}_{\Sigma_o}}, \Sigma^{\mathcal{M}_{\Sigma_o}}, \delta^{\mathcal{M}_{\Sigma_o}})$ is constructed.

Suppose $H = (X^H, x_0^H, \Sigma, \delta^H)$, $G = (X^G, x_0^G, \Sigma, \delta^G)$, Σ_o and Σ_c are given such that it is desired to test if $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c. Let Σ' be a copy of the event set Σ where for every event $\sigma \in \Sigma$, there is a corresponding event $\sigma' \in \Sigma'$. The following are then defined, $X^{\mathcal{M}_{\Sigma_o}} := X^H \times X^H \times X^G \cup \{d\}$,$x_0^{\mathcal{M}_{\Sigma_o}} := (x_0^H, x_0^H, x_0^G)$and $\Sigma^{\mathcal{M}_{\Sigma_o}} := \Sigma \cup \Sigma'$. A set of conditions at state (x_1, x_2, x_3) is also defined that is called the $(*)$ conditions.

$$\left.\begin{array}{ll} \delta^H(x_1,\sigma) & \text{is defined if } \sigma \in \Sigma_c \\ \delta^H(x_2,\sigma) & \text{is not defined} \\ \delta^G(x_3,\sigma) & \text{is defined} \end{array}\right\} \qquad (*)$$

The nondeterministic transition relation $\delta^{\mathcal{M}_{\Sigma_o}}$ is defined as follows.
For $\sigma' \in \Sigma'$ such that for the corresponding $\sigma \in \Sigma$ $\sigma \notin \Sigma_o$,

$$\delta^{\mathcal{M}_{\Sigma_o}}((x_1,x_2,x_3),\sigma') = \left\{ \begin{array}{ll} (\delta^H(x_1,\sigma), x_2, x_3) & \text{if } \delta^H(x_1,\sigma)! \\ (x_1, \delta^H(x_2,\sigma), \delta^G(x_3,\sigma)) & \text{if } \big(\delta^H(x_2,\sigma)! \wedge \delta^G(x_3,\sigma)!\big) \end{array} \right\}$$

For $\sigma \in \Sigma$,

$$\delta^{\mathcal{M}_{\Sigma_o}}((x_1,x_2,x_3),\sigma) = \\ \left\{ \begin{array}{ll} (\delta^H(x_1,\sigma), \delta^H(x_2,\sigma), \delta^G(x_3,\sigma)) & \text{if } \big(\delta^H(x_1,\sigma)! \wedge (\delta^H(x_2,\sigma)! \wedge (\delta^G(x_3,\sigma)!\big) \\ d & \text{if } (*) \end{array} \right\}$$

No other transitions are defined in $\mathcal{M}_{\Sigma_o}$. The notation is used such that $\delta^H(x,\sigma)!$ is true if $\delta^H(x,\sigma)$ is defined and false otherwise. A similar definition holds for $\delta^G(x,\sigma)!$.

For $\sigma \in \Sigma$, $\delta^{\mathcal{M}_{\Sigma_o}}(d,\sigma)$ is undefined. The $\mathcal{M}_{\Sigma_o}$-automaton here is modified from the original in [20] in that Σ' transitions replace some Σ transitions. These $\sigma' \in \Sigma'$ transitions correspond to transitions that would be cut if $\sigma \in \Sigma \setminus \Sigma_o$ were to be made observable. This construction prompts the following proposition.

Proposition 3 *[20] The state d is reachable in $\mathcal{M}_{\Sigma_o}$ if and only if $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c.*

Effectively $\mathcal{M}_{\Sigma_o}$ is a nondeterministic simulation of an observer's estimate of a system's behavior with respect to a specification. As system behavior progresses, in a state (x_1, x_2, x_3) of $\mathcal{M}_{\Sigma_o}$, the first state is an observers estimate of the specification state and the second and third states are the true states of the specification and system respectively. In $\mathcal{M}_{\Sigma_o}$, a Σ transition occurs if an event occurrence in H and G is correctly predicted by the observer and a Σ' transition occurs if the prediction is not correct. Therefore, if an event σ is observed and added to Σ_o, the occurrences of σ would therefore be predicted correctly and a σ' transitions would be removed from $\mathcal{M}_{\Sigma_o}$. This implies that $\mathcal{M}_{\Sigma_o \cup \{\sigma\}}$ can be constructed from $\mathcal{M}_{\Sigma_o}$ by cutting all σ' transitions, and cutting all occurrences of σ' transitions in $\mathcal{M}_{\Sigma_o}$ corresponds to adding σ to Σ_o.

Lemma 1 *The automaton $\mathcal{M}_{\Sigma_o}$ can be constructed from $\mathcal{M}_\emptyset$ by iteratively cutting Σ_o' labelled transitions in $\mathcal{M}_\emptyset$.*

Proof: This lemma is shown by a proof by induction on the cardinality of Σ_o.

Base: Suppose $\Sigma_o = \emptyset$. This case is trivial as $\mathcal{M}_{\Sigma_o} = \mathcal{M}_\emptyset$.

Induction hypothesis: For $|\Sigma_o| = n$, the $\mathcal{M}_{\Sigma_o}$ automaton can be constructed from $\mathcal{M}_\emptyset$ by iteratively cutting Σ_o' labelled transitions in $\mathcal{M}_\emptyset$.

Induction step: Let $|\Sigma_o| = n$. From the induction hypothesis it is known that the $\mathcal{M}_{\Sigma_o}$ automaton can be constructed from $\mathcal{M}_\emptyset$ by iteratively cutting Σ_o' labelled transitions in $\mathcal{M}_\emptyset$.

Let σ be some event in $\Sigma \setminus \Sigma_o$. From the construction of $\mathcal{M}_{\Sigma_o}$ and $\mathcal{M}_{(\Sigma_o \cup \{\sigma\})}$, the only difference in the transition structure of these two automata is that transitions labelled by σ' are absent in $\mathcal{M}_{(\Sigma_o \cup \{\sigma\})}$. Therefore the $\mathcal{M}_{(\Sigma_o \cup \{\sigma\})}$ automaton can be constructed from $\mathcal{M}_{\Sigma_o}$ by cutting all σ' labelled transitions in $\mathcal{M}_{\Sigma_o}$. Hence, the $\mathcal{M}_{(\Sigma_o \cup \{\sigma\})}$ automaton can be constructed from $\mathcal{M}_{\emptyset}$ by iteratively cutting $\Sigma_o' \cup \{\sigma'\}$ labelled transitions in $\mathcal{M}_{\emptyset}$. ■

Lemma 1 shows that the sensor selection problem is really a type of colored cut problem. Suppose the automaton $\mathcal{M}_{\emptyset}$ is considered to be a colored directed graph as introduced above such that the transition labels are defined to be edge colors. A colored $x_0^{\mathcal{M}_{\emptyset}} d$-cut for $\mathcal{M}_{\emptyset}$ where only Σ' transitions are cut corresponds to a sufficient sensor selection for observability to hold. This prompts the following theorem.

Theorem 3 *$\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c if and only if $\Sigma_o' \subseteq \Sigma'$ is a colored $x_0^{\mathcal{M}_{\emptyset}} d$-cut $\mathcal{M}_{\emptyset}$.*

Proof: This proof is demonstrated in two parts. Suppose that $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c. Therefore, due to Proposition 3, d is not reachable in the $\mathcal{M}_{\Sigma_o}$ automaton constructed from H, G, Σ_c and Σ_o. From Lemma 1 the automaton $\mathcal{M}_{\Sigma_o}$ can be constructed from $\mathcal{M}_{\emptyset}$ by iteratively cutting Σ_o' labelled transitions in $\mathcal{M}_{\emptyset}$. Hence, Σ_o' is a colored $x_0^{\mathcal{M}_{\emptyset}} d$-cut in $\mathcal{M}_{\emptyset}$.

Now suppose that $\mathcal{L}(H)$ is not observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c. Therefore, due to Proposition 3, d is reachable in the $\mathcal{M}_{\Sigma_o}$ automaton constructed from H, G, Σ_c and Σ_o. From Lemma 1 the automaton $\mathcal{M}_{\Sigma_o}$ can be constructed from $\mathcal{M}_{\emptyset}$ by iteratively cutting Σ_o' labelled transitions in $\mathcal{M}_{\emptyset}$. Hence, Σ_o' is a not a colored $x_0^{\mathcal{M}_{\emptyset}} d$-cut in $\mathcal{M}_{\emptyset}$. ■

The Σ transitions cannot be cut in the $\mathcal{M}_{\emptyset}$ automaton of Theorem 3 by making events observable, so the $\mathcal{M}_{\emptyset}$ cut problem is not currently in a form that can be used for a reduction to Problem 1. To counter this difference, the following construction is used which performs a form of state condensation and hides the Σ transitions in $\mathcal{M}_{\emptyset}$.

Construct $\mathcal{M}_{\Sigma_o}$ from H, G, Σ_c and Σ_o. Define:

$$X_x^{\mathcal{M}_{\Sigma_o}} = \left\{ y^{\mathcal{M}_{\Sigma_o}} | \exists t \in \Sigma^*, \delta^{\mathcal{M}_{\Sigma_o}}(x^{\mathcal{M}_{\Sigma_o}}, t) = y^{\mathcal{M}_{\Sigma_o}} \right\}.$$

$X_x^{\mathcal{M}_{\Sigma_o}}$ represents all states that could be reached from $x^{\mathcal{M}_{\Sigma_o}}$ in $\mathcal{M}_{\Sigma_o}$ if only Σ transitions were allowed. The states in $X_x^{\mathcal{M}_{\Sigma_o}}$ would be reachable from $x^{\mathcal{M}_{\Sigma_o}}$ in $\mathcal{M}_{\Sigma_o}$ no matter what events were added to the observability set because only Σ_o' transitions can be cut by making events observable. With this in mind, the following nondeterministic automaton $\tilde{\mathcal{M}}_{\Sigma_o}$ is constructed from $\mathcal{M}_{\Sigma_o}$. It is assumed that $d \notin X_{x_0}^{\mathcal{M}_{\Sigma_o}}$. Let $\tilde{\mathcal{M}}_{\Sigma_o} = (X^{\tilde{\mathcal{M}}_{\Sigma_o}}, x_0^{\tilde{\mathcal{M}}_{\Sigma_o}}, \Sigma^{\tilde{\mathcal{M}}_{\Sigma_o}}, \delta^{\tilde{\mathcal{M}}_{\Sigma_o}})$, where $X^{\tilde{\mathcal{M}}_{\Sigma_o}} := X^H \times X^H \times X^G \cup \{d\}, x_0^{\tilde{\mathcal{M}}_{\Sigma_o}} := (x_0^H, x_0^H, x_0^G)$ and $\Sigma^{\tilde{\mathcal{M}}_{\Sigma_o}} := \Sigma$.

The transition relation $\delta^{\tilde{\mathcal{M}}_{\Sigma_o}}$ is defined as follows.

Suppose three states $x^{\mathcal{M}_{\Sigma_o}}, y^{\mathcal{M}_{\Sigma_o}}, z^{\mathcal{M}_{\Sigma_o}} \in X^{\mathcal{M}_{\Sigma_o}}$ and an event $\sigma \in \Sigma$ exist such that $z^{\mathcal{M}_{\Sigma_o}} \in X_x^{\mathcal{M}_{\Sigma_o}}$, $\delta^{\mathcal{M}_{\Sigma_o}}(z^{\mathcal{M}_{\Sigma_o}}, \sigma') = y^{\mathcal{M}_{\Sigma_o}}$.

$$\delta^{\tilde{\mathcal{M}}_{\Sigma_o}}(x^{\mathcal{M}_{\Sigma_o}}, \sigma) = \begin{cases} y^{\mathcal{M}_{\Sigma_o}} & \text{if } d \notin X_y^{\mathcal{M}_{\Sigma_o}} \\ d & \text{if } d \in X_y^{\mathcal{M}_{\Sigma_o}} \end{cases}$$

The $\tilde{\mathcal{M}}_{\Sigma_o}$ automaton is really a colored directed graph where states are vertices, transitions are directed edges and the transition labels are the colors. This prompts one of the main results of this paper.

Theorem 4 *Given an $\tilde{\mathcal{M}}_\emptyset$ automaton constructed from H, G, Σ_c and $\emptyset$ as the set of observable events, $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c if and only if Σ_o is a colored $x_0^{\tilde{\mathcal{M}}_\emptyset}d$-cut in the colored directed graph $\tilde{\mathcal{M}}_\emptyset$.*

Proof: It has already been shown that $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c if and only if Σ'_o is a colored $x_0^{\mathcal{M}_\emptyset}d$-cut in the colored directed graph $\mathcal{M}_\emptyset$. Therefore it is sufficient to show that Σ_o is a colored $x_0^{\tilde{\mathcal{M}}_\emptyset}d$-cut in the colored directed graph $\tilde{\mathcal{M}}_\emptyset$ if and only if Σ'_o is a colored $x_0^{\mathcal{M}_\emptyset}d$-cut in the colored directed graph $\mathcal{M}_\emptyset$.

Define a natural projection operation $P' : \Sigma \cup \Sigma' \to \Sigma'$. Also define the translation operator $\tilde{\Psi} : \Sigma' \to \Sigma$ such that $\tilde{\Psi}(\sigma') = \sigma$. Both of these functions are extended in the usual manner to be defined over strings. Also define the function $\tilde{P} : \Sigma \cup \Sigma' \to \Sigma$ that is the composition of $P'(\cdot)$ and $\tilde{\Psi}(\cdot)$, i.e., $\tilde{P}(\sigma) = \tilde{\Psi}(P'(\sigma))$. These functions also have the normally defined inverse operations.

Suppose that Σ'_o is not a colored $x_0^{\mathcal{M}_\emptyset}d$-cut in the colored directed graph $\mathcal{M}_\emptyset$. Then there exists a string $s \in (\Sigma \cup \Sigma')^*$ such that $\delta^{\mathcal{M}_\emptyset}(x_0^{\mathcal{M}_\emptyset}, s) = d$. Due to the construction of $\tilde{\mathcal{M}}_\emptyset$, $\delta^{\tilde{\mathcal{M}}_\emptyset}(x_0^{\tilde{\mathcal{M}}_\emptyset}, \tilde{P}(s)) = d$.

Now suppose that Σ'_o is not a colored $x_0^{\tilde{\mathcal{M}}_\emptyset}d$-cut in the colored directed graph $\tilde{\mathcal{M}}_\emptyset$. Then there exists a string $s \in \Sigma^*$ such that $\delta^{\tilde{\mathcal{M}}_\emptyset}(x_0^{\tilde{\mathcal{M}}_\emptyset}, s) = d$. Due to the construction of $\tilde{\mathcal{M}}_\emptyset$, there exists some string $t \in \tilde{P}^{-1}(s)$ such that $\delta^{\mathcal{M}_\emptyset}(x_0^{\mathcal{M}_\emptyset}, t) = d$. ■

With the shown conversions between the graph cutting problem and the sensor selection problem any method developed to calculate approximate solutions to one problem can be used to calculate approximate solutions to the other problem.

Due the construction of the $\mathcal{M}_{\Sigma_o}$-automaton, it should be apparent that $|X^{\mathcal{M}}| \leq |X^G| * |X^H|^2 + 1$. Furthermore, at each state $x^{\mathcal{M}}_{\Sigma_o} \in X^{\mathcal{M}}_{\Sigma_o}$, the number of state transitions is at most three times the maximum number of output state transitions in any state of G or H. If E^G is the set of state transitions in G and E^H is the number of state transitions in H, let $e = \max\{|E^G|, |E^H|\}$. Therefore $\mathcal{M}_{\Sigma_o}$ can be constructed in time and space in $O(e * |X^G| * |X^H|^2)$ using standard breadth-first digraph construction algorithms. Therefore, because reachability can be tested in polynomial time, the observability of $\mathcal{L}(H)$ with respect to $\mathcal{L}(G)$, Σ_o and Σ_c can be tested in polynomial time.

7 A Deterministic Greedy Graph Cutting Method

An algorithm is now shown for approximating the solution to the minimal-cost sensor-selection problem. This algorithm is based on the $\tilde{\mathcal{M}}_\emptyset$ construction seen above for a system G, a specification H and a set of controllable events Σ_c. After constructing $\tilde{\mathcal{M}}_\emptyset$, events are made observable in order to cut all paths from $x_o^{\tilde{\mathcal{M}}_\emptyset}$ to d in $\tilde{\mathcal{M}}_\emptyset$.

A utility function is now used to decide which events to make observable by determining the relative desirability of cutting a set of transitions associated with an event in $\tilde{\mathcal{M}}_\emptyset$. Starting with a trim version of $\tilde{\mathcal{M}}_\emptyset$, suppose in this automaton it is desirable to find the "probability" $\mathcal{P}(\sigma, \tilde{\mathcal{M}}_\emptyset)$ that a "randomly" selected path from $x_o^{\tilde{\mathcal{M}}_\emptyset}$ to d contains an edge labelled by σ. The term "probability" and "randomly" are used in a loose and intuitive manner in order to develop an understanding for the solution method for this problem while avoiding the explicit definition of a probability distribution function at this time. Naturally it would be desirable to cut transitions associated with events that have the highest probability of occurrence as specified by $\mathcal{P}(\sigma, \tilde{\mathcal{M}}_\emptyset)$. If an event has a low sensor cost and a high probability of occurring on a simulation of the $\tilde{\mathcal{M}}_{\emptyset}$-automaton leading to the d state, then it is desirable to observe occurrences of that event as observing that event would cut the "most" paths to the d state in $\tilde{\mathcal{M}}_{\emptyset}$ per sensor cost. This prompts the following greedy approximation algorithm.

Algorithm 3 Deterministic Greedy Approximation Algorithm (DetGrAprx)

Input: $G = (X^G, \Sigma, \delta^G, x_0^G)$, $H = (X^H, \Sigma, \delta^H, x_0^H)$, $\Sigma_c \subseteq \Sigma$;
$\Sigma_o \leftarrow \emptyset$;
$i \leftarrow 1$;
Construct $\tilde{\mathcal{M}}_{\Sigma_o}$;
$\tilde{\mathcal{M}}_{\Sigma_o}^T \leftarrow Trim(\tilde{\mathcal{M}}_{\Sigma_o})$;
While $\mathcal{L}_m(\tilde{\mathcal{M}}_{\Sigma_o}^T) \neq \emptyset$;
{
$\sigma_i \leftarrow \arg\max_{\sigma \in \Sigma \setminus \Sigma_o} \left(\frac{\mathcal{P}(\sigma, \tilde{\mathcal{M}}_{\Sigma_o}^T)}{cost(\sigma_i)} \right)$;
$\rho_i \leftarrow \frac{\mathcal{P}(\sigma_i, \tilde{\mathcal{M}}_{\Sigma_o}^T)}{cost(\sigma_i)}$;
$\Sigma_o \leftarrow \Sigma_o \cup \{\sigma_i\}$;
$k \leftarrow i$;
$i \leftarrow i + 1$;
Construct $\tilde{\mathcal{M}}_{\Sigma_o}$;
$\tilde{\mathcal{M}}_{\Sigma_o}^T \leftarrow Trim(\tilde{\mathcal{M}}_{\Sigma_o})$;
}
Return Σ_o;

It now needs to be shown how $\mathcal{P}\left(\sigma, \tilde{\mathcal{M}}_{\Sigma_o}^T\right)$ is calculated. This is done by converting $\tilde{\mathcal{M}}_{\Sigma_o}^T$ into a stochastic automaton. At each state $x \in X^{\tilde{\mathcal{M}}_{\Sigma_o}}$, suppose there are κ_x output transitions. Assign the probability $\frac{1}{\kappa_x}$ to each output transition of x. That probability assignment models that all the output transitions of a state have the same probability of being followed. Therefore, using standard methods from stochastic systems theory [6], $\mathcal{P}(\sigma, \tilde{\mathcal{M}}_{\Sigma_o}^T)$ denotes the probability that a random walk in the stochastic version of $\tilde{\mathcal{M}}_{\Sigma_o}^T$ with uniform probability assignments starting at $x_o^{\tilde{\mathcal{M}}_{\Sigma_o}^T}$ traverses a σ transition on its way to d. It should be noted that this probability can be computed in polynomial time using standard methods.

Algorithm 3 iteratively chooses to observe an event with the highest probability of occurrence in $\tilde{\mathcal{M}}_{\Sigma_o}^T$ over the sets of all $x_o^{\tilde{\mathcal{M}}_{\Sigma_o}^T} d$ paths per the cost of observing that event. After an event is selected to be observed, it is added to Σ_o and all transitions associated with that event are removed in $\tilde{\mathcal{M}}_{\Sigma_o}^T$. The $\tilde{\mathcal{M}}_{\Sigma_o}^T$-automaton is continually trimmed as events are made observable until there are no paths from the initial state to the marked state d. Therefore, as Σ_o is updated, the next $\tilde{\mathcal{M}}_{\Sigma_o}^T$ can be calculated in polynomial time. Algorithm 3 runs in polynomial time with respect to the size of the encodings of G and H as the algorithm iterates at most $k \leq |\Sigma|$ times.

The deterministic greedy algorithm is now analyzed to obtain a bound on the ratio of the cost of the sensor selection Σ_o returned by Algorithm 3 to the cost of the minimal observability set. This analysis relies on the stored $\{\rho_1, \ldots, \rho_k\}$ probability to cost ratios saved during the operation of Algorithm 3. The set $\Sigma_o^{\min_i}$ denotes the minimum-cost observability set that could be chosen at iteration i given that events in Σ_o^i are already selected to be observed. Naturally, $\Sigma_o^{\min_1} = \Sigma_o^{\min}$.

Lemma 2 *In Algorithm 3, on the ith iteration,*

$$\frac{cost(\sigma_i)}{\mathcal{P}\left(\sigma_i, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \leq cost(\Sigma_o^{\min_i})$$

Proof: Let $\Sigma_o^{\min_i} = \{\gamma_i^1, \dots \gamma_i^{k_i}\}$. Also, for $\gamma_i^j, \tilde{\mathcal{M}}_{\Sigma_o^i}^T$, $j \in \{1, \dots, k_i\}$, let values of $\alpha_i^j \in \left(0, \mathcal{P}\left(\gamma_i^j, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)\right]$ be chosen such that $\sum_{j=1}^{k_i} \alpha_i^j = 1$. Because Algorithm 3 is a greedy algorithm, σ_i has the highest probability of any event not previously chosen for observation on round i, i.e.,

$$\begin{aligned}
& \forall j \in \{1, \dots, k_i\} \left(\frac{\mathcal{P}\left(\sigma_i, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)}{cost(\sigma_i)} \geq \frac{\mathcal{P}\left(\gamma_i^j, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)}{cost(\gamma_i^j)} \right) \\
\Rightarrow \quad & \forall j \in \{1, \dots, k_i\} \left(\frac{cost(\sigma_i)}{\mathcal{P}\left(\sigma_i, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \leq \frac{cost(\gamma_i^j)}{\mathcal{P}\left(\gamma_i^j, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \right) \\
\Rightarrow \quad & \forall j \in \{1, \dots, k_i\} \left(\frac{cost(\sigma_i)}{\mathcal{P}\left(\sigma_i, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \leq \frac{cost(\gamma_i^j)}{\alpha_i^j} \right) \\
\Rightarrow \quad & \forall j \in \{1, \dots, k_i\} \left(\frac{\alpha_i^j cost(\sigma_i)}{\mathcal{P}\left(\sigma_i, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \leq cost(\gamma_i^j) \right) \\
\Rightarrow \quad & \sum_{j=1}^{k_i} \frac{\alpha_i^j cost(\sigma_i)}{\mathcal{P}\left(\sigma_i, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \leq \sum_{j=1}^{k_i} cost(\gamma_i^j) \\
\Rightarrow \quad & \frac{cost(\sigma_i)}{\mathcal{P}\left(\sigma_i, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \sum_{j=1}^{k_i} \alpha_i^j \leq cost(\Sigma_o^{\min_i}) \\
\Rightarrow \quad & \frac{cost(\sigma_i)}{\mathcal{P}\left(\sigma_i, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \leq cost(\Sigma_o^{\min_i}).
\end{aligned}$$

■

Lemma 2 can now be used to show the following result on the closeness of the cost of the approximation to the minimum-cost observability set.

Theorem 5 *For the observability set Σ_o returned by Algorithm 3 and the minimum sensor selection $\Sigma_o^{\min}$,*

$$\frac{cost(\Sigma_o)}{cost(\Sigma_o^{\min})} \leq \sum_{i=1}^{|\Sigma_o|} \rho_i$$

where $\{\rho_1, \dots \rho_k\}$ are the values iterative probabilities stored during the operation of Algorithm 3.

Proof: It has already been shown in Lemma 2 that:

$$\begin{aligned}
\frac{cost(\sigma_i)}{\mathcal{P}\left(\sigma, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \leq cost(\Sigma_o^{\min_i}) \quad &\Rightarrow \quad cost(\sigma_i) \leq cost(\Sigma_o^{\min_i})\mathcal{P}\left(\sigma, \tilde{\mathcal{M}}_{\Sigma_o^i}\right) \\
&\Rightarrow \quad cost(\sigma_i) \leq cost(\Sigma_o^{\min_i})\rho_i \\
&\Rightarrow \quad cost(\Sigma_o) \leq \sum_{i=1}^{|\Sigma_o|} cost(\Sigma_o^{\min_i})\rho_i \\
&\Rightarrow \quad \frac{cost(\Sigma_o)}{cost(\Sigma_o^{\min})} \leq \sum_{i=1}^{|\Sigma_o|} \rho_i
\end{aligned}$$

■

Because of Theorem 5, a bound on the closeness of the approximation returned by Algorithm 3 can be calculated. Unfortunately $\sum_{i=1}^{k} \rho_i$ can be on the order of $n - \epsilon$ in the worst case where n is the number of system events and ϵ is some constant greater than 0. A lower bound on the closeness of the bound on the approximation ratio shown in Theorem 5 is now shown if the algorithm is used on the special problem case of uniform cost as seen in Problem 2.

Theorem 6 *From a set $\{\rho_1, \ldots, \rho_k\}$ calculated from a running of Algorithm 3 where $\forall \sigma \in \Sigma, cost(\sigma) = \lambda \in \Re^+$,*

$$\sum_{i=1}^{k} \rho_i \geq H_k = \sum_{j=1}^{k} \frac{1}{j}$$

where $H_k = \sum_{j=1}^{k} \frac{1}{j}$ is the sum of the harmonic series.

Proof: Suppose the events chosen to be observable by Algorithm 3, Σ_o, are chosen in the order $\sigma_1, \ldots, \sigma_k$. In the automaton $\tilde{\mathcal{M}}_{\Sigma_o^i}$, the transition labelled by σ_i has probability ρ_i of occurring on a random path from $x_o^{\tilde{\mathcal{M}}_{\Sigma_o^i}}$ to d. Before the event σ_i is chosen, there are $k-i+1$ events left to be chosen in the set $\{\sigma_i, \ldots, \sigma_k\}$. All unique paths from $x_o^{\tilde{\mathcal{M}}_{\Sigma_o^i}}$ to d in $\tilde{\mathcal{M}}_{\Sigma_o^i}$ must contain at least one state transition with a label in $\{\sigma_i, \ldots, \sigma_k\}$.

Therefore,

$$\sum_{j=i}^{k} \mathcal{P}(\sigma_j, \tilde{\mathcal{M}}_{\Sigma_o^i}) \geq 1.$$

Because σ_i has the highest probability of occurring on a path from $x_o^{\tilde{\mathcal{M}}_{\Sigma_o^i}}$ to d in $\tilde{\mathcal{M}}_{\Sigma_o^i}$,

$$\begin{aligned}
\rho_i \geq \frac{\sum_{j=i}^{k} \mathcal{P}(\sigma_j, \tilde{\mathcal{M}}_{\Sigma_o^i})}{k-i+1} \quad &\Rightarrow \quad \rho_i \geq \frac{1}{k-i+1} \\
&\Rightarrow \quad \sum_{i=k}^{1} \rho_i \geq \sum_{i=k}^{1} \frac{1}{k-i+1} \\
&\Rightarrow \quad \sum_{i=1}^{k} \rho_i \geq H_k.
\end{aligned}$$

■

Although Theorem 6 puts a lower bound on the guarantee of the approximation ratio show in Theorem 5, it is not implied that Algorithm 3 cannot have an approximation ratio better than H_k when the sensors have uniform cost.

7.1 A Randomized Greedy Algorithm

A randomized greedy minimal sensor selection algorithm is now given that combines elements of Algorithms 1 and 3. As with Algorithm 1, this new algorithm randomly enables events to be made observable, but uses the utility function $\mathcal{P}(\sigma, \tilde{\mathcal{M}}_{\Sigma_o})$ to weight the probability distribution of a sensor being selected. Therefore, an event with a relatively high probability of occurring over the set of all paths to d in $\tilde{\mathcal{M}}_{\Sigma_o}$ will have a higher probability of being added to the observable events when sensors are being selected.

Algorithm 4 Randomized Weighted Observability Set Search Algorithm (RanWObs):

Input: $G = (X^G, \Sigma, \delta^G, x_0^G)$, $H = (X^H, \Sigma, \delta^H, x_0^H)$, $\Sigma_c \subseteq \Sigma$

$\Sigma_o \leftarrow \emptyset$;

Construct $\tilde{\mathcal{M}}_{\Sigma_o}$ *from* G, H, Σ_c *and* Σ_o;

$i \leftarrow 1$;

$\tilde{\mathcal{M}}^T_{\Sigma_o} \leftarrow Trim(\tilde{\mathcal{M}}^T_{\Sigma_o})$;

While $\mathcal{L}_m(\tilde{\mathcal{M}}^T_{\Sigma_o}) \neq \emptyset$;

{

For all $\sigma \in \Sigma \setminus \Sigma_o$

{

$$\Pr(\sigma) \leftarrow \frac{\left(\frac{\mathcal{P}\left(\sigma, \tilde{\mathcal{M}}^T_{\Sigma_o}\right)}{cost(\sigma_i)}\right)}{\sum_{\gamma \in \Sigma \setminus \Sigma_o} \left(\frac{\mathcal{P}\left(\gamma, \tilde{\mathcal{M}}^T_{\Sigma_o}\right)}{cost(\gamma)}\right)};$$

}

Randomly select $\sigma_i \in \Sigma \setminus \Sigma_o$ *according to probability distribution* $\Pr(\sigma)$;

$k \leftarrow i$;

Remove σ_i *labelled transitions in* $\tilde{\mathcal{M}}^T_{\Sigma_o}$;

$\Sigma_o \leftarrow \Sigma_o \cup \{\sigma_i\}$;

$i \leftarrow i + 1$;

$\tilde{\mathcal{M}}^T_{\Sigma_o} \leftarrow Trim(\tilde{\mathcal{M}}^T_{\Sigma_o})$;

}

Return Σ_o;

Algorithm 4 can be iterated multiple times using a method similar to that in Algorithm 2 to obtain multiple approximations to the minimum-cost observability set. Unlike the deterministic approximation algorithm, Algorithm 3, Algorithm 4 may not always return the same approximation.

8 Integer Programming

Another approach to approximating the minimal cost sensor selection is to use integer programming based methods. Integer programming is a general optimization problem from the field of the combinatorial optimization that has been well explored in the literature [12].

This section discusses how to convert the minimal cost sensor selection problem to an integer programming problem so that the approximation methods developped for the integer programming problem can be used on the sensor selection problem.

First the integer programming problem is introduced.

Problem 4 The Integer Programming Problem: *Given a z element row vector C, a $y \times z$ matrix A and a y element column vector B, find a z element column vector $\vec{x} \in \{0,1\}^z$ that minimizes $C\vec{x}$ subject to $A\vec{x} \geq B$.*

The integer programming problem is known to be NP-complete, but there is a vast literature on calculating approximate solutions to this problem as outlined in [12, 22]. Unfortunately, the integer programming problem is known to be NPO-complete [1] which means that it is in the most difficult class of NP-complete optimization problems. However, if the sensor selection problem is in the form of an integer programming problem, already developed methods for the well understood integer programming can be used to find solutions to the sensor selection problem.

8.1 Problem Conversion

It is now shown how to convert the minimal cost sensor selection problem to an integer programming problem. Suppose a system automaton G, a specification automaton H and a set of controllable events Σ_c are given along with a cost function $cost : \Sigma \to \Re^+ \cup \{0\}$. From this the automaton $\tilde{\mathcal{M}}_\emptyset$ and $\tilde{\mathcal{M}}_{\Sigma_o}$ can be constructed for some $\Sigma_o \subseteq \Sigma$. Note that for the sets of reachable states, $X^{\tilde{\mathcal{M}}_{\Sigma_o}} \subseteq X^{\tilde{\mathcal{M}}_\emptyset}$. That is, some reachable states in $\tilde{\mathcal{M}}_\emptyset$ may not be reachable in $\tilde{\mathcal{M}}_{\Sigma_o}$.

Let the events in Σ and states in $X^{\tilde{\mathcal{M}}_\emptyset}$ are treated as binary variables. The events in the set $\Sigma_o \subseteq \Sigma$ are all assigned 1 and all events in $\Sigma \setminus \Sigma_o$ are assigned 0. For all states $x \in X^{\tilde{\mathcal{M}}_\emptyset}$ such that x is reachable from $x_o^{\tilde{\mathcal{M}}_\emptyset}$ according to the transition rules of $\tilde{\mathcal{M}}_{\Sigma_o}$, then $x = 1$. Arbitrary binary assignments are made to the variables representing the unreachable states. Note that $x_o^{\tilde{\mathcal{M}}_\emptyset} = x_o^{\tilde{\mathcal{M}}_{\Sigma_o}} = 1$.

To express the validity of variable assignments as a set of inequalities, suppose that in $\tilde{\mathcal{M}}_\emptyset$ there is an event $\sigma_{i_1} \in \Sigma$ and two states $x_{i_2}, x_{i_3} \in X^{\tilde{\mathcal{M}}_\emptyset}$ such that $x_{i_2} \overset{\sigma_{i_1}}{\longmapsto}_{\tilde{\mathcal{M}}_\emptyset} x_{i_3}$. Therefore, using the variable assignments described above, this transition can be written as an inequality $x_{i_3} \geq x_{i_2} - \sigma_{i_1}$. This represents the property for the automaton $\tilde{\mathcal{M}}_{\Sigma_o}$ that if x_{i_2} is reachable in $\tilde{\mathcal{M}}_{\Sigma_o}$ and there is a transition caused by an unobserved event σ_{i_1} that leads to x_{i_3}, then x_{i_3} should also be reachable. This inequality can be manipulated so that $x_{i_3} - x_{i_2} + \sigma_{i_1} \geq 0$.

Therefore, if all of the state transitions in $\tilde{\mathcal{M}}_\emptyset$ are expressed as integer inequalities as was done for the $x_{i_2} \overset{\sigma_{i_1}}{\longmapsto}_{\tilde{\mathcal{M}}_\emptyset} x_{i_3}$ transition above and the initial state $x_o^{\tilde{\mathcal{M}}_\emptyset}$ is constrained to be reachable (that is, assigned to be 1), then the problem is to find the minimal cost set of events assigned to be 1 (that is, assigned to be observable) such that the d state does not have to be assigned 1 in order for the set of all transition inequalities to be valid. This set of conditions can now be converted into an integer programming problem. Let a vector $\vec{x}$ be defined such that if $\Sigma = \{\sigma_1, \ldots, \sigma_k\}$ and $X^{\tilde{\mathcal{M}}_\emptyset} = \{x_0, x_1, \ldots, x_{n-1}, d\}$:

$$\vec{x} = \begin{bmatrix} \sigma_1 & \cdots & \sigma_k & x_0 & x_1 & \cdots & x_{n-1} & d \end{bmatrix}^T.$$

Therefore, $z = k + n$.

The row vector C can be a z element constant vector such that for all $i \in \{1, \ldots, k\}$, the ith entry of C is $cost(\sigma_i)$ and all other entries of C are 0.

$$C = \begin{bmatrix} cost(\sigma_1) & \cdots & cost(\sigma_k) & 0 & 0 & \cdots & 0 & 0 \end{bmatrix}$$

Therefore, minimizing $C\vec{x}$ is equivalent to minimizing the cost of a sensor selection.

To encode the integer inequalities representing the state transitions such as $x_{i_2} \overset{\sigma_{i_1}}{\mapsto}_{\tilde{\mathcal{M}}_\emptyset} x_{i_3}$ in $\tilde{\mathcal{M}}_\emptyset$ such that $A\vec{x} \geq B$ and ensure that d is assigned 0 while $x_0^{\tilde{\mathcal{M}}_\emptyset}$ is assigned 1, construct A and B as follows.

First, suppose there are e transitions in $\tilde{\mathcal{M}}_\emptyset$ and let $y = e + 2$. The matrix B is a y element column vector and A is a $y \times z$. Let the transitions be ordered from 1 to e, such that transition l represents $x_{l_2} \overset{\sigma_{l_1}}{\mapsto}_{\tilde{\mathcal{M}}_\emptyset} x_{l_3}$. For the lth transition, the occurrence of the l_1th event at the l_2th state results in $\tilde{\mathcal{M}}_\emptyset$ entering the l_3th state. Therefore, entry (l, l_1) in A is assigned to be 1, entry $(l, (k + l_3))$ in A is assigned to be 1, entry $(l, (k + l_2))$ in A is assigned to be -1 and entry l of B is assigned to be 0. All other entries in the top e rows of A are assigned 0. This results in the inequality $x_{l_3} - x_{l_2} + \sigma_{l_1} \geq 0$ being encoded in $A\vec{x} \geq B$.

The (y, z) entry of A is assigned to be -1 and the y entry of B is assigned to be 0. This ensures that state d is not reachable. That is, $-d \geq 0$, so that $d \leq 0$. If $l_{x_0^{\tilde{\mathcal{M}}_\emptyset}}$ represents the row of $\vec{x}$ corresponding to the state $x_0^{\tilde{\mathcal{M}}_\emptyset}$, then the $(y - 1, l_{x_0^{\tilde{\mathcal{M}}_\emptyset}})$ entry of A and the $l_{x_0^{\tilde{\mathcal{M}}_\emptyset}}$ entry of B are assigned to be 1. This guarantees the 1 is assigned to the binary variable representing the reachability of the initial state $x_0^{\tilde{\mathcal{M}}_\emptyset}$. All of the other entries in the bottom two rows of A are assigned to be 0. That is $x_0^{\tilde{\mathcal{M}}_\emptyset} \geq 1$.

Therefore, B is a column vector of $(e + 1)$ 0's and a 1 at the row corresponding to $x_0^{\tilde{\mathcal{M}}_\emptyset}$.

$$B = \begin{bmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}^T.$$

Using the given construction of A, B, C and $\vec{x}$ constructed from G, H, Σ_c and $cost(\cdot)$, the minimal cost sensor selection problem is now in the form of an integer programming problem where $C\vec{x}$ should be minimized subject to the constraint that $A\vec{x} \geq B$. An example of the use of the integer programming construction for the sensor selection problem is now given.

Example 2 *As a simple example of how to convert a graph cutting version of a sensor selection problem into an integer programming problem, consider the $\tilde{\mathcal{M}}_\emptyset$ automaton seen in Figure 4.*

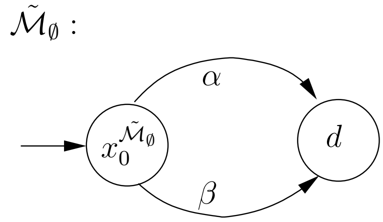

Figure 4: An example of a $\tilde{\mathcal{M}}_\emptyset$ automaton.

It is assumed that the system events have uniform cost of being observed. Therefore,

$$\vec{x} = \begin{bmatrix} \alpha \\ \beta \\ x_o^{\tilde{\mathcal{M}}_\emptyset} \\ d \end{bmatrix}, \ A = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 0 & 1 & -1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \ B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \ C = \begin{bmatrix} 1 & 1 & 0 & 0 \end{bmatrix}.$$

9 Applications to Minimal Communication Decentralized Control

An approximation problem for communicating decentralized control systems is now investigated where decentralized controllers make local observations, communicate the occurrence of various events between each other and enforce local control actions that are combined globally using an intersection operation. That is, an event is enabled globally only if it is enabled by all local controllers. Two controller systems are investigated, but the results presented can be trivially extended to n controller systems.

For this problem a system G and a specification H are given with respective sets of locally controllable events Σ_{c1}, Σ_{c2} and locally observable events Σ_{o1}, Σ_{o2}. It is assumed that the controllers communicate by reporting all occurrences of some locally observed events to the other controller. More formally, controller 1 communicates all observations of the occurrences of a subset of the locally observable events $\Sigma_{o12} \subseteq \Sigma_{o1}$ to controller 2 and controller 2 communicates all observations of the occurrence of locally observable events $\Sigma_{o21} \subseteq \Sigma_{o2}$ to controller 1.

It is assumed that $\mathcal{L}(H)$ is not co-observable with respect to $\mathcal{L}(G)$, Σ_{o1}, Σ_{o2} and Σ_{c1}, Σ_{c2}, but $\mathcal{L}(H)$ is co-observable with respect to $\mathcal{L}(G)$, $(\Sigma_{o1} \cup \Sigma_{o2}), (\Sigma_{o2} \cup \Sigma_{o2})$ and Σ_{c1}, Σ_{c2}. This assumption implies that there exists non-trivial Σ_{o12} and Σ_{o21} such that $\mathcal{L}(H)$ is co-observable with respect to $\mathcal{L}(G)$, $(\Sigma_{o1} \cup \Sigma_{o21}), (\Sigma_{o2} \cup \Sigma_{o12})$ and Σ_{c1}, Σ_{c2}. When the communicated event sets Σ_{o12} and Σ_{o21} can be used to solve the communicating controller problem by augmenting the local observability sets, then the pair of sets $\Sigma_{o12}, \Sigma_{o21}$ are called sufficient communication selections.

It is assumed that there is a non-zero cost associated with communicating events and this cost may be uniform over all events. Two functions $cost_{12} : \Sigma_{o1} \rightarrow \Re^+$ and $cost_{21} : \Sigma_{o2} \rightarrow \Re^+$ exist such that $cost_{12}(\sigma)$ represents the cost to controller 1 of communicating all observed occurrences of σ to controller 2, and $cost_{21}(\sigma)$ represents the cost to controller 2 of communicating all observed occurrences of σ to controller 1. The cost function functions can also be extended to be defined over sets of states as was done with $cost(\cdot)$ above. Due to reasons of economy or limited bandwidth the total communication cost may want to be minimized. This prompts the following minimization problem definition.

Problem 5 Minimal-Cardinality Communication-Selection: *Given a system G, a specification H and controllable events $\Sigma_{c1}, \Sigma_{c2} \subseteq \Sigma$, find a sufficient communication selection $\Sigma_{o12}^{min}, \Sigma_{o21}^{min}$ such that for any other sufficient communication selections $\Sigma_{o12}, \Sigma_{o21}$,*

$$cost_{12}(\Sigma_{o12}^{min}) + cost_{21}(\Sigma_{o21}^{min}) \leq cost_{12}(\Sigma_{o12}) + cost_{21}(\Sigma_{o21}).$$

Problem 5 is a special case of the communicating controller open problem discussed in [21]. It is hypothesized that the general communicating controller problem of [21] may be undecidable, but the computational methods developed for the centralized sensor selection problem above can be intuitively applied to the communicating decentralized control problem discussed here. It is now shown how to convert Problem 5 into a type of graph cutting problem.

9.1 Graph Cutting for Communication Selection

Problem 5 is a modification of Problem 1 and there exists a similar $\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}$ construction for converting the communication selection problem to a type of graph cutting problem. Let Σ_1 and Σ_2 be disjoint sets of events such that for all $i \in \{1, 2\}$, $\Sigma_i \cap \Sigma = \emptyset$. Furthermore, define $\Psi_i : \Sigma \to \Sigma_i$ for $i \in \{1,2\}$ to be a one-to-one function, and for $\sigma \in \Sigma$, $\Psi_i(\sigma)$ is called σ_i when it can be done without ambiguitively. Consider the following construction for the two controller case that can be easily extended for the n controller case.

$$\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}} = (X^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}, x_0^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}, (\Sigma \cup \Sigma_1 \cup \Sigma_2), \delta^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}, X_m^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}})$$

where

$$\begin{aligned} X^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} &:= X^H \times X^H \times X^H \times G^G \cup \{d\}, \\ x_0^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} &:= (x_0^H, x_0^H, x_0^H, x_0^G), \\ X_m^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} &:= \{d\}. \end{aligned}$$

Let a set of conditions be defined that together imply a violation of co-observability. Note that these conditions are only defined for the controllable events. For $\sigma \in \Sigma_c$, these conditions are called the $(*)$ conditions.

$$\left.\begin{array}{ll} \delta^H(x_1,\sigma) & \text{is defined if } \sigma \in \Sigma_{c1} \\ \delta^H(x_2,\sigma) & \text{is defined if } \sigma \in \Sigma_{c2} \\ \delta^H(x_3,\sigma) & \text{is not defined} \\ \delta^G(x_4,\sigma) & \text{is defined} \end{array}\right\} \qquad (*)$$

The transition relation for $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ is defined as follows:
For $\sigma \in \Sigma \setminus (\Sigma_{o1} \cup \Sigma_{o2})$:

$$\delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\sigma) =$$
$$\left\{\begin{array}{ll} d & \text{if } (*) \\ (\delta^H(x_1,\sigma),x_2,x_3,x_4) & \text{if } \delta^H(x_1,\sigma)! \\ (x_1,\delta^H(x_2,\sigma),x_3,x_4) & \text{if } \delta^H(x_2,\sigma)! \\ (x_1,x_2,\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) & \text{if } \big(\delta^H(x_3,\sigma)! \wedge \delta^G(x_4,\sigma)!\big) \\ (\delta^H(x_1,\sigma),\delta^H(x_2,\sigma),\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) & \text{if } \left(\begin{array}{c} \delta^H(x_1,\sigma)! \wedge \delta^H(x_2,\sigma)! \\ \wedge\delta^H(x_3,\sigma)! \wedge \delta^G(x_4,\sigma)! \end{array}\right) \end{array}\right\}.$$

For $\sigma \in \Sigma_{o2} \setminus (\Sigma_{o1} \cup \Sigma_{o21})$:

$$\delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\sigma) =$$
$$\left\{\begin{array}{ll} d & \text{if } (*) \\ (\delta^H(x_1,\sigma),\delta^H(x_2,\sigma),\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) & \text{if } \left(\begin{array}{c} \delta^H(x_1,\sigma)! \wedge \delta^H(x_2,\sigma)! \\ \wedge\delta^H(x_3,\sigma)! \wedge \delta^G(x_4,\sigma)! \end{array}\right) \end{array}\right\},$$

and

$$\delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\Psi_1(\sigma)) =$$

$$\left\{\begin{array}{ll} (\delta^H(x_1,\sigma),x_2,x_3,x_4) & \text{if } \delta^H(x_1,\sigma)! \\ (x_1,\delta^H(x_2,\sigma),\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) & \text{if } \big(\delta^H(x_2,\sigma)! \wedge \delta^H(x_3,\sigma)! \wedge \delta^G(x_4,\sigma)!\big) \end{array}\right\}.$$

For $\sigma \in \Sigma_{o1} \setminus (\Sigma_{o2} \cup \Sigma_{o12})$:

$$\delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\sigma) =$$
$$\left\{\begin{array}{ll} d & \text{if } (*) \\ (\delta^H(x_1,\sigma),\delta^H(x_2,\sigma),\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) & \text{if } \left(\begin{array}{c} \delta^H(x_1,\sigma)! \wedge \delta^H(x_2,\sigma)! \\ \wedge\delta^H(x_3,\sigma)! \wedge \delta^G(x_4,\sigma)! \end{array}\right) \end{array}\right\},$$

and

$$\delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\Psi_2(\sigma)) = \left\{ \begin{array}{ll} (x_1,\delta^H(x_2,\sigma),x_3,x_4) & \text{if } \delta^H(x_2,\sigma)! \\ (\delta^H(x_1,\sigma),x_2,\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) & \text{if } \left(\delta^H(x_1,\sigma)! \wedge \delta^H(x_3,\sigma)! \wedge \delta^G(x_4,\sigma)!\right) \end{array} \right\}.$$

For $\sigma \in \Sigma_{o1} \cap \Sigma_{o2}$:

$$\delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\sigma) = \left\{ \begin{array}{ll} d & \text{if } (*) \\ (\delta^H(x_1,\sigma),\delta^H(x_2,\sigma),\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) & \text{if } \left(\begin{array}{c} \delta^H(x_1,\sigma)! \wedge \delta^H(x_2,\sigma)! \\ \wedge\delta^H(x_3,\sigma)! \wedge \delta^G(x_4,\sigma)! \end{array} \right) \end{array} \right\}.$$

No other transitions are defined in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$.

The $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ construction prompts the following corollary to the main result of [16].

Corollary 2 *The state d is reachable from the initial state in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ if and only if $\mathcal{L}(H)$ is co-observable with respect to $\mathcal{L}(G)$, $(\Sigma_{o1} \cup \Sigma_{o21})$, $(\Sigma_{o2} \cup \Sigma_{o12})$, Σ_{c1} and Σ_{c2}.*

Note that the $\mathcal{M}_{(\Sigma_{o12}\cup\{\sigma\}),\Sigma_{o21}}$ automaton can be constructed from the $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ automaton by cutting all transitions labelled by σ_1 and the $\mathcal{M}_{\Sigma_{o12},(\Sigma_{o21}\cup\{\sigma\})}$ automaton can be constructed from the $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ automaton by cutting all transitions labelled by σ_2. Therefore, the act of controller i communicating all observed occurrences of event σ to controller j corresponds to trimming all σ_j labelled transitions in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$.

For the set of events $\Sigma_{oij} \subseteq \Sigma$, let $\Sigma_j^{oij} \subseteq \Sigma_j$ represent the corresponding set of events such that $\sigma \in \Sigma_{oij}$ if and only if $\sigma_j \in \Sigma_j^{oij}$. A set of events $\Sigma_1^{o21} \cup \Sigma_2^{o12}$ is a $x_0^{\mathcal{M}_{\emptyset,\emptyset}}d$-cut in $\mathcal{M}_{\emptyset,\emptyset}$ if and only if $\mathcal{L}_m(\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}) = \emptyset$ and consequently $\mathcal{L}(H)$ is co-observable with respect to $\mathcal{L}(G)$, $(\Sigma_{o1} \cup \Sigma_{o21})$, $(\Sigma_{o2} \cup \Sigma_{o12})$, Σ_{c1} and Σ_{c2}. Therefore, the pair of sets $\left(\Sigma_{o12}^{min}, \Sigma_{o21}^{min}\right)$ is a minimal-cost communication-selection if and only if the corresponding events $\Sigma_1^{o21min} \cup \Sigma_2^{o12min} \subseteq \Sigma_1 \cup \Sigma_2$ is a minimal-cost $x_0^{\mathcal{M}_{\emptyset,\emptyset}}d$-cut in $\mathcal{M}_{\emptyset,\emptyset}$ when restricted to cutting transitions labelled with events in $\Sigma_1 \cup \Sigma_2$.

As with the $\mathcal{M}_{\Sigma_o}$ construction given above the $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ construction converts the communicating controller selection problem into a type of graph cutting problem as long as only events in Σ_1 and Σ_2 are cut. There is a $\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}$ construction that can be used to convert this graph cutting problem into a true edge-colored directed graph st-cut problem.

Define:

$$X_x^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} = \left\{ y^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} | \exists t \in \Sigma^*, \delta^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}(x^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}, t) = y^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} \right\}.$$

$X_x^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}$ represents all states that could be reached from $x^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}$ in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ if only Σ transitions were allowed. The states in $X_x^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}$ would be reachable from $x^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}$ in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ no matter what events are communicated between the controllers because only transitions labelled by events in $\Sigma_1 \cup \Sigma_2$ can be cut in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ through the communication of events. With this in mind, the following nondeterministic automaton $\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}$ is constructed from $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$. It is assumed that $d \notin X_{x_0}^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}$.

Let $\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}} = (X^{\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}}, x_0^{\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}}, \Sigma^{\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}}, \delta^{\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}})$, where $X^{\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}} := X^H \times X^H \times X^H \times X^G \cup \{d\}, x_0^{\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}} := (x_0^H, x_0^H, x_0^H, x_0^G)$ and $\Sigma^{\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}} := \Sigma_1 \cup \Sigma_2$. The transition relation $\delta^{\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}}$ is defined as follows. Suppose there exists three states $x^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}, y^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}, z^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} \in X^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}$ and $\sigma \in \Sigma$ such that $z^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} \in X_x^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}$, $\delta^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}(z^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}, \sigma_i) = y^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}$ where $\sigma_i \in \Sigma_1 \cup \Sigma_2$. Then,

$$\delta^{\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}}(x^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}}, \sigma_i) = \begin{cases} y^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} & \text{if } d \notin X_y^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} \\ d & \text{if } d \in X_y^{\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}} \end{cases}$$

The $\tilde{\mathcal{M}}_{\Sigma_{o12},\Sigma_{o21}}$ automaton is really a colored directed graph where states are vertices, transitions are directed edges and the transition labels are the colors. This prompts another of the main contributions of this paper.

Theorem 7 *Suppose an $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$ automaton constructed from H, G, Σ_{o1}, Σ_{o2}, Σ_{c1}, Σ_{c2} and $\emptyset,\emptyset$ as the sets of communicated events is given. The language $\mathcal{L}(H)$ is co-observable with respect to $\mathcal{L}(G)$, $(\Sigma_{o1} \cup \Sigma_{o21}), (\Sigma_{o2} \cup \Sigma_{o12})$ and Σ_{c1}, Σ_{c2}. if and only if $\Sigma_1^{o21} \cup \Sigma_2^{o12}$ is a colored $x_0^{\tilde{\mathcal{M}}_{\emptyset,\emptyset}}$d-cut in the colored directed graph $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$.*

Proof: $\mathcal{L}(H)$ is co-observable with respect to $\mathcal{L}(G)$, $(\Sigma_{o1} \cup \Sigma_{o21}), (\Sigma_{o2} \cup \Sigma_{o12})$ and Σ_{c1}, Σ_{c2} if and only if $\Sigma_1^{o21} \cup \Sigma_2^{o12}$ is a colored $x_0^{\mathcal{M}_{\emptyset,\emptyset}}$*d*-cut in the colored directed graph $\mathcal{M}_{\emptyset,\emptyset}$. Therefore it is sufficient to show that $\Sigma_1^{o21} \cup \Sigma_2^{o12}$ is a colored $x_0^{\tilde{\mathcal{M}}_{\emptyset,\emptyset}}$*d*-cut in the colored directed graph $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$ if and only if $\Sigma_1^{o21} \cup \Sigma_2^{o12}$ is a colored $x_0^{\mathcal{M}_{\emptyset,\emptyset}}$*d*-cut in the colored directed graph $\mathcal{M}_{\emptyset,\emptyset}$

Define the natural projection $P_{12} : \Sigma \cup \Sigma_1 \cup \Sigma_2 \to \Sigma_1 \cup \Sigma_2$. Suppose that $\Sigma_1^{o21} \cup \Sigma_2^{o12}$ is not a colored $x_0^{\mathcal{M}_{\emptyset,\emptyset}}$*d*-cut in the colored directed graph $\mathcal{M}_{\emptyset,\emptyset}$. Then there exists a string $s \in (\Sigma \cup \Sigma_1 \cup \Sigma_2)^*$ such that $\delta^{\mathcal{M}_{\emptyset,\emptyset}}(x_0^{\mathcal{M}_{\emptyset,\emptyset}}, s) = d$. Due to the construction of $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$, $\delta^{\tilde{\mathcal{M}}_{\emptyset,\emptyset}}(x_0^{\tilde{\mathcal{M}}_{\emptyset,\emptyset}}, P_{12}(s)) = d$.

Now suppose that $\Sigma_1^{o21} \cup \Sigma_2^{o12}$ is not a colored $x_0^{\tilde{\mathcal{M}}_{\emptyset,\emptyset}}$*d*-cut in the colored directed graph $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$. Then, there exists a string $s \in \left(\Sigma_1^{o21} \cup \Sigma_2^{o12}\right)^*$ such that $\delta^{\tilde{\mathcal{M}}_{\emptyset,\emptyset}}(x_0^{\tilde{\mathcal{M}}_{\emptyset,\emptyset}}, s) = d$. Due to the construction of $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$, there exists some string $t \in \tilde{P}_{12}^{-1}(s)$ such that $\delta^{\mathcal{M}_{\emptyset,\emptyset}}(x_0^{\mathcal{M}_{\emptyset,\emptyset}}, t) = d$. ■

With the shown conversion between the graph cutting problem and the sensor selection problem the methods outlined above to approximate minimal solutions to the graph cutting problem can be used to approximate solutions to the communication selection problem. The $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$ automaton shown above can be constructed in polynomial time, but this most likely does not hold if the number of controllers is unbounded. This is due to the result in [15] that problem of deciding co-observability for systems with an unbounded number of controllers is PSPACE-complete.

10 Actuator Selection

An interesting dual of the sensor selection problem discussed above is the actuator selection problem where instead of selecting events to be observed, events are selected to be controlled in order for a controller to be used with a system in order to satisfy a specification. Given a system G, a specification automaton H, the unique minimal set of controllable events Σ_c can be found in polynomial time for an admissible controller S to exist such that $\mathcal{L}(S/G) = \mathcal{L}(H)$. To do this, a automaton $H^{\neg}$ can be constructed in polynomial time that marks $(\mathcal{L}(H)\Sigma \cap \mathcal{L}(G)) \setminus \mathcal{L}(H)$ using known methods as discussed in [2]. Therefore, $H^{\neg}$ marks a string $s\sigma$ if $s \in (\mathcal{L}(H) \cap \mathcal{L}(G))$, but $s\sigma \notin \mathcal{L}(H)$ and $s\sigma \in \mathcal{L}(G)$. For a set of strings K, let $\Sigma^{ter}(K)$ be the set of events that end all strings in K. If $\Sigma_c = \Sigma^{ter}(H^{\neg})$ and $\Sigma_{uc} = \Sigma \setminus \Sigma_c$, then

$\mathcal{L}(H)\Sigma_{uc} \cap \mathcal{L}(G) \subseteq \mathcal{L}(H)$. Therefore, a set of controllable events $\Sigma^{ter}(H^{\neg})$ can be found in polynomial time such that $\mathcal{L}(H)$ is controllable with respect to $\mathcal{L}(G)$ and $\Sigma \setminus \Sigma^{ter}(H^{\neg})$ and in the case of full observation there exists an admissible controller S such that $\mathcal{L}(S/G) = \mathcal{L}(H)$.

However, suppose instead of dealing with the actuator selection problem for exact specifications (i.e., so that $\mathcal{L}(S/G) = \mathcal{L}(H)$), suppose a minimal-cost set of control actuators should be selected to satisfy a safety specification (i.e., so that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$). A set of controllable events Σ_c for there to exist an admissible controller S such that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$ is called a sufficient actuator selection. Unfortunately there is not always a unique set of controllable events Σ_c for there to exist an admissible controller S such that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$. If there was a cost function $cost : \Sigma \rightarrow \Re^+$ such that $cost(\sigma)$ is the cost of controlling σ, then an interesting problem would be to find the minimal-cost set of controllers Σ_c^{min} for there to exist an admissible controller S such that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$.

Problem 6 Minimal-Cost Actuator-Selection: *Given G and H, find a sufficient actuator selection Σ_c^{min} such that for any other sufficient actuator selection Σ_c, $cost(\Sigma_c^{min}) \leq cost(\Sigma_c)$.*

As with the minimal sensor selection problems, the minimal-cost actuator-selection problem has an important special case where the cost of controlling an event is uniform over the set of all system events. Unfortunately Problem 6 is NP-complete and solutions to it are as difficult to approximate as the minimal-cost sensor-selection problem. This is demonstrated by converting the edge colored directed graph *st*-cut problem to a minimal-cost actuator-selection problem and vice-versa using polynomial-time many-one reductions. With these reductions it is then known that the computational difficulty results and heuristic approximation methods developed for the sensor selection problems can be directly applied to the actuator selection problems.

It is now shown how to reduce the graph cutting to a minimal-cost actuator-selection problem using a many-one polynomial time reduction. Suppose an edge colored directed graph $D = (V, A, C)$ and two vertices s, t are given. A system G and a specification H are now constructed from D. For the colors $C = \{c_1, \ldots, c_p\}$, let the event set Σ include a corresponding set of events $\{\sigma_1, \ldots, \sigma_p\}$ such that color c_i is paired with event σ_i and define $\Sigma = \{\sigma_1, \ldots, \sigma_p\}$. Suppose that for all c_i, σ_i pairs that $cost(c_i) = cost(\sigma_i)$. Also define $X^G = V \cup \{s''\}$ where s'' is a state not in V. Let $x_0^G = s$. To define the state transition function, let v_1, v_2 be any vertices except s. If $(v_1, v_2) \in A_i$, then $\delta^G(v_1, \sigma_i) = v_2$. If $(s, v_2) \in A_i$, then $\delta^G(s, \sigma_i) = v_2$ and $\delta^G(s'', \sigma_i) = v_2$. If $(v_1, s) \in A_i$, then $\delta^G(v_1, \sigma_i) = s''$. For simplicity it is assumed that $(s, s) \notin A$. Let H be a copy of G except that for any state v and event σ such that $\delta^G(v, \sigma) = t$, then $\delta^H(v, \sigma)$ is undefined. Note that G and H can be constructed in polynomial time with respect to the size of D.

Let $\Sigma_c = \{\sigma_a, \ldots, \sigma_z\}$ be a set of events such that for a controller S that always disables these events, $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$ and leaves all other events enabled. Therefore, for any string of events τ that leads from s to t in G, $\tau \in \Sigma^*\Sigma_c\Sigma^*$ because due to the fact that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$, there must be some event in τ that is disabled by S such that $\delta^G(s, \tau) = t$. Let $I = \{c_a, \ldots, c_z\}$ be the set of colors that correspond to $\Sigma_c = \{\sigma_a, \ldots, \sigma_z\}$ according to the construction of G and H. Consequently, in the directed graph D used to construct G and H, for any path of transitions from s to t, there must be an edge with a color in I.

Now suppose that $\Sigma_c = \{\sigma_a, \ldots, \sigma_z\}$ is the set of controllable events and there does not exist a (full observation) controller S such that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$. Therefore, for some string of events τ that leads from s to t in G, $\tau \notin \Sigma^*\Sigma_c\Sigma^*$ because due to the fact that $\mathcal{L}(S/G) \not\subseteq \mathcal{L}(H)$, there must be some τ without transitions labelled by events in $\Sigma_c = \{\sigma_a, \ldots, \sigma_z\}$. Let $I = \{c_a, \ldots, c_z\}$ be the set of colors that correspond to $\Sigma_c = \{\sigma_a, \ldots, \sigma_z\}$ according to the construction of G and H. Consequently, in the directed graph D used to construct G and H, there is some path from s to t, with no edges with a color in I.

Hence, for the given construction of G and H from D, a set of colors $I = \{c_a, \ldots, c_z\}$ is a colored cut for D if and only if selecting actuators for $\{\sigma_a, \ldots, \sigma_z\}$ corresponding to I allows there to be an admissible controller S such that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$. Therefore, the minimal-cost actuator-selection problem is NP-complete and any approximation algorithm for the minimal-cost actuator-selection problem can also be used with the same absolute effectiveness for the minimal-cost colored cut problem.

An example of such a system construction for converting a directed graph D to a system and specification G and H is given in Figure 5.

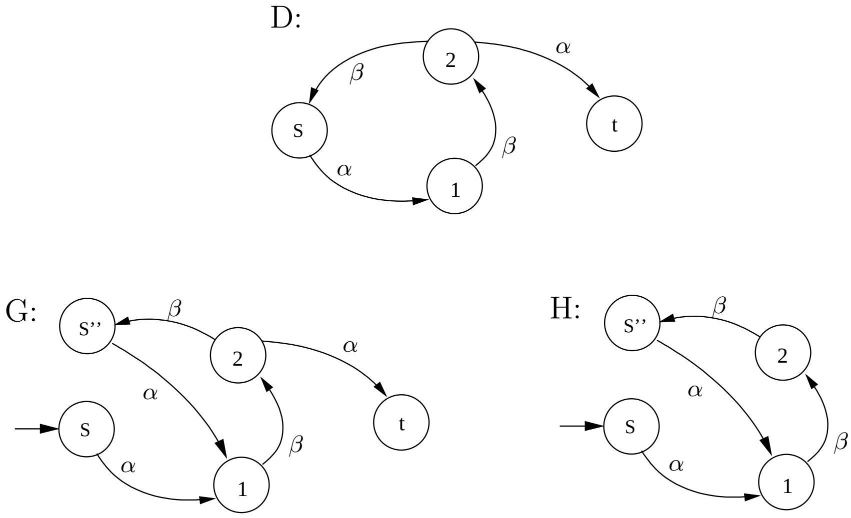

Figure 5: A directed graph D and the systems G and H constructed from it.

It is now shown how to convert the actuator-selection problem into a type of graph cutting problem. There exists an admissible controller S such that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$ if and only if for any string τ and event σ such that $\tau \in \mathcal{L}(G) \cap \mathcal{L}(H)$, $\tau\sigma \in \mathcal{L}(G)$ and $\tau\sigma \notin \mathcal{L}(H)$, there must be some event in $\tau\sigma$ that can be disabled. With this in mind, a automaton $H^{\bowtie}$ can be constructed in polynomial timing using methods discussed in [2] that marks the language $(\mathcal{L}(G) \setminus \mathcal{L}(H)) \cap (\mathcal{L}(H)\Sigma)$ and has a unique marked state. Then, there is a set $\Sigma_c \subseteq \Sigma$ such that $\mathcal{L}_m(H^{\bowtie}) \subseteq \Sigma^*\Sigma_c\Sigma^*$ if and only if there exists an admissible controller S such that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$. Therefore, if s is the initial state of $H^{\bowtie}$ and t is the unique marked state of $H^{\bowtie}$ and if $H^{\bowtie}$ is thought of as a edge colored directed graph with the set of transition labels Σ as colors, the set $\Sigma_c \subseteq \Sigma$ is a colored st-cut for $H^{\bowtie}$ if and only if the set of events are controllable Σ_c then there exists an admissible controller S such that $\mathcal{L}(S/G) \subseteq \mathcal{L}(H)$.

Now that it has been shown how to convert the actuator selection problem into an edge colored directed graph st-cut problem, the approximation methods developed above for the graph cutting problem can be effectively used to approximate solutions to the actuator selection problem.

11 Discussion

This paper has shown results related to the approximation of minimal sensor selections for centralized supervisory control synthesis. It was shown that minimal sensor selections cannot be approximated within a constant factor in polynomial time unless P=NP. It was shown how to convert the sensor selection problem into an edge colored directed graph *st*-cut problem. Several deterministic and randomized heuristic approximation methods for this directed graph problem were shown and a conversion of the directed graph problem to an integer programming problem was given. An open communicating controller problem was also discussed and it was shown how to convert this minimal communication decentralized control problem into an edge colored directed graph *st*-cut problem. Therefore the methods discussed in this paper for dealing with the edge colored directed graph *st*-cut problem can be used to solve the minimal communication problem.

12 Acknowledgment

This paper reports on work commenced while the first author was temporarily employed at CWI and cooperating with the second author. Financial support in part for the investigation was made available by the European Commission through the project Control and Computation (IST-2001-33520) of the Information Society Technologies Program.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kahn, A. Marchetti-Spaccamela, and M. Protasi. *Complexity and Approximation Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag, Berlin Heidelberg, 1999.

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

[3] H. Cho. Designing observation functions in supervisory control. In *Proc. Korean Automatic Control Conference*, pages 523–528, 1990.

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

[5] A. Haji-Valizadeh and K.A. Loparo. Minimizing the cardinality of an events set for supervisors of discrete-event dynamical systems. *IEEE Trans. Auto. Contr.*, 41(11):1579–1593, November 1996.

[6] P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to Probability Theory*. Houghton Mifflin Co., 1971.

[7] S. Jiang and R. Kumar. Supervisory control of discrete event systems with CTL* temporal logic specifications. Preprint.

[8] S. Jiang, R. Kumar, and H.E. Garcia. Optimal sensor selection for discrete-event systems with partial observation. *IEEE Trans. Auto. Contr.*, 48(3):369–381, 2003.

[9] S. Khuller, G. Kortsarz, and K. Rohloff. Approximating the minimal sensor selection for supervisory control. In *Proc. 7th Workshop on Discrete Event Systems*, Reims, France, September 2004. To appear.

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

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

[12] C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.

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

[14] P.J. Ramadge and W.M. Wonham. The control of discrete-event systems. *Proc. IEEE*, 77(1):81–98, 1989.

[15] K. Rohloff, T.-S. Yoo, and S. Lafortune. Deciding co-observability is PSPACE-complete. *IEEE Trans. Auto. Contr.*, 48(11):1995–1999, 2003.

[16] K. Rudie and J.C. Willems. The computational complexity of decentralized discrete-event control problems. *IEEE Trans. Auto. Contr.*, 40(7):1313–1318, 1995.

[17] K. Rudie and W.M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Trans. Auto. Contr.*, 37(11):1692–1708, November 1992.

[18] R. Sengupta and S. Lafortune. A graph-theoretic optimal control problem for terminating discrete-event processes. *Journal of Discrete Event Dynamical Systems: Theory and Applications*, 2:139–172, 1992.

[19] R. Sengupta and S. Lafortune. An optimal control theory for discrete event systems. *SIAM Journal on Control and Optimization*, 36(2):488–541, 1998.

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

[21] Jan H. van Schuppen. Decentralized control with communication between controllers. In Vincent D. Blondel and Alexandre Megretski, editors, *Unsolved problems in mathematical systems and control theory*, pages 144–150. Princeton University Press, Princeton, 2004.

[22] V. Vazirani. *Approximation Algorithms.* Springer-Verlag, Berlin Heidelberg, 2001.

[23] W. M. Wonham and P. J. Ramadge. Modular supervisory control of discrete event systems. *Maths. of Control, Signals and Systems*, 1(1):13–30, 1988.

[24] T.-S. Yoo and S. Lafortune. On the computational complexity of some problems arising in partially-observed discrete-event systems. In *Proc. of the American Control Conference*, pages 307–312, June 2001.

[25] T.-S. Yoo and S. Lafortune. NP-completeness of sensor selection problems arising in partially-observed discrete-event systems. *IEEE Trans. Auto. Contr.*, 47(9):1495–1499, 2002.

[26] S. Young and V. K. Garg. Optimal sensor and actuator choices for discrete event systems. In *Proc. 31st Allerton Conference on Communication, Control, and Computing*, October 1994.