Approximation Algorithms, Sensor Selections and Information Sharing

Kurt R. Rohloff[1] and Jan H. van Schuppen[2]

[1] Coordinated Science Laboratory
The University of Illinois at Urbana-Champaign
1308 West Main St., Urbana, IL 61801, USA
krohloff@control.csl.uiuc.edu
black.csl.uiuc.edu/∼krohloff

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

Abstract. This paper discusses recent results related to the approximation of optimal sensor selection and information sharing policies for the supervisory control of discrete-event systems. Although the framework of supervisory control is used, the results and methods contained herein are also applicable to fault diagnosis systems. It is shown how the problems of computing an optimal sensor selection is related to the problem of computing edge-colored direct graph *st*-cuts. Heuristic approaches to the approximation of optimal solutions to the sensor selection problem are proposed based on graph cutting and integer programming methods. It discussed how the graph cutting conversion method can be used for other problems of partial information in supervisory control such as a communication selection problem for decentralized control systems.

1 Introduction

When a controller operates on a system so that the behavior of the controlled system matches some specification, the controller may not need sensors to observe all behavior in the system. That is, there may be several sets of sensors that could be selected for the controller to use that would be sufficient for the controller to match the specification. Therefore, if there is a cost associated with allowing a controller to use a sensor, then for reasons of economy or simplicity it may be required that the controller uses a set of sensors with the lowest cost possible. This paper surveys recent results in [7, 12, 13] related to information selection problems in supervisory control. This paper uses the framework of supervisory control theory and discrete-event systems introduced from the seminal works [8, 10, 11].

Unfortunately, even for the specialized case of uniform sensor cost, the optimal sensor selection problem for supervisory control is NP-complete ([19]). This means that there is most likely no algorithm that runs in polynomial time and always calculates the minimal cost sensor selection. Fortunately, effective polynomial time approximation algorithms exist for many real-world NP-complete optimization problems ([1]). With this in mind, an approximation of the minimal sensor selection may commonly be sufficient and acceptable for practical use. Therefore, an interesting compromise to designing algorithms to find the minimum cost sensor selection would be to develop methods to approximate the minimal cost sensor selection. Hopefully some bounds could be placed on the closeness of the approximations found this way as not all NP-complete problems have equally effective polynomial-time approximation methods. However, there has been little investigation into the calculation of provably good approximate solutions to many computationally difficult supervisory control problems. Therefore, this paper explores the problem of approximating solutions to the sensor cost minimization problem.

Variations of the sensor selection problem using frameworks similar to the one used in this paper have been investigated in [3, 4, 6, 19, 20]. 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 [4] for achieving both observability and normality for a problem setting very similar to that discussed here. An exponential-time algorithm is shown in [4] for giving an optimal observable set. An algorithm is given in [20] for optimizing the sensor selection set in exponential time along with a polynomial time algorithm for finding exactly one locally minimal sensor selection. An optimal sensor selection problem is also discussed in [6], except the observation function is different from the one assumed in this paper.

This paper presents a method to convert a supervisory control sensor selection problem into a type of edge colored graph cutting problem. This approach is very general and can be applied to a wide range of partial information discrete-event system problem areas such as fault diagnosis, decentralized control and so on. It is shown in this paper how the graph cutting methodology can also be used for a communication selection problem for decentralized supervisory control systems. A variation of this problem is also discussed by [18] where asymmetric communication is assumed.

In the next section the necessary background information from supervisory control is given as presented in [8, 10, 11]. A more indepth review of this material is given in [2]. The problem statement of the sensor selection problem is formulated in Section 3. The sensor selection problem is related to a type of directed graph *st*-cut problem in Section 4, and inapproximability results for the sensor selection and graph cutting problems are shown in Section 5. Section 6 shows several heuristic approaches to the sensor selection problem based on graph cutting and integer programming. It is shown how the reduction method to convert the sensor selection problem into the graph cutting problem can also be used for a decentralized control communication selection problem in Section 7. The paper closes with a brief discussion of the results in Section 8.

2 Notational Review

In the supervisory control framework systems and specifications are respectively modeled as the deterministic automata $G = (X^G, x_0^G, \Sigma, \delta^G)$ and $H = (X^H, x_0^H, \Sigma, \delta^H)$. The notation $x \overset{s}{\mapsto}_B y$ is also sometimes used in this paper to denote that according to the transition rules of a possibly nondeterministic automaton B, there is a path of transitions from x to y labeled by the string s. As outlined above, the supervisory control systems discussed in this paper have a set of sensors to observe a set of system events $\Sigma_o \subseteq \Sigma$ with each sensor assigned to deterministically observe all occurrences of exactly one event. Given a controller S and a system G, the composed system of S controlling G is denoted as the controlled system S/G. The generated behavior of the controlled system S/G is said to match the generated behavior of a specification H if $\mathcal{L}(S/G) = \mathcal{L}(H)$. For mathematical simplicity it is always assumed in this paper that $\mathcal{L}(H)$ is controllable with respect to $\mathcal{L}(G)$.

For a given set of observable events $\Sigma_o \subseteq \Sigma$, a natural projection operation $P : \Sigma \rightarrow \Sigma_o$ is used to model a controller's observations of system behavior. 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 the observation projection $P(s)$ to estimate the current system state and determine its control action. See Figure 1 for a schematic of a system G that is controlled by the controller S to match a specification H. In this figure, a string of behavior s is generated by G and $P(s)$ is observed by the controller. After observing $P(s)$, the controller enforces control action $S(P(s))$ on the behavior of G.

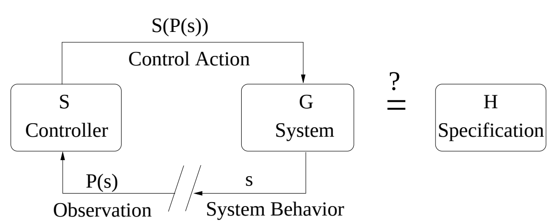

Fig. 1. Schematic of a supervisory control system

An important property for controller existence is now presented.

Definition 1. *[8] Consider the languages K and M such that $M = \overline{M}$ and the sets of controllable, Σ_c, and observable Σ_o events. The language K is* observable *with respect to M, $P(\cdot)$ and Σ_c if for all $t \in \overline{K}$ and for all $\sigma \in \Sigma_c$,*

$$\left[\left(t\sigma \notin \overline{K}\right) \wedge (t\sigma \in M)\right] \Rightarrow \left(P^{-1}\left[P(t)\right]\sigma \cap \overline{K} = \emptyset\right). \tag{1}$$

The concept of observability captures the notion that for a set of generated system behaviors M and a set of marked specification behaviors K, that for every possible string of behavior $t\sigma \in M$ such that σ is controllable, t is legal, but $t\sigma$ is not, then there must be no control conflict associated with a controller's estimate of disabling σ. That is, $P^{-1}[P(t)]\sigma$ must not contain a string $t'\sigma$ that is legal but indistinguishable from $t\sigma$ with respect to the sensor selection Σ_o. With the above assumption of controllability, for a finite state automaton system G, a finite state automaton specification H such that $\mathcal{L}(H) \subseteq \mathcal{L}(G)$, observability is necessary and sufficient for there to exist a controller S such that $\mathcal{L}(S/G) = \mathcal{L}(H)$ [8].

3 The Sensor Selection Problem

Before the sensor selection problems are introduced some important concepts are defined.

Definition 2. *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.*

When given a system G, a specification H and a set of controllable events Σ_c, it may be desired to find the lowest cardinality sufficient sensor selection Σ_o in order to ensure that there exists a controller S such that $\mathcal{L}(S/G) = \mathcal{L}(H)$. This prompts the formal definition of the minimal cardinality sufficient sensor selection problem.

Problem 1. 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, $|\Sigma_o^{min}| \leq |\Sigma_o|$.

A simple example of the sensor selection problem is now shown.

Example 1. Consider the system and specification seen in Figure 2. Suppose that $\Sigma_c = \{\alpha\}$. There are several sufficient sensor selections for this specification with respect to the given system: $\{\alpha\}, \{\beta,\gamma\}, \{\gamma,\lambda\}, \{\beta,\lambda\}$. The minimal cardinality sufficient sensor selection is $\{\alpha\}$.

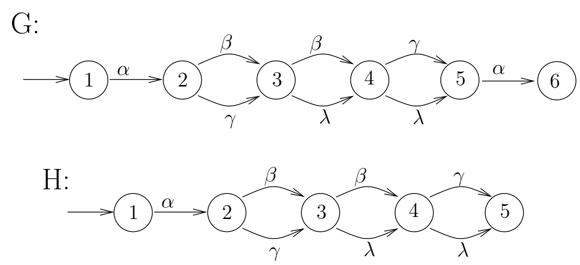

Fig. 2. The system G and specification H of Example 1.

There is a generalized version of Problem 1 where the sensors might have non-uniform cost and the problem is to minimize the total cost of sensor usage.

Because of the NP-completeness of Problem 1, the minimal cardinality sensor selection cannot always be found in a computationally efficient manner [19]. This result therefore shows that Problem 1 is similarly computationally difficult. However, despite the computational difficulties of the sensor selection problems, 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, a considerable subset of NP-complete minimization problems have fairly accurate polynomial time approximation algorithms [1, 17]. This means sufficient and approximate solutions can be found for many computationally difficult problems in a reasonable amount of time. However, to the best of the authors' knowledge, there has been no investigation into the approximation difficulty of sensor selection problems.

4 The Graph Cutting Problem

It was mentioned above that the observability of languages marked by finite state automata can be tested using a nondeterministic automata construction introduced in [16]. This construction can be used to convert sensor selection problems into a special type of graph cutting problem called an "edge colored directed graph *st*-cut problem".

For the edge colored directed graph *st*-cut problem, assume an edge colored directed graph $D = (V, A, C)$ is given where V is a set of vertices, $A \subseteq V \times V$ are directed edges and $C = \{c_1, \dots, c_p\}$ is a set of colors. Each edge is assigned a color in C. Let A_i be the set of 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, I is a colored *st*-cut if $(V, (A \setminus A_I), C)$ has no path from s to t. This prompts the definition of the Minimal Cardinality Colored Cut Problem seen below.

Problem 2. Minimal Cardinality Colored Cut: For an edge colored directed graph $D = (V, A, C)$, two vertices $s, t \in V$, find a colored *st*-cut $I^{min} \subseteq C$ such that for any other colored *st*-cut $I \subseteq C$, $|I^{min}| \leq |I|$.

The directed graph in Figure 3 is an example of an edge colored directed graph where the edges are assigned colors $\{\alpha, \beta, \gamma, \lambda\}$. Note that $I = \{\beta, \gamma\}$ is one of several colored *st*-cuts for this graph.

Similar to the Minimal Cardinality Sensor Selection Problem, the Minimal Cardinality Colored Cut problem has a generalized version for situations when sensor costs are non-uniform. In the following subsections it is shown how to convert sensor selection problems into colored cut problems. This implies that approximation methods for one problem can be used for the other.

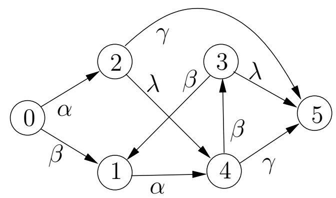

Fig. 3. A directed graph with colored cut $I = \{\beta, \gamma\}$.

4.1 Converting Sensor Selection Problems into Colored Cut Problems

A construction is now shown to convert an instance of Problem 1 into an instance of Problem 2. 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 and it is desired to test if $\mathcal{L}(H)$ is observable with respect to $\mathcal{L}(G)$, Σ_o and Σ_c. This is done by constructing an automaton $\mathcal{M}_{\Sigma_o} = (X^{\mathcal{M}_{\Sigma_o}}, x_0^{\mathcal{M}_{\Sigma_o}}, \Sigma^{\mathcal{M}_{\Sigma_o}}, \delta^{\mathcal{M}_{\Sigma_o}})$ that is a modification of the $\mathcal{M}$ automaton method for testing observability and co-observability in [14, 16]. The $\mathcal{M}_{\Sigma_o}$ automaton is effectively a nondeterministic simulation of estimates an observer may make of unobservable system behavior with respect to a specification based on imperfect predictions of occurrences of unobservable events $(\Sigma \setminus \Sigma_o)$ in the system.

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:

$$\begin{aligned} X^{\mathcal{M}_{\Sigma_o}} &:= X^H \times X^H \times X^G \cup \{d\}, \\ x_0^{\mathcal{M}_{\Sigma_o}} &:= (x_0^H, x_0^H, x_0^G), \\ \Sigma^{\mathcal{M}_{\Sigma_o}} &:= \Sigma \cup \Sigma'. \end{aligned}$$

Suppose a string of events s has been simulated to occur in the system G by $\mathcal{M}_{\Sigma_o}$ and the simulation is at state $(x_1, x_2, x_3) \in X^{\mathcal{M}_{\Sigma_o}}$. State x_3 represents the true state of the system G and x_2 represents the corresponding true state of the specification H after s has occurred. States x_2 and x_3 always update simultaneously. However, as was stated above, the observer attempts to predict the occurrence of system events and the state x_1 represents a possible observer estimate of the specification state based on imperfect predictions of the simulated system behavior due to the observation of $P(s)$.

At state (x_1, x_2, x_3) of the simulation, if an event σ is correctly predicted by the observer in the simulation, there is a transition from (x_1, x_2, x_3) labeled by σ where all of the component states of (x_1, x_2, x_3) update on the occurrence of σ according to the transition rules of H, H and G respectively. A correct prediction may occur for either observable or unobservable events.

However, if an event σ occurs in the system that is not predicted correctly by the observer in the simulation, there is a transition from (x_1, x_2, x_3) labeled by σ' where the x_2, x_3 component states of (x_1, x_2, x_3) update on the occurrence of σ according to the transition rules of H and G respectively. Similarly, if an event

σ does not occur in the system but is incorrectly predicted to have occurred by the observer in the simulation, then there is a transition from (x_1, x_2, x_3) labeled by σ' where the x_1 component state of (x_1, x_2, x_3) updates on the occurrence of σ according to the transition rules of H. Therefore, because unobservable event occurrences can not be guaranteed to be perfectly predicted, the $\mathcal{M}_{\Sigma_o}$ simulation is nondeterministic.

If the $\mathcal{M}_{\Sigma_o}$ simulation ever reaches a composed state where the observer believes the occurrence of a controllable event is allowed by the specification due to the properties of state x_1, but in reality it is not due to specification state x_2, yet still possible due to system state x_3, then there is a control conflict as the observer would believe a controllable event is illegal when in reality it is not. This possibility is captured by the $(*)$ condition such that if the simulation could reach a state (x_1, x_2, x_3) where $(*)$ holds, then illegal controllable behavior could occur in the system without an observer being able to resolve the control conflict.

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

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

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

For $\sigma \in \Sigma$,

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

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 $\delta^G(x, \sigma)!$. Note that because $\mathcal{M}_{\Sigma_o}$ is nondeterministic, $\delta^{\mathcal{M}_{\Sigma_o}}(\cdot, \cdot)$ possibly returns a set of states. The $\mathcal{M}_{\Sigma_o}$ automaton construction prompts the following proposition which follows from the results in [16].

Proposition 1. *[16] The state d is not 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.*

An example is now given of an $\mathcal{M}_{\Sigma_o}$ automaton construction.

Example 2. Recall the system and specification shown in Example 1. The $\mathcal{M}_\emptyset$ automaton constructed for this system and specification can be seen in Figure 4.

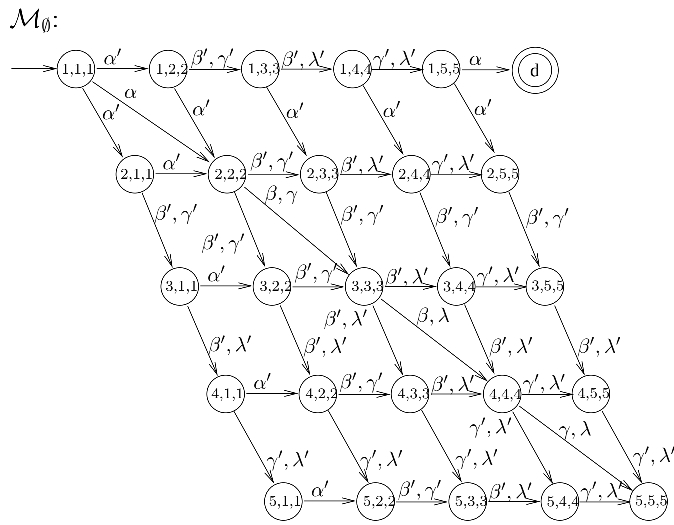

Fig. 4. The $\mathcal{M}_\emptyset$ automaton constructed from G and H of Example 1.

The construction of $\mathcal{M}_\emptyset$ prompts the following theorem.

Theorem 1. *$\mathcal{L}(H)$ 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 for $\mathcal{M}_\emptyset$.*

The $\mathcal{M}_\emptyset$ cut problem is not in the same form as in Problem 2 as Σ labeled transitions can never be cut in the $\mathcal{M}_\emptyset$ automaton of Theorem 1 by making events observable. To counter this difference the $\tilde{\mathcal{M}}_{\Sigma_o}$ construction below is used which is a copy of $\mathcal{M}_{\Sigma_o}$ except that some states are combined to hide Σ transitions in $\mathcal{M}_\emptyset$.

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

$$X_x^{\mathcal{M}_{\Sigma_o}} = \left\{ y^{\mathcal{M}_{\Sigma_o}} \middle| \exists t \in \Sigma^* \text{ such that } x^{\mathcal{M}_{\Sigma_o}} \overset{t}{\mapsto}_{\mathcal{M}_{\Sigma_o}} y^{\mathcal{M}_{\Sigma_o}} \right\}.$$

Notice the x subscript on $X_x^{\mathcal{M}_{\Sigma_o}}$. The set $X_x^{\mathcal{M}_{\Sigma_o}}$ represents all states that could be reached from $x^{\mathcal{M}_{\Sigma_o}}$ in $\mathcal{M}_{\Sigma_o}$ if and only if Σ transitions were allowed. These are the same transitions in $\mathcal{M}_{\Sigma_o}$ that could not be cut by making more events observable. Due to this the states in $X_x^{\mathcal{M}_{\Sigma_o}}$ would be reachable from $x^{\mathcal{M}_{\Sigma_o}}$ according to the transition rules of $\mathcal{M}_{\Sigma_o}$ no matter what events are made observable.

With this in mind, the following nondeterministic automaton $\tilde{\mathcal{M}}_{\Sigma_o}$ is constructed from $\mathcal{M}_{\Sigma_o}$ such that if there are two states $x^{\mathcal{M}_{\Sigma_o}}, y^{\mathcal{M}_{\Sigma_o}}$ and some string of transitions labeled by $s\sigma' \in \Sigma^*\Sigma'$ such that according to the transition rules of $\mathcal{M}_{\Sigma_o}$, $x^{\mathcal{M}_{\Sigma_o}} \overset{s\sigma'}{\mapsto}_{\mathcal{M}_{\Sigma_o}} y^{\mathcal{M}_{\Sigma_o}}$, then according to the transition rules of $\tilde{\mathcal{M}}_{\Sigma_o}$, $x^{\mathcal{M}_{\Sigma_o}} \overset{\sigma}{\mapsto}_{\tilde{\mathcal{M}}_{\Sigma_o}} y^{\mathcal{M}_{\Sigma_o}}$ where $x^{\mathcal{M}_{\Sigma_o}}$ and $y^{\mathcal{M}_{\Sigma_o}}$ are states in both $\mathcal{M}_{\Sigma_o}$ and

$\tilde{\mathcal{M}}_{\Sigma_o}$, but with different outgoing state transitions in the two automata. This construction effectively condenses all $\mathcal{M}_{\Sigma_o}$ states reachable by Σ transitions and replaces the remaining Σ' labels with the corresponding Σ labels. It is assumed that $d \notin X_{x_0}^{\mathcal{M}_{\Sigma_o}}$ because if $d \in X_{x_0}^{\mathcal{M}_{\Sigma_o}}$, then even if a controller could observe the occurrences of all events (i.e., $\Sigma_o = \Sigma$), the system could not be made observable in any case.

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

$$\begin{aligned} X^{\tilde{\mathcal{M}}_{\Sigma_o}} &:= X^H \times X^H \times X^G \cup \{d\}, \\ x_0^{\tilde{\mathcal{M}}_{\Sigma_o}} &:= (x_0^H, x_0^H, x_0^G), \\ \Sigma^{\tilde{\mathcal{M}}_{\Sigma_o}} &:= \Sigma. \end{aligned}$$

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

Suppose there exists 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 $\sigma \in \Sigma$ such that $z^{\mathcal{M}_{\Sigma_o}} \in X_x^{\mathcal{M}_{\Sigma_o}}$ and $z^{\mathcal{M}_{\Sigma_o}} \overset{\sigma'}{\mapsto}_{\mathcal{M}_{\Sigma_o}} y^{\mathcal{M}_{\Sigma_o}}$. Then,

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

An example is now given of an $\tilde{\mathcal{M}}_{\Sigma_o}$ automaton construction.

Example 3. Recall the system and specification shown in Example 1 and the resulting $\mathcal{M}_{\Sigma_o}$ automaton seen in Figure 4. The corresponding $\tilde{\mathcal{M}}_\emptyset$ automaton constructed for this system and specification with $\Sigma_c = \{\alpha\}$ can be seen in Figure 5.

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 chapter.

Theorem 2. *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 $\tilde{\mathcal{M}}_\emptyset$.*

5 Inapproximability Results

To the knowledge of the authors, the edge colored directed graph cutting problem has not been explored in the standard literature (from graph theory or computer science). Unfortunately, although many other types of graph cutting problems are computationally simple, it is shown here that solutions to Problem 2 are most likely difficult to approximate. Because of the above results, solutions to the sensor selection problem are similarly difficult to approximate.

Corollary 1. *[7] The minimal cardinality colored cut problem admits no $2^{(\log n)^{1-\epsilon}}$-approximation, for any $\epsilon > 0$ unless $NP \subseteq DTIME(n^{polylog\ n})$.*

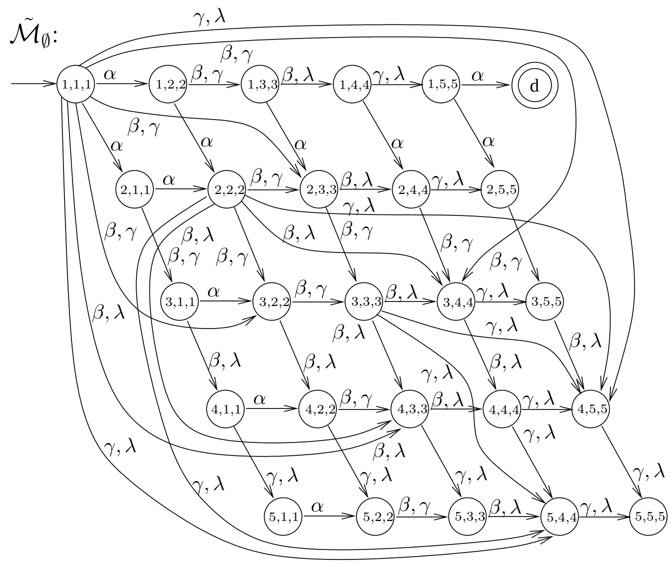

Fig. 5. The $\tilde{\mathcal{M}}_\emptyset$ automaton constructed from G and H of Example 1.

It follows that if solutions to any of the colored cut or sensor selection problems discussed in this paper can be approximated with better than a $2^{(\log n)^{1-\epsilon}}$-approximation, then a method for solving NP-complete problems in quasipolynomial time has been found. This lower bound is generally considered to be a very poor lower bound in the computer science community. Indeed, as ϵ approaches 0, then $2^{(\log n)^{1-\epsilon}}$ approaches n. So far, for all problems admitting this bound, the best existing approximation known is n^ϵ for some constant $\epsilon > 0$. Such an approximation for a special case of the graph-cutting problem is shown in [7].

6 Heuristic Approximation Methods

Heuristic algorithms are now shown to approximate solutions to the sensor selection problem. These algorithms are based on graph cuttings of $\tilde{\mathcal{M}}_\emptyset$. After constructing $\tilde{\mathcal{M}}_\emptyset$, events are iteratively assigned to be observed by the controller in order to cut all paths from $\tilde{x}_0^{\mathcal{M}_\emptyset}$ to d in $\tilde{\mathcal{M}}_\emptyset$. The first algorithm, called *DetGreedyAprx*, is a deterministic greedy algorithm that uses a utility function to identify and iteratively cut transitions associated with event labels in $\tilde{\mathcal{M}}_{\Sigma_o}$.

6.1 A Deterministic Greedy Algorithm

Starting with a trim version of $\tilde{\mathcal{M}}_\emptyset$, suppose it is desirable to find the "probability" $\mathcal{P}(\sigma, \tilde{\mathcal{M}}_\emptyset)$ that a "randomly" selected path from $\tilde{x}_o^\emptyset$ to d contains an edge labeled by $\Psi_1(\sigma) \in \Sigma_1$ or $\Psi_2(\sigma) \in \Sigma_2$ corresponding to $\sigma \in \Sigma$. By selecting an event which occurs with a relatively high probability on paths from $\tilde{x}_0^{\mathcal{M}_\emptyset}$ to d, then that event should have a high utility of being observed by the controller. The terms "probability" and "randomly" are used here 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.

In order to remove all paths to d in $\tilde{\mathcal{M}}_\emptyset$, it would be desirable to first cut transitions associated with events with the highest utility $\mathcal{P}(\sigma, \tilde{\mathcal{M}}_\emptyset)$. After an event is selected to be cut in $\tilde{\mathcal{M}}_\emptyset$, the utility function $\mathcal{P}(\cdot,\cdot)$ is updated to reflect the changes in the sensor selections and another event is then chosen to be observed. This procedure, seen in Algorithm DetGreedyAprx, is iterated until there are no paths to d.

As DetGreedyAprx iteratively chooses events to be selected for observation, the $\tilde{\mathcal{M}}^T_{\Sigma_o}$ automaton is continually trimmed. Therefore, as Σ_o is updated, the next $\tilde{\mathcal{M}}^T_{\Sigma_o}$ can be calculated in polynomial time. The relative probabilities $\{\rho^1, \dots, \rho^k\}$ associated with the events selected for observation, $\{\sigma^1, \dots, \sigma^k\}$, are stored for later analysis of the accuracy of the found approximation $|\Sigma_o|$.

Algorithm 1 Deterministic Greedy Approximation Algorithm (DetGreedyAprx)

Input: $\tilde{\mathcal{M}}_\emptyset$;
$\Sigma_o \leftarrow \emptyset,\ i \leftarrow 1$;
$\tilde{\mathcal{M}}^T_{\Sigma_o} \leftarrow Trim(\tilde{\mathcal{M}}_{\Sigma_o})$;
While d *reachable in* $\tilde{\mathcal{M}}^T_{\Sigma_o}$;
{
$\sigma^i \leftarrow \arg\max_{\sigma \in (\Sigma_o)} \left(\mathcal{P}\left(\sigma, \tilde{\mathcal{M}}^T_{\Sigma_o}\right)\right)$;
$\rho^i \leftarrow \mathcal{P}\left(\sigma^i, \tilde{\mathcal{M}}^T_{\Sigma_o}\right)$;
$\Sigma_o \leftarrow \Sigma_o \cup \{\sigma^i\}$;
$k \leftarrow i$;
$i \leftarrow i + 1$;
Reconstruct $\tilde{\mathcal{M}}_{\Sigma_o}$;
$\tilde{\mathcal{M}}^T_{\Sigma_o} \leftarrow Trim(\tilde{\mathcal{M}}_{\Sigma_o})$;
}
Return Σ_o;

It remains to be discussed how $\mathcal{P}\left(\sigma, \tilde{\mathcal{M}}^T_{\Sigma_o}\right)$ is calculated. At each state x in $\tilde{\mathcal{M}}^T_{\Sigma_o}$, suppose there are κ_x output transitions. $\tilde{\mathcal{M}}^T_{\Sigma_o}$ is converted into a stochastic automaton by assigning a probability of occurrence $\frac{1}{\kappa_x}$ to each output transition of x. Let $\mathcal{P}(\sigma, \tilde{\mathcal{M}}^T_{\Sigma_o})$ denote the probability that a random walk in $\tilde{\mathcal{M}}^T_{\Sigma_o}$ traverses a σ transition from the initial state to d. It should be noted that this probability can be computed in polynomial time using standard methods from [5]. Therefore, this approximation algorithm runs in polynomial time.

Algorithm DetGreedyAprx is now analyzed to obtain a bounds on accuracy of the approximation returned by the algorithm. The set $\Sigma_o^{\min_i}$ denotes the minimum cardinality sensor selection sets that could be chosen at iteration i given that events in Σ_o^i are already observed. Naturally, $\Sigma_o^{\min_1} = \Sigma_o^{\min}$.

Lemma 1. *In DetGreedyAprx, on the ith iteration,*

$$\frac{1}{\mathcal{P}\left(\sigma^i, \tilde{\mathcal{M}}_{\Sigma_o^i}\right)} \leq |\Sigma_o^{\min_i}|$$

Lemma 1 can be used to show the following result on the closeness of the approximation returned by DetGreedyAprx.

Theorem 3. *For the set Σ_o returned by DetGreedyAprx and a minimal sensor selection $\Sigma_o^{\min}$,*

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

where $\{\rho^1, \dots \rho^k\}$ are the iterative probabilities stored during the operation of DetGreedyAprx.

Because of Theorem 3, a bound on the closeness of the approximation returned by DetGreedyAprx 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 3 is now shown.

Theorem 4. *From a set $\{\rho^1, \dots, \rho^k\}$ from a running of DetGreedyAprx,*

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

Although Theorem 4 puts a lower bound on the guarantee of the approximation ratio shown in Theorem 3, DetGreedyAprx may return a solution with an approximation ratio better than H_k.

A randomized version of DetGreedyAprx is also shown in [13] where on each iteration of the algorithm, the event selected to be made observable is chosen with a weight random distribution. This algorithm returns a different solution every time it is run so that it can be run multiple times to boost the probability that a good approximate solution will be found.

6.2 Integer Programming

It is now shown that another approach to approximating the minimal cost sensor selection is to use integer programming based methods. The integer programming problem is a general optimization problem from the field of the combinatorial optimization that has been well explored in the literature [9]. This section discusses how to convert the minimal cost colored cut problem to an integer programming problem. Therefore, using the reduction methods discussed above to convert the sensor selection problem into a colored cut problem, integer programming methods can also be used for the sensor selection problem. First the integer programming problem is introduced.

Problem 3. The Integer Programming Problem: Given a z element row vector $\boldsymbol{C}$, a $y \times z$ matrix $\boldsymbol{A}$ and a y element column vector $\boldsymbol{B}$, find a z element column vector $\boldsymbol{x} \in \{0,1\}^z$ that minimizes $\boldsymbol{Cx}$ subject to $\boldsymbol{Ax} \geq \boldsymbol{B}$.

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 [9, 17]. 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, because of the problem conversion methods discussed in this section, already developed and mature methods for the well understood integer programming can be used to find solutions to the sensor selection problem.

6.3 Problem Conversion

It is now shown how to convert the minimal cost colored cut problem into an integer programming problem. Suppose an edge-colored directed graph $D = (V, A, C)$ is given with a cost function $cost : C \rightarrow \Re^+ \cup \{0\}$. Suppose $V = \{v_1, v_2, \ldots, v_{n_V}\}$, $A = \{a_1, a_2, \ldots, a_{n_A}\}$ and $C = \{c_1, c_2, \ldots, c_{n_C}\}$. Without loss of generality assume that for the graph cutting problem the task is to find the minimal cost colored $v_1 v_{n_V}$-cut $I \subseteq C$.

For the colors C, let there be a set of boolean variables $\{b_{c_1}, b_{c_2}, \ldots, b_{c_{n_C}}\}$ and for the set of vertices V, let there be another set of boolean variables $\{b_{v_1}, b_{v_2}, \ldots, b_{v_{n_V}}\}$. For a cut I, values can be assigned to $\{b_{c_1}, b_{c_2}, \ldots, b_{c_{n_C}}\}$ such that $b_{c_i} = 1$ if and only if $c_i \in I$. Note that by definition, $\sum_{i=1}^{n_C} b_{c_i} cost(c_i)$ is the cost of the colored cut I. For a cut I, values are assigned to $\{b_{v_1}, b_{v_2}, \ldots, b_{v_{n_V}}\}$ such that $b_{v_1} = 1$, and for all $i, j \in \{1, \ldots, n_V\}$ and $k \in \{1, \ldots, n_C\}$ such that if $(v_i, v_j) \in A_{c_k}$, then $(b_{v_i} = 1) \wedge (b_{c_k} = 0) \Rightarrow (b_{v_j} = 1)$. These constraints on the assignment of values to $\{b_{v_1}, b_{v_2}, \ldots, b_{v_{n_V}}\}$ can be thought of as a form of a reachability condition. That is, for a vertex v_i, if $b_{v_i} = 1$, then for all vertices v_j that are reachable from v_i along edges in $A \setminus A_I$, it must hold that $b_{v_j} = 1$. Therefore, because $b_{v_1} = 1$, these constraints imply that if v_j is reachable from v_1 along edges in $A \setminus A_I$, then $b_{v_j} = 1$.

Note that if it is necessary to assign $b_{v_{n_V}} = 1$ with the above constraints, then I is not a $v_1 v_{n_V}$-cut in D. However, if it is possible to assign $b_{v_{n_V}} = 0$ with the above constraints, then I is a $v_1 v_{n_V}$-cut in D. This is demonstrated in the following lemma. If the constraint is added that $b_{v_{n_V}} = 0$, then all vertices v_k which can reach v_{n_V} along edges in $A \setminus A_I$, it must hold that $b_{v_k} = 0$. Note that for any vertex v_j not reachable from v_1 with the colored cut I, or which cannot reach v_{n_V} with the colored cut I, $b_{v_1} = 1$ and $b_{v_{n_V}} = 0$, then the corresponding boolean variable b_{v_j} can be assigned arbitrarily.

Note that because $\{b_{c_1}, b_{c_2}, \ldots, b_{c_{n_C}}\}$ and $\{b_{v_1}, b_{v_2}, \ldots, b_{v_{n_V}}\}$ are boolean variables, $(b_{v_i} = 1) \wedge (b_{c_k} = 0) \Rightarrow (b_{v_j} = 1)$ holds if and only if $-b_{v_i} + b_{v_j} + b_{c_k} \geq 0$, $b_{v_1} = 1$ holds if and only if $b_{v_1} \geq 1$ and $b_{v_{n_V}} = 0$ holds if and only if $-b_{v_{n_V}} \geq 0$. It is now shown how to construct the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ and a vector $\boldsymbol{x}$ such that $\boldsymbol{Ax} \geq \boldsymbol{B}$ if and only if the linear inequalities $b_{v_1} \geq 1$, $-b_{v_{n_V}} \geq 0$ and for all $(v_i, v_j) \in A$, if $(v_i, v_j) \in A_{c_k}$, then $b_{c_k} - b_{v_i} + b_{v_j} \geq 0$ are satisfied.

First, let $\boldsymbol{x}$ be the boolean $(n_C + n_V)$-element column vector defined as follows:

$$\boldsymbol{x}^T = \left[b_{c_1}, b_{c_2}, \cdots, b_{c_{n_C}}, b_{v_1}, b_{v_2}, \cdots, b_{v_{n_V}}, \right].$$

To encode the constraints that for all $(v_i, v_j) \in A$, if $(v_i, v_j) \in A_{c_k}$, then $b_{c_k} - b_{v_i} + b_{v_j} \geq 0$, suppose that the edges in A are given an arbitrary ordering and suppose without loss of generality that the mth edge corresponds to the constraint that $b_{c_{k_m}} - b_{v_{i_m}} + b_{v_{j_m}} \geq 0$. Note that the k_mth entry in $\boldsymbol{x}$ is $b_{c_{k_m}}$, the $(n_C + i_m)$th entry in $\boldsymbol{x}$ is $b_{v_{i_m}}$ and the $(n_C + j_m)$th entry in $\boldsymbol{x}$ is $b_{v_{j_m}}$. Construct the $(n_C + n_V)$-element row vector $\boldsymbol{A}_m$ such that the k_mth entry in $\boldsymbol{A}_m$ is 1, the $(n_C + i_m)$th entry in $\boldsymbol{A}_m$ is -1, the $(n_C + j_m)$th entry in $\boldsymbol{A}_m$ is 1 and all other entries in $\boldsymbol{A}_m$ are 0. Also define the variable B_m to be 0. Note that due to construction of $\boldsymbol{A}_m$ and B_m, $\boldsymbol{A}_m\boldsymbol{x} \geq B_m$ if and only if $b_{c_k} - b_{v_i} + b_{v_j} \geq 0$.

To encode the constraint that $b_{v_1} \geq 1$, note that the $(n_C + 1)$th entry in $\boldsymbol{x}$ is b_{v_1}. Therefore, construct the $(n_C + n_V)$-element row vector $\boldsymbol{A}_{v_1}$ such that the $(n_C + 1)$th entry in $\boldsymbol{A}_{v_1}$ is 1 and all other entries are 0. Also define the variable B_{v_1} to be 1. Therefore $\boldsymbol{A}_{v_1}\boldsymbol{x} \geq B_{v_1}$ if and only if $b_{v_1} \geq 1$.

Finally, to encode the constraint that $-b_{v_{n_V}} \geq 0$, note that the $(n_C + n_V)$th entry in $\boldsymbol{x}$ is $b_{v_{n_V}}$. Therefore, construct the $(n_C + n_V)$-element row vector $\boldsymbol{A}_{v_{n_V}}$ such that the $(n_C + n_V)$th entry in $\boldsymbol{A}_{v_{n_V}}$ is -1 and all other entries are 0. Also define the variable $B_{v_{n_V}}$ to be 0. Therefore $\boldsymbol{A}_{v_{n_V}}\boldsymbol{x} \geq B_{v_{n_V}}$ if and only if $-b_{v_{n_V}} \geq 0$.

Note that $\boldsymbol{A}_{v_1}\boldsymbol{x} \geq B_{v_1}$, $\boldsymbol{A}_{v_{n_V}}\boldsymbol{x} \geq B_{v_{n_V}}$ and for all $m \in \{1, \ldots, |A|\}$, $\boldsymbol{A}_m\boldsymbol{x} \geq B_m$ if and only if $b_{v_1} \geq 1$, $-b_{v_{n_V}} \geq 0$ and for all $(v_i, v_j) \in A$, if $(v_i, v_j) \in A_{c_k}$, then $b_{c_k} - b_{v_i} + b_{v_j} \geq 0$. With this in mind, define $\boldsymbol{A}$ and $\boldsymbol{B}$ as follows:

$$\boldsymbol{A}^T = \left[\boldsymbol{A}_1^T, \boldsymbol{A}_2^T, \cdots, \boldsymbol{A}_{|A|}^T, \boldsymbol{A}_{v_1}^T, \cdots, \boldsymbol{A}_{v_{n_V}}^T \right]$$

$$\boldsymbol{B} = \left[B_1, B_2, \cdots, B_{|A|}, B_{v_1}, \cdots, B_{v_{n_V}} \right].$$

Due to the construction of $\boldsymbol{A}$ and $\boldsymbol{B}$, $\boldsymbol{A}_{v_1}\boldsymbol{x} \geq B_{v_1}$, $\boldsymbol{A}_{v_{n_V}}\boldsymbol{x} \geq B_{v_{n_V}}$ and for all $m \in \{1, \ldots, |A|\}$, $\boldsymbol{A}_m\boldsymbol{x} \geq B_m$ if and only if $\boldsymbol{Ax} \geq \boldsymbol{B}$. This implies the following theorem.

Theorem 5. *Suppose an edge-colored directed graph $D = (V, A, C)$ is used to construct $\boldsymbol{A}$ and $\boldsymbol{B}$ as described above. A set of colors $I \subseteq C$ is a colored $v_1v_{n_V}$-cut in D if and only if $\boldsymbol{Ax} \geq \boldsymbol{B}$ where $b_{c_i} = 1$ if and only if $c_i \in I$ and*

$$\boldsymbol{x}^T = \left[b_{c_1}, b_{c_2}, \cdots, b_{c_{n_C}}, b_{v_1}, b_{v_2}, \cdots, b_{v_{n_V}} \right].$$

Because of Theorem 5, if the set of colors $I \subseteq C$ corresponds to a set of boolean variables $\{b_{c_1}, b_{c_2}, \ldots, b_{c_{n_C}}\}$ such that $b_{c_i} = 1$ if and only if $c_i \in I$, then the minimal cost colored cut problem is to find the the set $I \subseteq C$ that minimizes $\sum_{i=1}^{n_C} b_{c_i} cost(c_i)$ subject to $\boldsymbol{Ax} \geq \boldsymbol{B}$.

Now define the $(n_C + n_V)$-element row vector $\boldsymbol{C}$ as follows:

$$\boldsymbol{C} = \left[cost(c_1)\ cost(c_1) \cdots\ cost(c_{n_C})\ 0\ 0 \cdots\ 0\ 0 \right].$$

Note that $\boldsymbol{Cx} = \sum_{i=1}^{n_C} b_{c_i} cost(c_i)$. The above definitions imply the following corollary of Theorem 5.

Corollary 2. *For the constructions of* $\boldsymbol{A}$*,* $\boldsymbol{B}$ *and* $\boldsymbol{C}$ *from* $D = (V, A, C)$ *and some constant* k*, a boolean vector* $\boldsymbol{x}$ *subject to the constraint that* $\boldsymbol{Ax} \geq \boldsymbol{B}$ *exists such that* $\boldsymbol{Cx} = k$ *if and only if there is a colored cut* $I \subseteq C$ *is a* $v_1 v_{n_V}$*-cut in* D *subject to* $\sum_{i=1}^{n_C} b_{c_i} cost(c_i) = k$ *where in* $\boldsymbol{x}$*,* $(b_{c_i} = 1) \iff (c_i \in I)$*.*

An example of the construction of the integer programming matrices from an instance of the minimal cost colored cut problem is now given.

Example 4. Consider the edge-colored directed graph seen in Figure 3.

With the directed graph in Figure 3, let:

$$\boldsymbol{x}^T = \left[c_\alpha, c_\beta, c_\gamma, c_\lambda, r_0, r_1, r_2, r_3, r_4, r_5, \right].$$

Suppose the edges in the directed graph in Figure 3 are given the arbitrary ordering $(0,1), (3,1), (0,2), (4,3), (1,4), (2,4), (2,5), (3,5), (4,5)$. Also note that $A_\alpha = \{(0,2), (1,4)\}$, $A_\beta = \{(0,1), (3,1), (4,3)\}$, $A_\gamma = \{(2,5), (4,5)\}$ and $A_\lambda = \{(2,4), (3,5)\}$.

With the above ordering on the edges and the construction methods given above, the following assignments are made to the integer programming matrices:

$$\boldsymbol{A} = \begin{bmatrix} 0\,1\,0\,0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0\,1\,0\,0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 1\,0\,0\,0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0\,1\,0\,0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 1\,0\,0\,0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0\,0\,0\,1 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0\,0\,1\,0 & 0 & 0 & 1 & 0 & 0 & -1 \\ 0\,0\,0\,1 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0\,0\,1\,0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0\,0\,0\,0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0\,0\,0\,0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\boldsymbol{B}^T = \left[0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, \right]$$
$$\boldsymbol{C} = \left[cost(\alpha)\ cost(\beta)\ cost(\gamma)\ cost(\lambda)\ 0\ 0\ 0\ 0\ 0 \right].$$

7 Graph Cutting for Communication Selection

The method described above to convert the supervisory control sensor selection problem into an edge colored graph cutting problem is very general and can be applied to a wide range of information sharing problems in discrete-event systems. Another information sharing problem where this edge colored graph cutting problem approach can be applied is now shown in the situation of decentralized supervisory control where the task is to select the set of events which should be communicated between controllers. The material presented in this section is drawn primarily from [13].

It is assumed that for the decentralized control framework there are two controllers S_1 and S_2 controlling G denoted by $\mathcal{L}(S_1 \wedge S_2/G)$ (assuming conjunctive decentralized control as in [15]). As above, the system $S_1 \wedge S_2/G$ is said to match the specification H if $\mathcal{L}(S_1 \wedge S_2/G) = \mathcal{L}(H)$.

The local controllable events of controller S_i ($\Sigma_{ci} \subseteq \Sigma$) and the local observable events of controller S_i ($\Sigma_{oi} \subseteq \Sigma$) are those events that can be respectively disabled or observed by controller S_i. Due to the controllability and co-observability theorem from [15], there exists conjunctive controllers S_1 and S_2 such that $\mathcal{L}(S_1 \wedge S_2/G) = \mathcal{L}(H)$ if and only if $\mathcal{L}(H)$ is controllable with respect to $\mathcal{L}(G)$ and $\Sigma \setminus (\Sigma_{c1} \cup \Sigma_{c2})$ and $\mathcal{L}(H)$ is co-observable with respect to $\mathcal{L}(G)$, Σ_{o1}, Σ_{o2}, Σ_{c1} and Σ_{c2}.

It might not always be true that $\mathcal{L}(H)$ is co-observable with respect to $\mathcal{L}(G)$, Σ_{o1}, Σ_{o2}, Σ_{c1} and Σ_{c2}. This occurs when decentralized controllers do not have sufficient information to make appropriate control decisions about the system due to their observations alone. This deficiency could be overcome if the controllers are allowed to communicate. Let $\Sigma_{oij} \subseteq \Sigma_{oi}$ be the set of events that when observed by controller i are immediately communicated to controller j. This communication protocol effectively makes the events Σ_{oij} observable to controller j. With this in mind, 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}, then the pair $(\Sigma_{o12}, \Sigma_{o21})$ is called a *sufficient communication selection* because the communication of the events $(\Sigma_{o12}, \Sigma_{o21})$ gives the decentralized controllers sufficient local information about system behavior to achieve the specification.

Also without loss of generality, it is assumed that $\mathcal{L}(H)$ is always co-observable with respect to $\mathcal{L}(G)$, $\Sigma_{o1} \cup \Sigma_{o2}$, $\Sigma_{o2} \cup \Sigma_{o1}$, Σ_{c1} and Σ_{c2}. That is, a control objective can be achieved if all event observations are communicated. Unfortunately, due to reasons of economy or simplicity, it may be desired that as few events as possible are selected to be communicated. That is, the cardinality of $\Sigma_{o12} \cup \Sigma_{o21}$ should be as small as possible. This problem of finding such a minimal cardinality sufficient communication selection is known as the *communication selection problem.*

Problem 4. Communication Selection: Given G, H, Σ_{o1}, Σ_{o2}, Σ_{c1} and Σ_{c2}, find a sufficient communication selection (Σ_{o12}^{min}, Σ_{o21}^{min}) such that for any other sufficient communication selection $(\Sigma_{o12}, \Sigma_{o21})$, $|\Sigma_{o12}^{min} \cup \Sigma_{o21}^{min}| \leq |\Sigma_{o12} \cup \Sigma_{o21}|$.

Example 5. An example of Problem 4 is now given. Consider the system G in Figure 6 with $\Sigma = \{a, b, c, d\}$ and H is a copy of G such that states 4 and 6 are removed. Let $\Sigma_{o1} = \Sigma$, $\Sigma_{c1} = \emptyset$, $\Sigma_{o2} = \emptyset$ and $\Sigma_{c1} = \Sigma$. Note that controllers

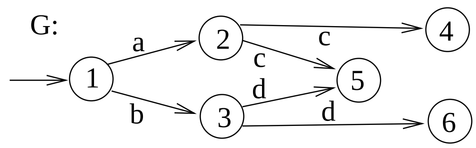

Fig. 6. System G for Example 5.

cannot be synthesized to achieve the given specification unless the controllers are allowed to communicate. This is because Controller 1 has insufficient actuation to perform the correct control action while Controller 2 has insufficient information. For this problem the minimal sufficient communication selection pair is $(\{a, b\}, \emptyset)$.

A conversion is now shown to convert Problem 4 into an instance of Problem 2 as was done with the sensor selection problem. Therefore, the approximation methods described above for the graph cutting problem can be used to approximate optimal solutions to the communication selection problem. This edge colored graph cutting problem conversion is based on a nondeterministic automaton construction is given by [14] to test if $\mathcal{L}(H)$ is co-observable with respect to $\mathcal{L}(G)$, Σ_{o1}, Σ_{o2}, Σ_{c1} and Σ_{c2}. This section presents a modified version of this construction to convert an instance of Problem 4 into an instance of Problem 2.

Suppose G, H, Σ_{o1}, Σ_{o2}, Σ_{o12}, Σ_{o21}, Σ_{c1} and Σ_{c2} are given. A nondeterministic automaton $\mathcal{M}_{\Sigma_{o12}\Sigma_{o21}}$ can be constructed to test if $(\Sigma_{o12}, \Sigma_{o21})$ is a sufficient communication selection.

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 ambiguity. The automaton $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}} = (X^{\Sigma_{o12},\Sigma_{o21}}, x_0^{\Sigma_{o12},\Sigma_{o21}}, (\Sigma \cup \Sigma_1 \cup \Sigma_2), \delta^{\Sigma_{o12},\Sigma_{o21}}, X_m^{\Sigma_{o12},\Sigma_{o21}})$ can then be defined where $X^{\Sigma_{o12},\Sigma_{o21}} = X^H \times X^H \times X^H \times G^G \cup \{d\}$, $x_0^{\Sigma_{o12},\Sigma_{o21}} = (x_0^H, x_0^H, x_0^H, x_0^G)$. The notation is used that $x \overset{\gamma}{\mapsto}_{\Sigma_{o12},\Sigma_{o21}} y$ represents that there is a transition according to the transition rules of $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ from state x to state y labeled by event γ.

The transition structure of $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ is nondeterministic so for a state, $x \in X^{\Sigma_{o12},\Sigma_{o21}}$ and an event $\gamma \in \Sigma \cup \Sigma_1 \cup \Sigma_2$, $\delta^{\Sigma_{o12},\Sigma_{o21}}(x, \gamma)$ can be a set of states as is represented below. Therefore, $y \in \delta^{\Sigma_{o12},\Sigma_{o21}}(x, \gamma)$ if an only if $x \overset{\gamma}{\mapsto}_{\Sigma_{o12},\Sigma_{o21}} y$. The state transition representations are also extended in the usual manner to be defined over strings of transitions.

In the formal definition of the transition relation, the $(*)$ condition holds at a state $x = (x_1, x_2, x_3, x_4)$ if

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

The transition relation for $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ is defined such that if $(*)$ then, $d \in \delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\sigma)$ and for all $\sigma \in \Sigma$,

$$(\delta^H(x_1,\sigma),\delta^H(x_2,\sigma),\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) \in \delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\sigma).$$

In addition, for $\sigma \in \Sigma \setminus (\Sigma_{o1} \cup \Sigma_{o2})$,

$$\delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\sigma) \subseteq \left\{\begin{array}{l} (\delta^H(x_1,\sigma),x_2,x_3,x_4) \\ (x_1,\delta^H(x_2,\sigma),x_3,x_4) \\ (x_1,x_2,\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) \end{array}\right\}.$$

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

$$\delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\Psi_1(\sigma)) \subseteq \left\{\begin{array}{l} (\delta^H(x_1,\sigma),x_2,x_3,x_4) \\ (x_1,\delta^H(x_2,\sigma),\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) \end{array}\right\}.$$

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

$$\delta^{\Sigma_{o12},\Sigma_{o21}}((x_1,x_2,x_3,x_4),\Psi_2(\sigma)) \subseteq \left\{\begin{array}{l} (x_1,\delta^H(x_2,\sigma),x_3,x_4) \\ (\delta^H(x_1,\sigma),x_2,\delta^H(x_3,\sigma),\delta^G(x_4,\sigma)) \end{array}\right\}.$$

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

The construction for $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ is modified from the construction presented by [14] in that $\Psi_1(\sigma)$ and $\Psi_2(\sigma)$ transitions correspond to state estimation updates that could be removed if σ observances would be communicated between the controllers. The $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ construction prompts the following corollary to the main result of [14].

Corollary 3. *State d is reachable from the initial state in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ if and only if $\mathcal{L}(H)$ is not 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 $\mathcal{M}_{(\Sigma_{o12}\cup\{\sigma\}),\Sigma_{o21}}$ can be constructed from $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$ by cutting all transitions labeled by $\Psi_2(\sigma)$. Therefore, the act of controller 1 communicating all occurrences of event σ to controller 2 corresponds to trimming all $\Psi_2(\sigma)$ labeled transitions in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$. Similar properties hold for the adding a σ event to Σ_{o21} and respectively trimming $\Psi_1(\sigma)$ labeled transitions in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$.

Define $\Sigma_j^{oij} = \{\Psi_j(\sigma) | \sigma \in \Sigma^{oij}\}$. A set of events $\Sigma_1^{o21} \cup \Sigma_2^{o12}$ is a $x_0^{\emptyset,\emptyset}d$-cut in $\mathcal{M}_{\emptyset,\emptyset}$ if and only if d is not reachable in $\mathcal{M}_{\Sigma_{o12},\Sigma_{o21}}$. Therefore, the pair $\left(\Sigma_{o12}^{min}, \Sigma_{o21}^{min}\right)$ is the smallest cardinality communication selection if and only if the corresponding events $\Sigma_1^{o21min} \cup \Sigma_2^{o12min} \subseteq \Sigma_1 \cup \Sigma_2$ is the smallest cardinality $x_0^{\emptyset,\emptyset}d$-cut in $\mathcal{M}_{\emptyset,\emptyset}$ when restricted to cutting transitions labeled with events in $\Sigma_1 \cup \Sigma_2$. This realization effectively converts the communication selection problem into a restricted form of Problem 2. Similar to the construction above for the sensor selection problem to convert a restricted graph cutting instance $\mathcal{M}_{\Sigma_o}$ into a true edge colored graph cutting instance $\tilde{\mathcal{M}}_{\Sigma_o}$, a similar polynomial time construction is shown in [13] to convert a restricted graph cutting problem instance $\mathcal{M}_{\emptyset,\emptyset}$ into a related graph cutting problem instance $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$ that prompts the following theorem.

Theorem 6. *Given an $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$ as constructed above, $\mathcal{L}(H)$ 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 $\tilde{x}_0^{\emptyset,\emptyset}d$-cut in the colored directed graph $\tilde{\mathcal{M}}_{\emptyset,\emptyset}$.*

8 Discussion

This paper has discussed the approximation properties of a computationally difficult sensor selection problem in supervisory control. This sensor selection problem is shown to be related to a general type of edge-colored directed graph *st*-cut problem. Solutions to both the sensor selection and graph cutting problems are difficult to approximate, but the directed graph conversion aids in the development of effective heuristic methods to approximate optimal sensor selections. The graph cutting conversion is a very general method and there are a number of important discrete-event system problems which can be converted to an edge-colored directed graph *st*-cut problem such as the decentralized control communication selection problem shown above.

Acknowledgment

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. Support for the first author was also provided by NSF grants CCR-0082784, CCR-0325571 and CCR 00-85917 ITR.

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. 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.
5. P. G. Hoel, S. C. Port, and C. J. Stone. *Introduction to Probability Theory*. Houghton Mifflin Co., 1971.
6. 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.
7. 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.
8. F. Lin and W. M. Wonham. On observability of discrete-event systems. *Information Sciences*, 44:173–198, 1988.
9. C.H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1982.
10. 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.
11. P.J. Ramadge and W.M. Wonham. The control of discrete-event systems. *Proc. IEEE*, 77(1):81–98, 1989.
12. K. Rohloff and J. H. van Schuppen. Approximating the minimal-cost sensor-selection for discrete-event systems. Technical Report MAS-R0404, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, the Netherlands, December 2004.
13. K. Rohloff and J. H. van Schuppen. Approximating minimal communicated event sets for decentralized supervisory control. In *16th IFAC World Congress*, Prague, Czech Republic, July 2005.
14. K. Rudie and J.C. Willems. The computational complexity of decentralized discrete-event control problems. *IEEE Trans. Auto. Contr.*, 40(7):1313–1318, 1995.
15. K. Rudie and W.M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Trans. Auto. Contr.*, 37(11):1692–1708, November 1992.
16. J. Tsitsiklis. On the control of discrete-event dynamical systems. *Mathematics of Control, Signals and Systems*, 2:95–107, 1989.
17. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, Berlin Heidelberg, 2001.
18. K. Wong and J. H. van Schuppen. Decentralized supervisory control of discrete event systems with communications. In *Proc. 3rd Workshop on Discrete Event Systems*, pages 284–289. Published by IEE, London, England, August 1996.
19. 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.
20. 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.