Optimal Nonblocking Directed Control of Discrete Event Systems

Jing Huang, Student Member, IEEE and Ratnesh Kumar, Fellow, IEEE
Department of Electrical and Computer Engineering
Iowa State University, Ames, Iowa 50011

Abstract—For the control of discrete event systems, we introduced the notion of directed control in [1], where a directed controller, simply referred as director, is one that selects at most one controllable event to be enabled at any instant. In [2], we developed a framework for the computation of optimal directors and a polynomial synthesis algorithm for acyclic plants. In this paper, we present a novel synthesis approach that works for general plants, i.e., plants with or without cycles, thus providing a complete solution to the optimal directed control problem introduced in [2]. Moreover, we show that the complexity of the new approach remains polynomial in the size of plant, in spite of the fact that the number of choices for a candidate director is exponential in the number of plant states.

Index Terms—discrete event system, optimal control, nonblocking, directed control, director, supervisory control, supervisor, automata, formal language

I. INTRODUCTION

For the control of discrete event systems, we introduced the notion of directed control in [1], where a directed controller, simply referred as director, is one that selects at most one controllable event to be enabled at any instant. This is in contrast to supervisory control, where a supervisory controller, simply referred as supervisor, enables a maximum allowable set of controllable events at any instant, i.e., no selection for executing an enabled event is made [3], [4], [5].

The design of a supervisor is meaningful for applications in which the plant is an autonomous generator of controllable events. However for many applications, the plant is an executor of controllable events, i.e., it does not autonomously generate such events, rather executes them when commanded by a controller. In a transportation system, for example, a supervisory control action will specify a maximal set of permissible routes for a vehicle. However, what is more appropriate is a directed control action commanding the vehicle to follow a specific route. So for systems that are executor of events, it is more meaningful to issue a command consisting of at most one possible controllable event, rather than a set of controllable events (as issued by a supervisor).

In [6], an antenna rotor control system (ARCS) has been designed where a controller enforces the given safety, liveness, and real-time control constraints, while selecting a single controllable event at each state of the system (i.e., the controller is a director). The controllable event is selected from the ones allowed by a maximally permissive supervisor, but on an ad hoc basis. Similar ad hoc selection of controllable events is made in another application consisting of an educational assembly line [7]. In [8], the authors pointed out some main issues facing the implementation of supervisors on programmable logic controllers (PLCs), one of which is the need to choose one controllable event among alternatives. The suggested solution, however, is either making the choice explicitly at the compile time, or letting a PLC choose one based on the ordering of its rungs at the run time.

The problem of computing a director can also be viewed as a generalization of the classical planning problem considered in the AI community [9], where the objective is to compute a plan (a map that selects control actions at each state) so as to steer the system from any of the given initial states to the desired final or goal states. In that setting, the notion of uncontrollability of events is missing, so our setting of directed control is more general. Thus the approach developed in the paper extends the scope of AI-planning problems to settings where uncontrollable events are also present.

In [2], we presented an optimization based approach for the computation of a director. (Optimization based approaches for computing a supervisor have been presented in [10], [11], [12], [13], [14]). There is a cost associated with each event that is a function of the trace it follows. In practice, this cost will typically depend on a bounded history of executed events. For an uncontrollable event, this cost represents the cost of executing the event and the payoff of reaching the resulting state, whereas for a controllable event, this represents the cost of executing the event and the payoff of reaching the resulting state, together with the cost of disabling other feasible controllable events. Thus a single cost function is able to capture both the “path cost” and the “control cost” (e.g., [10][11]). The cost function can represent completion time, production cost, etc. The optimization task is to direct a system in such a way that regardless of the history of evolution, it accomplishes a pending task in a minimal cost. In the absence of uncontrollable events this amounts to finding the shortest cost path between any state and its nearest reachable marked states. In the presence of uncontrollable events, control is exercised in a manner that the worst cost over all surviving paths between any state and its nearest reachable marked states is minimized. We proved in [2] that there exists an optimal director if and only if there exists a subset of marked states such that its region of directive attraction is invariant and includes the initial state. Further, for a trim acyclic plant, we showed that an optimal director always exists and developed an algorithm with polynomial complexity for computing such a director.

In this paper, we present a novel approach for computing an optimal director for general plants, i.e., plants with or without cycles. The total number of state-based directors is exponential...
in the number of plant states, and since the complexity of computing the cost of any given director is polynomial \[2\], the complexity of identifying an optimal director by an exhaustive search (i.e., by enumerating all directors and comparing their costs) will be exponential in the number of plant states. The main contribution of this paper is that it provides a complete solution to the optimal directed control problem introduced in \[2\] while its complexity remains polynomial in the size of plant.

The remainder of this paper is organized as follows. Section II gives the basic notation and preliminaries. Section III reviews the notion of directed control and the problem of optimal directed control. Section IV provides a two-step approach to synthesize an optimal director for general plants, along with some examples to aid the understanding of the approach. Section V presents an application example to demonstrate our result and section VI concludes the paper. An appendix is provided recalling some relevant algorithms from \[2\].

II. NOTATION AND PRELIMINARIES

A DES to be controlled, called plant, is modeled as an automaton, \(G := (X, \Sigma, \alpha, x_0, X_m)\), where \(X\) denotes the set of states, \(\Sigma\) denotes the finite set of events, \(\alpha : X \times \Sigma \to X\) denotes the partial deterministic state transition function and is extended in a natural way to \(\alpha : X \times \Sigma^* \to X\), \(x_0 \in X\) denotes the initial state, and \(X_m \subseteq X\) denotes the set of marked states. For \(x \in X\), we use \(\Sigma(x) \subseteq \Sigma\) to denote the set of events defined at \(x\), i.e., \(\Sigma(x) := \{\sigma \in \Sigma \mid \alpha(x, \sigma)\text{ is defined}\}\). \(\Sigma^*\) is used to denote the set of all finite-length sequences of events, called traces, which includes the zero-length trace \(\epsilon\). A subset of \(\Sigma^*\) is called a language. The generated language of \(G\) is defined as, \(L(G) := \{s \in \Sigma^* \mid \alpha(x_0, s)\text{ is defined}\}\), whereas the marked language of \(G\) is defined as, \(L_m(G) := \{s \in L(G) \mid \alpha(x_0, s) \in X_m\}\). \(X_t := \{x \in X \mid \Sigma(x) = \emptyset\}\) is used to denote the set of all terminating states and \(X_{tm} := X_t \cap X_m\) is used to denote the set of all terminating marked states. We use \(L_t(G) := \{s \in L(G) \mid \alpha(x_0, s) \in X_t\}\) to denote the set of terminating traces of \(G\). \(X(G)\) is used to denote the state set of \(G\). A state \(x \in X\) is called accessible if there exists a trace \(s \in \Sigma^*\) such that \(x = \alpha(x_0, s)\). A state \(x \in X\) is called coaccessible to \(X_m\), or simply coaccessible, if there exists a trace \(s \in \Sigma^*\) such that \(\alpha(x, s) \in X_m\). We denote the operation to delete the states in \(G\) that are not accessible (resp. coaccessible) as \(Ac(G)\) (resp. \(CoAc(G)\)). An automaton \(G\) is called accessible (resp. coaccessible) if \(G = Ac(G)\) (resp. \(G = CoAc(G)\)). An automaton \(G\) that is both accessible and coaccessible is said to be trim. For a language \(K \subseteq \Sigma^*\), the notation \(pr(K)\), called the prefix-closure of \(K\), denotes the set of all prefixes of traces in \(K\). \(K\) is said to be prefix-closed if \(K = pr(K)\). We use \(K\backslash s\) to denote the set of traces that occur in the language \(K\) after the trace \(s\) has occurred, i.e., \(K\backslash s := \{t \in \Sigma^* \mid st \in K\}\). The notation \(s \leq t\) is used to denote that the trace \(s \in \Sigma^*\) is a prefix of the trace \(t \in \Sigma^*\). When \(s\) is a proper prefix of \(t\) (i.e., \(s \leq t\) and \(s \neq t\)), it is denoted as \(s < t\).

For control purposes, the event set of \(G\) is partitioned into the set of controllable events \(\Sigma_c \subseteq \Sigma\) and the set of uncontrollable events \(\Sigma_u \subseteq \Sigma\). We define, \(\Sigma_c(x) := \Sigma(x) \cap \Sigma_c\) and \(\Sigma_u(x) := \Sigma(x) \cap \Sigma_u\). An uncontrollable event can be either a disturbance input or a sensor output. Occurrence of a disturbance input is unexpected while that of a sensor output is something that can be expected. The set of uncontrollable events that are disturbance inputs is denoted as \(\Sigma_d \subseteq \Sigma_u\). (The remaining uncontrollable events in \(\Sigma_u - \Sigma_d\) are the sensor outputs.)

III. NOTION OF DIRECTED CONTROL AND OPTIMAL DIRECTED CONTROL PROBLEM

A directed controller, or simply a director, enables at most one controllable event following each trace. This control selection is what distinguishes a director from a supervisor.

It should be noted that following certain traces, disabling all controllable events is not a good option. These consist of

- Traces \(s \in L_m(G) - L_t(G)\) such that \(L(G)\backslash s \cap \Sigma_c = \emptyset\). These traces are non-terminating marked traces and not followed by any uncontrollable events (so disabling all controllable events at such traces will block system from performing future tasks), and
- Traces \(s \in L(G)\) such that \(L(G)\backslash s \cap \Sigma_c \neq \emptyset\) and \(\emptyset \subset L(G)\backslash s \cap \Sigma_u \subseteq \Sigma_d\). These traces are followed by at least one controllable event (so a control can be exercised at such traces) while all uncontrollable events occur after the traces are disturbance inputs and at least one such disturbance input is present (so disabling all uncontrollable events will make the system wait for a disturbance input to occur in order to evolve further).

We use \(L_c(G)\) to denote the set of traces mentioned above, i.e.,

\[
L_c(G) := \{s \in L_m(G) - L_t(G) \mid L(G)\backslash s \cap \Sigma_u = \emptyset\} \cup \{s \in L(G) \mid (L(G)\backslash s \cap \Sigma_c \neq \emptyset) \land (\emptyset \subset L(G)\backslash s \cap \Sigma_u \subseteq \Sigma_d)\}
\]

**Definition 1:** A director is a map \(D : L(G) \to 2^{\Sigma^*}\) such that \(\forall s \in L(G) : |D(s)| \leq 1; s \in L_c(G) \Rightarrow |D(s)| = 1\).

Following the execution of a trace \(s \in L(G)\), the director enables at most one controllable event unless the trace belongs to \(L_c(G)\), in which case the director enables exactly one controllable event. Also note that no control decision is defined with respect to uncontrollable events; such events remain enabled. Thus the set of events enabled by a director following a trace \(s \in L_c(G)\) is given by \(D(s) \cup \Sigma_u\).

The directed plant is denoted by \(G^D\), and the languages generated and marked by the directed plant are denoted by \(L(G^D)\) and \(L_m(G^D)\) respectively, which are defined as follows:

\[
\epsilon \in L(G^D) \\
\{s \in L(G^D), \sigma \in D(s) \cup \Sigma_u, s\sigma \in L(G) \} \Leftrightarrow [s\sigma \in L(G^D)] \\
L_m(G^D) := L(G^D) \cap L_m(G)
\]

It is clear that \(pr(L_m(G^D)) \subseteq L(G^D)\). A director \(D\) is said to be nonblocking if \(pr(L_m(G^D)) = L(G^D)\).

Next we review the problem of optimal directed control, introduced in \[2\]. Let \(c : \Sigma^* \times \Sigma \to R^+\) denote a cost function of control, where \(R^+\) denotes the set of positive reals,
including infinity. For a trace \( s \in \Sigma^* \) and an uncontrollable event \( \sigma \in \Sigma_u \), \( c(s, \sigma) \) represents the execution cost of \( \sigma \) following the execution of \( s \) and the payoff of reaching the resulting state \( \alpha(x_0, s\sigma) \), whereas for a controllable event \( \sigma \in \Sigma_c \), \( c(s, \sigma) \) represents the execution cost of \( \sigma \) following the execution of \( s \) and the payoff of reaching the resulting state \( \alpha(x_0, s\sigma) \), together with the disableness cost of all other controllable events feasible at trace \( s \). Thus a single cost function suffices to capture both the “control cost” and “path cost” introduced in [11]. Also, the cost \( c(s, \sigma) \) can include the cost of reaching the plant state \( \alpha(x_0, s\sigma) \). For example, if the state \( \alpha(x_0, s\sigma) \) is an illegal state (or equivalently, the trace \( s\sigma \) does not belong to the specification language), then we can set \( c(s, \sigma) = \infty \).

By induction, the cost function can be extended to a mapping \( c : \Sigma^* \times \Sigma^* \to \mathbb{R}^+ \) as follows:

\[
\forall s, t \in \Sigma^*, \sigma \in \Sigma : \begin{cases} 
(c(s, \epsilon) := 0 \text{ and } & (c(s, t\sigma) := c(s, t) + c(st, \sigma) \\
\end{cases}
\]

We next introduce the notion of frontier traces needed for defining the cost of a director.

**Definition 2:** Given a trace \( s \in \Sigma^* \), its set of frontier traces in a language \( K \), denoted by \( (K\backslash s)_f \), is given by:

\[
\forall s \in \Sigma^* : (K\backslash s)_f := \{ t \neq \epsilon | st \in K \text{ and } \forall u < t : su \notin K \}
\]

The frontier traces of \( s \) are the minimal extensions of \( s \) such that the extensions belong to \( K \). If \( K \) is the set of marked traces, then execution of any such extension implies accomplishing a task that is pending to be completed.

**Definition 3:** Given a plant \( G \), a director \( D : L(G) \to 2\Sigma_u \), a trace \( s \in L(G^D) \) and a marked frontier (trace) extension \( t \in (L_m(G^D)\backslash s)_f \), \( c(s, t) \) denotes the cost of “completing the task” along the trace \( t \) following the trace \( s \). We consider the costs of all such marked frontier extensions \( t \) of \( s \) in \( L(G^D) \), and determine the worst possible cost \( \max_{t \in (L_m(G^D)\backslash s)_f} c(s, t) \). We do this for all traces \( s \) of \( L(G^D) \), and use the largest worst case cost among all traces \( s \) as the cost \( P(D) \) of the director \( D \), i.e.,

\[
P(D) := \begin{cases} 
\max_{s \in L(G^D)} \max_{t \in (L_m(G^D)\backslash s)_f} (c(s, t)) & \text{if } \forall s \in L(G^D) : ||L_m(G^D)\backslash s)_f < \infty \\
\infty & \text{otherwise}
\end{cases}
\]

where for any \( s \in L(G^D) \) such that \( (L_m(G^D)\backslash s)_f = \emptyset \), we define

\[
\max_{t \in (L_m(G^D)\backslash s)_f} c(s, t) := \begin{cases} 
0 & \text{if } s \in L_t(G) \cap L_m(G^D) \\
\infty & \text{otherwise}
\end{cases}
\]

Since executing a marked trace amounts to finishing a pending task, \( P(D) \) thus represents the worst case cost of finishing a pending task under the control of \( D \). It follows from the definition that if \( D \) is blocking, i.e., if there exists \( s \in L(G^D) - L_m(G^D) \) such that \( (L_m(G^D)\backslash s)_f = \emptyset \), then \( P(D) = \infty \). Thus any director with finite cost must be nonblocking.

The objective of the optimal control is to find a director with the least cost.

**Optimal directed control problem:** Determine whether there exists \( D^* \in \arg \{ \min_D P(D) \} \) with \( P(D^*) < \infty \) and if yes, then find one such \( D^* \).

**Remark 1:** In our framework, once a non-terminating marked state is reached, the system gets “re-initialized” with the marked state just reached acting as a new initial state and the existing marked states continuing to act as the final states. Note that for a plant \( G := (X, \Sigma, \alpha, x_0, X_m) \), \( I(G) := \{ x_0 \} \cup (X_m - X_t) \) is used to denote the set of all possible states from which the plant can “re-initialize”. It is possible that a re-initialization results in a completely new set of marked states. Our framework can easily accommodate this case by defining an extended model as the concatenation of a suitable number of appropriately initialized plant models.

By defining a certain equivalence relation over the set of traces, it is possible to suitably refine the plant model so that the cost function can be viewed as “state-based” (with respect to the states of the refined plant). While in general such a refinement may not preserve the finiteness of the state space, it will do so in any practical setting where typically the given trace-based cost function will depend only on a bounded history of the system evolution. With this understanding, we formulate and study the optimal director problem in a state-based setting. A cost function is state-based if for all \( s, t \in \Sigma^* \) such that \( \alpha(x_0, s) = \alpha(x_0, t) := x \), it holds that for all \( u \in \Sigma^* \) such that \( \alpha(x, u) \) is defined, \( c(s, u) = c(t, u) \). In such a case, the cost function can be specified as a map, \( c : X \times \Sigma \to \mathbb{R}^+ \) so that for \( x \in X \) and \( \sigma \in \Sigma \), \( c(x, \sigma) \) is the cost of executing \( \sigma \) at state \( x \). A director \( D \) is state-based if it computes control action as a function of plant state, i.e., \( D : X \to 2^\Sigma^* \).

**Definition 4:** A state-based director is a map \( D : X \to 2\Sigma_c \) such that \( \forall x \in X : |D(x)| \leq 1 \); \( x \in X_e \Rightarrow |D(x)| = 1 \), where

\[
X_e := \{ x \in X_m - X_t | \Sigma_u(x) = \emptyset \} \cup \{ x \in X | (\Sigma_c(x) \neq \emptyset) \land (\emptyset \subset \Sigma_u(x) \subset \Sigma_d) \}
\]

In the remainder of the paper, we represent a plant \( G := (X, \Sigma, \alpha, x_0, X_m) \) as a weighted directed-graph \( G := (X, E, X_m) \), where \( E \subseteq X \times \Sigma \times X \) is the set of transitions, called the set of edges. An edge of \( G \) is an ordered triple \( e = (x_1, \sigma, x_2) \in E \) with \( \alpha(x_1, \sigma) = x_2 \), and is said to be labeled by event \( \sigma \), and directed from state \( x_1 \) to state \( x_2 \). The edge \( e \) is assigned a control cost, denoted \( c(e) \), with value \( c(e) = c(x_1, \sigma) \).

We use \( E(x) \) to denote the set of edges defined at \( x \), and \( E_c(x) \) (resp. \( E_u(x) \)) to denote the set of edges labeled with controllable (resp. uncontrollable) events at \( x \). The trim directed plant under a state-based director \( D \) is given by \( G^D := (X^D, E^D, X_m^D) \), where \( E^D \subseteq E \) represents the set of transitions enabled by the director \( D \) and \( X^D \) (resp. \( X_m^D \)) represents the set of states (resp. the set of marked states) in the trim directed plant. Recall that the set of all possible initialization states of \( G^D \) is given by \( I(G^D) := \{ x_0 \} \cup (X_m - X_t) \).
For $x \in X$, $D \setminus x$ is used to denote the director $D$ with $x$ treated as the initial state, and we call it a director rooted at $x$. Note that $D\setminus x_0 = D$.

A path $p$ is a finite sequence of edges $p = (x_1, \sigma_1, x_2)(x_2, \sigma_2, x_3) \cdots (x_{n-1}, \sigma_{n-1}, x_n)$, in which $p$ is said to start from $x_1$ and end at $x_n$. The cost of path $p$ is the sum of the costs of its edges, and is denoted by $c(p)$. For each $x \in X$, let $\Pi^f_G(x)$, called the set of marked frontier paths from $x$ in $G$, denote the set of all paths of the form $p = (x_1, \sigma_1, x_2)(x_2, \sigma_2, x_3) \cdots (x_{n-1}, \sigma_{n-1}, x_n)$ such that $x_1 = x, x_n \in X_m$, and for all $i = 2, \cdots, n-1, x_i \notin X_m$. The states reached from $x$ via the marked frontier paths of $\Pi^f_G(x)$ are called $x$'s marked frontier states.

Definition 5: The distance of $x$ (to its marked frontier states in $G$), denoted $d_G(x)$, is defined as the worst cost of all paths in $\Pi^f_G(x)$, i.e.,

$$d_G(x) := \begin{cases} \max\{c(p) \mid p \in \Pi^f_G(x)\} & \text{if } 0 < |\Pi^f_G(x)| < \infty \\ 0 & \text{if } \Pi^f_G(x) = \emptyset \text{ and } x \in X_{tm} \\ \infty & \text{otherwise} \end{cases}$$

where

$$d_1 := \max\{d_G(x') + c(e) \mid e \in (x, \sigma, x') \in E\}$$
$$d_2 := \max\{0 + c(e) \mid e \in (x, \sigma, x') \in E\}$$

From Definitions 3 and 5, it follows that the cost of director $D$ is given by,

$$P(D) = \max_{x \in X_m} d_{G^D}(x)$$

(2)

Note that if two states $x_1$ and $x_2$ are such that every path in $\Pi^f_G(x_1)$ is a suffix of some path in $\Pi^f_G(x_2)$, then $d_G(x_1) \leq d_G(x_2)$. So Equation 2 is equivalent to the following:

$$P(D) = \max_{x \in \Omega(G^D)} d_{G^D}(x)$$

(3)

It follows that the optimal control problem is to find

$$D^* \in \arg\left\{\min_{D} P(D)\right\} = \arg\left\{\min_{D} \max_{x \in \Omega(G^D)} d_{G^D}(x)\right\}$$

Remark 2: In our setting, we do not explicitly define “illegal” states, but they can be identified by suitably defining the cost function. Any state that is deemed illegal should have an infinite cost for all incoming edges to the state. As a result, none of such edges should be present in a trim optimally directed plant, ensuring the unreachability of that state.

Remark 3: If a plant $G$ is not trim or contains edges with infinite cost, then such a plant can be reduced to a trim one having no edges with infinite cost while preserving the solution to the optimal director problem. We provide an algorithm (Algorithm 3 in the Appendix), adapted from [2], for such a reduction. It follows that there is no loss in generality to consider only the plants that are trim and all edges are of finite cost.

Since at any state $x \in X$, a director can enable one of the feasible controllable events in $\Sigma(x)$ or none of such events, there are $|\Sigma(x)| + 1 \leq |\Sigma| + 1$ choices for control per state. It is clear that the total number of state-based directors is upper bounded by $|\Sigma| + 1)^{|X|}$, whereas the complexity of computing the cost of any given director $D$ is known to be $O(|X^P| \times |\Sigma|)$ [2]. So if an optimal director is identified by way of enumerating all directors and comparing their costs, the complexity of such an exhaustive search will be exponential in the number of plant states.

IV. SYNTHESIS OF OPTIMAL DIRECTOR FOR GENERAL PLANTS

Central to our director synthesis is the notion of attraction introduced in [15], which is presented below.

Definition 6: Given a plant $G$, consider a state set $\hat{X} \subseteq X$. A state $x \in X$ is said to be $1$-step $\hat{X}$-attractable in $G$ if $x, x \in X_{\hat{X}}$ and $x \in X$ is said to be $1$-step directly $\hat{X}$-attractable in $G$ if there exists a director $D$ such that $x$ is $1$-step $\hat{X}$-attractable in $G^D$. In other words, $x \in X$ is $1$-step directly $\hat{X}$-attractable in $G$ if $\alpha(x, \Sigma) \cap \hat{X} \neq \emptyset$. $x \in X$ is $1$-step directly $\hat{X}$-attractable in $G$ if $\alpha(x, \Sigma) \cap \hat{X} \neq \emptyset$. In the following definition, the notion of single-step attractability is generalized to multi-step attractability.

Definition 7: Given a reference state set $\hat{X} \subseteq X$, $x \in X$ is $\hat{X}$-attractable in $G$ if there exists a non-negative integer $N$ such that for all paths $p$ from $x$ of length greater than or equal to $N$, $p$ visits $x \in X$ is $\hat{X}$-directly $\hat{X}$-attractable in $G$ if there exists a director $D$ such that $x$ is $\hat{X}$-attractable in $G^D$. We use $\Omega(\hat{X})$, called the region of attraction of $\hat{X}$, to denote the set of all $\hat{X}$-attractable states, and $\Omega_D(\hat{X})$, called the region of directive attraction of $\hat{X}$, to denote the set of all directly $\hat{X}$-attractable states. A state set $\hat{X} \subseteq X$ is said to be $\hat{X}$-attractable in $\hat{X}$ if $\hat{X} \subseteq \Omega(\hat{X})$, and directly $\hat{X}$-attractable in $\hat{X}$ if $\hat{X} \subseteq \Omega_D(\hat{X})$.

An algorithm to compute an optimal director for acyclic plants with complexity that is polynomial in the number of plant states was presented in [2] (included in the Appendix as Algorithm 4). It is based on the observation that the region of attraction of the set of terminating marked states is the whole state set in a trim acyclic plant, i.e., $X \subseteq \Omega(X_{tm})$. Algorithm 4 constructs the region of attraction of $X_{tm}$ iteratively: it starts with $\Omega_1 := X_{tm}$; in the $k$th iteration, all the states that are 1-step attractable to $\Omega_k$ are combined with $\Omega_k$ to form the state set $\Omega_{k+1}$. For each added state $x$, Algorithm 4 optimizes the control action by comparing all the feasible transitions from $x$ to the states in $\Omega_k$. Note that the value of $\rho(x)$ and $\Pi(D^\land(x))$ computed by Algorithm 4 represents $x$'s distance (to its marked frontier states) in $G^D$, and the cost of the director rooted at $x$, respectively.

In general, a plant model can contain cycles and in which case Algorithm 4 is not applicable. This situation is illustrated by the following examples.

Example 1: Consider a plant $G$ shown in Figure 1(a). Algorithm 4 starts with $\Omega_1 = \{x_3\}$. Due to the cycle between
$x_1$ and $x_2$, neither $x_1$ nor $x_2$ is 1-step attractable to $\Omega_1$ and thus Algorithm 4 can not proceed or terminate because $X \subseteq \Omega(X_{tm})$ no longer holds in this case. An optimal director for $G$, however, does exist, which yields the optimally directed plant as shown in Figure 1(b) with director cost of 4.

**Example 2:** Consider another plant $G$ shown in Figure 1(c). In this case there is no terminating marked state and so Algorithm 4 does not even start. Suppose we let Algorithm 4 start from all marked states (including the non-terminating marked states), then since both $x_1$ and $x_2$ are marked states, we will have $\Omega_1 = \{x_1, x_2\}$. Then Algorithm 4 will compute the values of $\rho(x)$ and $\Pi(D^{\prime \setminus x})$ for each state $x$. Each state in Figure 1(c) is labeled with such a pair of numbers. When Algorithm 4 terminates, it will disable the transition from $x_0$ to $x_1$ while enable that from $x_0$ to $x_2$. The resulting directed plant is shown in Figure 1(d) with director cost of 9. However, the optimally directed plant should be the one shown in Figure 1(e), with director cost of 7.

Note that Algorithm 4 is “greedy” or “locally optimal” in the sense that when computing $\Omega_2 = \Omega_1 \cup \{x_0\}$, it picks the edge from $x_0$ to $x_2$ with cost of 3 instead of the edge from $x_0$ to $x_1$ with cost of 5. This ignores the fact that the distance of $x_2$ is 6, larger than the distance of $x_1$, which is 2. Since the cost of a director is determined by the worst distance among those of the initialization states (see Equation 3), the above “locally optimal” selection is not “globally optimal”. If, however, the set of marked states to be present in a trim optimally directed plant could be identified, then Algorithm 4 could be forced to consider only those marked states. For example, by omitting state $x_2$ in $\Omega_1$, Algorithm 4 would be forced to pick the edge from $x_0$ to $x_1$ and yields an optimal director. Thus by trying all subsets of $X_m$ as a choice for $\Omega_1$ and applying a locally optimal algorithm, a globally optimal director could be obtained. However, such a strategy could have complexity that is exponential in the number of marked states. We will show that not all subsets of $X_m$ need to be considered as a choice for $\Omega_1$, which is the key to a polynomial complexity algorithm.

We present a two-step algorithm in this paper. We first provide a synthesis algorithm (Algorithm 1) that is applicable for general plant models, cyclic or acyclic. The director computed by Algorithm 1 is “locally optimal” in the sense that the distance of each state under directed control is minimized. We then present another algorithm (Algorithm 2) which iteratively applies Algorithm 1 on different input plants starting from $G$ and compares the resulting directors from various iterations. During each iteration, Algorithm 2 removes the states with maximum distance to produce the input plant for the next iteration. It will be shown that among all the directors computed during iterations of Algorithm 2, the one with the minimum cost is indeed optimal.

It will also be shown that Algorithm 1 is of polynomial complexity in the number of plant states, and the number of times in which Algorithm 1 is executed inside Algorithm 2 is at most one more than the number of non-terminating marked states. So the overall complexity of synthesis of an optimal director remains polynomial in the plant size.

We first present Algorithm 1.

**Algorithm 1:** Input a plant $G := (X, E, X_m)$, the following steps compute a “locally optimal” director $D^*$ and the distance of each state $x \in X$ under the control of $D^*$.

**Initiation:**
Set $k = 1$, $\Omega_k = X_m$, $\sum_k = X_{tm}$, $\forall x \in \Omega_k : \lambda(x) = 0$ and $\forall x \in \Omega_k : D^o(x) = \emptyset$ and $\rho(x) = 0$.

**Iteration:**
1) Let $U_k$ be the set of all 1-step directly $\Omega_k$-attractable states excluding those in $\Omega_k^l$, i.e.,

$$U_k := \{x \in X - \Omega_k^l \mid \alpha(x), \Sigma) \cap \Omega_k \neq \emptyset, \alpha(x, \Sigma_u) \subseteq \Omega_k \land x \notin X_e \Rightarrow \alpha(x, \Sigma_e) \cap \Omega_0 \neq \emptyset\}$$

2) For each $x \in U_k$, we define

$$E_{ck}(x) := \{e = (x, \sigma, x') \in E_e(x) \mid x' \in \Omega_k\}$$

and compute the following:

$$D_k(x) = \begin{cases} \{x \in X_e \mid \Sigma_u(x) = \emptyset\} & \text{if} \quad x \in X_e \\ \emptyset & \text{otherwise} \end{cases}$$

such that $e_k = (x, \sigma_k, x') \in E_{ck}(x)$ is any edge that belongs to the argument of

$$\rho^u_k(x) = \begin{cases} \max_{e \in E_{ck}(x)} \{\lambda(x') + c(e) \mid e = (x, \sigma, x')\} & \text{if} \quad x \in X_e \lor \Sigma_u(x) = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$$\rho^u_k(x) = \begin{cases} \max_{e \in E_{ck}(x)} \{\lambda(x') + c(e) \mid e = (x, \sigma, x')\} & \text{if} \quad E_u(x) \neq \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$$\rho^u_k(x) = \max(\rho^u_k(x), \rho_k^u(x))$$

$$E^{D_k}(x) = \{\{x, \sigma, x' \in E \mid \sigma \in D_k(x) \cup \Sigma_u(x)\}\}$$

![Fig. 1. Application of Algorithm 4 on plants with cycles](image-url)
3) Let $V_k \subseteq U_k$ be the set of states that belong to the argument of $\min_{x \in U_k} \rho^*_k(x)$. For each state $x \in V_k$, set $\rho(x) = \rho^*_k(x)$, $D^p(x) = D_k(x)$ and $E^{D_p}(x) = E^{D_k}(x)$. If $x \notin X_m$, then let $\lambda(x) = \rho(x)$.

4) Set $\Omega_{k+1} = \Omega_k \cup V_k$ and $\Omega'_{k+1} = \Omega'_k \cup V_k$.

**Termination:**

- If $U_k = \emptyset$, then stop. Set $\rho(x) = \infty$ for each state $x \in X - \Omega_k$ and define $P(D^p) := \max_{x \in \text{int}(G^{D^p})} \rho(x)$.
- If $U_k \neq \emptyset$, then continue iteration with $k = k + 1$.

**Remark 4:** At each iteration, at least one state is added into $\Omega_k$ and so there is a maximum of $|X|$ iterations. Also for every iteration, we perform certain computations for those states that are 1-step directly-attainable to $\Omega_k$-attractable. The complexity of this step is linear in the number of such states and the number of their edges. So, the overall complexity of Algorithm 1 is $O(|X| \times |\Sigma|)$.

We present an example to aid the understanding of Algorithm 1.

**Example 3:** The plant is shown in Figure 2, where an edge with double arrows represents a transition on an uncontrollable event and an edge with single arrow represents a transition on a controllable event. Note that all the uncontrollable events in the plant are disturbance inputs, so we have $X_c = \{x_8\}$ and the only controllable transition at $x_6$ should not be disabled by any director. As Algorithm 1 proceeds, it iteratively computes the values of $\Omega_k$, which is shown in Figure 2 as the set of states encircled. For each state $x$ in the plant, the values of $\lambda(x)$, $\rho(x)$ are also shown as a pair beside the state $x$.

**Initiation:**

Since $X_{t_0} = \emptyset$ and $X_m = \{x_3, x_4\}$, $\Omega_1 = \{x_3, x_4\}$, $\Omega'_1 = \emptyset$ and $\lambda(x_3) = \lambda(x_4) = 0$.

**Iteration no. 1:**

The states $x_0$, $x_1$ and $x_2$ are all 1-step directly-attractable to $\Omega_1$. Therefore $U_1 = \{x_0, x_1, x_2\}$. Since $\rho^*_1(x_0)$ is the minimum among $\rho^*_1(x_0), \rho^*_1(x_1)$ and $\rho^*_1(x_2)$, we get $V_1 = \{x_0\}$ and $\rho(x_1) = \lambda(x_1) = 2$. We have $\Omega_2 = \Omega_1 \cup V_1 = \{x_1, x_3, x_4\}$ and $\Omega'_2 = \Omega'_1 \cup V_1 = \{x_1\}$.

**Iteration no. 2:**

The states $x_0$ and $x_2$ are both 1-step directly-attractable to $\Omega_2$. Therefore $U_2 = \{x_0, x_2\}$. Since $\rho^*_2(x_0)$ is the minimum between $\rho^*_2(x_0)$ and $\rho^*_2(x_2)$, we get $V_2 = \{x_0\}$ and $\rho(x_2) = \lambda(x_2) = 3$. The control selection at $x_0$ is to select the controllable transition from $x_0$ to $x_4$. We have $\Omega_3 = \Omega_2 \cup V_2 = \{x_0, x_1, x_3, x_4\}$ and $\Omega'_3 = \Omega'_2 \cup V_2 = \{x_0, x_1\}$.

**Iteration no. 3:**

The states $x_2$ and $x_3$ are both 1-step directly-attractable to $\Omega_3$. Therefore $U_3 = \{x_2, x_3\}$. Since $\rho^*_3(x_3)$ is the minimum between $\rho^*_3(x_2)$ and $\rho^*_3(x_3)$, we get $V_3 = \{x_3\}$ and $\rho(x_3) = 5$. We have $\Omega_4 = \Omega_3 \cup V_3 = \{x_0, x_1, x_3, x_4\}$ and $\Omega'_4 = \Omega'_3 \cup V_3 = \{x_0, x_1, x_3\}$.

**Iteration no. 4:**

The state $x_2$ is the only state that is 1-step directly-attractable to $\Omega_4$. Therefore $U_4 = \{x_2\}$, $V_4 = \{x_2\}$ and $\rho(x_2) = \lambda(x_2) = 6$. We have $\Omega_5 = \Omega_4 \cup V_4 = \{x_0, x_1, x_2, x_3, x_4\}$ and $\Omega'_5 = \Omega'_4 \cup V_4 = \{x_0, x_1, x_2, x_3\}$.

**Iteration no. 5:**

The state $x_4$ is the only state that is 1-step directly-attractable to $\Omega_5$. Therefore $U_5 = \{x_4\}$, $V_5 = \{x_4\}$ and $\rho(x_4) = 11$. We have $\Omega_6 = \Omega_5 \cup V_5 = \{x_0, x_1, x_2, x_3, x_4\}$ and $\Omega'_6 = \Omega'_5 \cup V_5 = \{x_0, x_1, x_2, x_3, x_4\}$.

**Termination:**

Since $U_k = \emptyset$, the algorithm terminates. Note that $\rho(x_5) = \rho(x_6) = \rho(x_7) = \infty$. The directed plant is shown in Figure 2(h) with control cost of 11. Note that the director computed by Algorithm 1 may not be optimal. The optimally directed plant for this example should be the one shown in Figure 2(i) with control cost of 9.

The function $\rho : X \rightarrow \mathbb{R}$ computed by Algorithm 1, where $\mathbb{R}$ denotes non-negative reals including infinity, is called the distance function of $D^p$. The value of $\rho(x)$ for each state $x \in X^{D^p}$ represents the distance of $x$ under the control of $D^p$, i.e., we have following theorem.

**Theorem 1:** Given a plant $G := (X, E, X_m)$ and consider the notation of Algorithm 1. It holds that $\forall x \in X^{D^p} : \rho(x) = d_{G^{D^p}}(x)$.

**Proof:** We prove by induction on the iteration ordinal $k$ of Algorithm 1. Since $\Omega'_1 = X_{t_m}$ and $\forall x \in \Omega'_1 : D^p(x) = \emptyset$ and $\rho(x) = 0$, the base step trivially holds.

Assume for induction that the theorem holds for the states of $\Omega_k$, we need to show the theorem holds for the states of $\Omega_{k+1} = \Omega_k \cup V_k$.

Consider a state $x \in V_k$. Then from the definition of 1-step directly-attractability, for every edge $e = (x, \sigma, x') \in E^{D^p}$, we have $x' \in \Omega_k$. Note that $\Omega_k \supseteq \Omega_k$ and $\Omega_k - \Omega'_k \subseteq X_m$. So either $x' \in \Omega'_k - X_m$ or $x' \in X_m$.

If $x' \in \Omega'_k - X_m$, then it follows from the induction hypothesis that $\rho(x') = d_{G^{D^p}}(x')$. Furthermore, from the construction of the algorithm, we have $\lambda(x') = \rho(x') = d_{G^{D^p}}(x')$. If $x' \in X_m$, then we have $\lambda(x') = 0$. Using these values of $\lambda(x')$, and applying definition of $\rho(x) = \rho^*_k(x)$, we obtain:

$$\rho(x) = \max\{\lambda(x') + c(e) \mid e = (x, \sigma, x') \in E^{D^p}\} = \max\{d_1, d_2\}$$

where

$$d_1 := \max\{d_{G^{D^p}}(x') + c(e) \mid e = (x, \sigma, x') \in E^{D^p}, x' \notin X_m\}$$

$$d_2 := \max\{0 + c(e) \mid e = (x, \sigma, x') \in E^{D^p}, x' \in X_m\}$$

Then the result follows from Equation (1).

We claim that for each state $x \in X^{D^p}$, the value of $\rho(x)$ computed by Algorithm 1 is the minimum distance of $x$ under directed control. We need the following lemma to establish this claim.

**Lemma 1:** Consider the notation of Algorithm 1, assume Algorithm 1 terminates after the $n$th iteration for some $n \geq 0$ and let $x_n \in V_k$ be any state of $V_k$, then $0 < \rho(x_1) < \rho(x_2) < \ldots < \rho(x_n) < \ldots < \rho(x_{k+n})$.

**Proof:** Note that for any two states $\hat{x} \in V_k$ and $\bar{x} \in V_k$, we have $\rho(\bar{x}) = \rho(\hat{x}) = \min_{e \in U_k} \rho^*_k(e)$.

We prove the lemma by induction on the iteration ordinal $k$ of Algorithm 1. First we show that the base step holds, i.e., $0 < \rho(x_1)$.

Since only terminating marked states have the distance of 0 and $x_1 \notin \Omega'_k = X_{t_m}$, it follows from Theorem 1 that $\rho(x_1) = d_{G^{D^p}}(x_1) > 0$. Assume for induction that the lemma holds after the $k$th iteration, i.e., $0 < \rho(x_1) < \rho(x_2) < \ldots < \rho(x_{k})$. We need to show $\rho(x_k) < \rho(x_{k+1})$. 


Clearly, \( x_{k+1} \in U_{k+1} \). So either \( x_{k+1} \in U_{k+1} \cap U_k \) or \( x_{k+1} \in U_{k+1} - U_k \).

For the latter case, \( x_{k+1} \) is 1-step directly attractable to \( \Omega_{k+1} \) but not 1-step directly attractable to \( \Omega_k \). So there must exist an edge \( e = (x, \sigma, x') \in E^{D^x}(x_{k+1}) \) such that \( x_{k+1} = x \) and \( x' \in V_k - X_m \). Since \( x' \notin X_m \), it follows that \( \rho(x_{k+1}) > \lambda(x') = \rho(x') = \rho(x_k) \).

For the former case, \( x_{k+1} \) is 1-step directly attractable to \( \Omega_k \). However, \( x_{k+1} \notin V_k \) and \( \rho(x_k) = \min_{x \in U_k} \rho_k^0(x) \), so it must be \( \rho_k^0(x_{k+1}) > \rho_k^0(x_k) = \rho(x_k) \). The computation of \( \rho_k^0(x_{k+1}) \) differs from that of \( \rho_k^0(x_{k+1}) \) if there is at least one edge \( e = (x, \sigma, x') \in E^{D^x}(x_{k+1}) \) such that \( x_{k+1} = x \) and \( x' \in V_k - X_m \). If such an edge exists, then \( \rho(x_{k+1}) = \rho_k^0(x_{k+1}) \geq \max((\lambda(x') + c(e)), \rho_k^0(x_{k+1})) > \rho(x_k) \). Such an edge doesn’t exist, then we have \( \rho(x_{k+1}) = \rho_k^0(x_{k+1}), \rho_k^0(x_{k+1}) > \rho(x_k) \).

With Lemma 1 in hand, we are ready to prove our claim in the following theorem.

**Theorem 2:** Given a plant \( G := (X, E, X_m) \) and consider the notation of Algorithm 1. It holds that \( \forall x \in X^{D^x}: \rho(x) = \min_D d_{G^0}(x) \).

**Proof:** Suppose \( \hat{x} \) is the state that belongs to the argument of \( \min_{x \in U_k} \rho_k^0(x) \) in the \( k \)th iteration of Algorithm 1, then we have \( \rho(\hat{x}) = \rho_k^0(\hat{x}) = \max(\rho_k^0(\hat{x}), \rho_k^0(\hat{x})) \). To show \( \rho(\hat{x}) \) is the minimum distance of \( \hat{x} \), we only need to show \( \rho_k^0(\hat{x}) \) is not greater than \( \min_{x \in E_i(\hat{x})} \{\lambda(x') + c(e) \mid e = (\hat{x}, \sigma, x')\} \) since \( \rho_k^0(\hat{x}) \) is fixed for any iteration of Algorithm 1. Note that \( \rho_k^0(\hat{x}) \) is not greater than \( \min_{x \in E_i(\hat{x})} \{\lambda(x') + c(e) \mid e = (\hat{x}, \sigma, x')\} \) such that \( x' \in V_k \) due to the definition of \( \rho_k^0(\hat{x}) \). So we just need to show \( \rho_k^0(\hat{x}) \) is not greater than \( \lambda(\hat{x}) + c((\hat{x}, \sigma, x')) \) for any state \( \hat{x} \notin \Omega_k \).

It is clear that \( \hat{x} \notin X_m \) because \( X_m \subseteq \Omega_k \). It follows that \( \lambda(\hat{x}) = \rho(\hat{x}) \). Since \( \hat{x} \notin \Omega_k \), then \( \rho(\hat{x}) \) is computed in some \( i \)th iteration of Algorithm 1 such that \( i > k \) or \( \rho(\hat{x}) = \infty \). Then it follows from Lemma 1 that \( \rho(\hat{x}) > \rho(\hat{x}) \), so \( \rho_k^0(\hat{x}) \leq \rho(\hat{x}) < \rho(\hat{x}) = \lambda(\hat{x}) + c((\hat{x}, \sigma, x')) \).

Next we present an algorithm that computes an optimal director by applying Algorithm 1 iteratively on different input plants starting from \( G \). During the \( k \)th iteration, the input plant \( G_k \) is processed to derive the next input plant \( G_{k+1} \), which is a sub-plant of \( G_k \). This is conceptually illustrated by Figure 3. (Note that all the uncontrollable events in Figure 3 are disturbance inputs.)

**Algorithm 2:** Input a plant \( G := (X, E, X_m) \), the following steps compute an optimal director \( D^* \).

**Initialization:**
1. Set \( k = 1 \) and \( G_k = G \).
2. Apply Algorithm 1 on \( G_k \). We denote the resulting director and its distance function as \( D_k \) and \( \rho_k \), respectively.
3. \( D^* = D_k \).
Iteration:

1) $I_k := I(G_k^{D_k})$.
2) Let $V_k \subseteq I_k$ be the set of states that belong to the argument of $\max_{x \in I_k} \rho_k(x)$.
3) If $x_0 \in V_k$, then skip the remaining iteration steps.
4) $G_{k+1} := G_k \setminus V_k$, where $G_k \setminus V_k$ represents the operation to reduce $G_k$ to a trim plant in which the states of $V_k$ are removed. This can be accomplished by setting the cost of all incoming edges to $V_k$ to $\infty$ (so that the states of $V_k$ are seen as illegal states) and then applying Algorithm 3 (presented in the Appendix) on the resulting modified $G_k$.
5) If $G_{k+1} = \emptyset$, then skip the remaining iteration steps.
6) Apply Algorithm 1 on $G_{k+1}$. We denote the resulting director and its distance function as $D_{k+1}$ and $\rho_{k+1}$, respectively.
7) If $P(D^*) \geq P(D_{k+1})$, then set $D^* = D_{k+1}$.

Termination:

- If $x_0 \in V_k$ or $G_{k+1} = \emptyset$, then stop. If $P(D^*) = \infty$, then there exists no optimal director, otherwise the optimal director is $D^*$.
- If $x_0 \notin V_k$ and $G_{k+1} \neq \emptyset$, then continue the iteration with $k = k + 1$.

Remark 5: Algorithm 2 iteratively applies Algorithm 1 on different input plants and compares the results. It also uses Algorithm 3 to derive the input plant for the next iteration. It is clear that there is a maximum $|X_m - X_i| + 1$ (resp. $|X_m - X_i|$) number of executions of Algorithm 1 (resp. Algorithm 3) inside Algorithm 2. Note that the complexity of Algorithm 1 and that of Algorithm 3 are both $O(|X| \times |\Sigma|)$. So the overall complexity of Algorithm 2 is $O(|X| \times |\Sigma| \times (|X_m - X_i| + 1))$, which is linear in the number of states, events and non-terminating mark states of $G$.

We present an example to aid the understanding of Algorithm 2.

Example 4: The plant $G$ is the same as in Example 3 and repeats in Figure 4(a). In the initiation, Algorithm 2 applies Algorithm 1 on $G_1 = G$. The resulting plant is repeated in Figure 4(b) with cost of control being $P(D_1) = 11$. From Figure 4(b), we can see $p_{1}(x_4) > p_{1}(x_3) > p_{1}(x_0)$, so in the 1st iteration of Algorithm 2, $\{x_4\}$ is removed and the resulting (trim) plant is denoted as $G_2$ and shown in Figure 4(c). Next we apply Algorithm 1 on $G_2$, whose progress is illustrated in Figure 4(c)-(d), with the text inside parentheses in the captions tracking the progress. Then we have $G_2^{D_2}$ as shown in Figure 4(e) with cost of control being $P(D_2) = 9$.

Since $p_{2}(x_3) > p_{2}(x_0)$, Algorithm 2 then removes $\{x_3\}$. The resulting (trim) plant $G_3$ is empty, so Algorithm 2 terminates with optimal director as $D^* = D_2$.

Remark 6: We use $W_k := X(G_k) - X(G_{k+1})$ to represent the set of states removed (by Algorithm 3) during the $k$th iteration of Algorithm 2 (if $x_0 \notin V_k$, as illustrated by Figure 3. Note that $W_k$ contains not only the states of $V_k$ but also the states that have at least one path visiting $V_k$ in all directed plants and the states that should be trimmed away if the above states are absent. Also note that both $G_k$ and $G_{k+1}$ are trim. So for any director $D$, it holds that $W_k \cap X^D \neq \emptyset$ if and only if $V_k \cap X^D \neq \emptyset$.

Next we prove the correctness of Algorithm 2. We need two lemmas, which assume the following notation: Let $m$ be the last iteration ordinal of Algorithm 2, i.e., Algorithm 2 terminates after the $m$th iteration; let $n$ be the largest iteration ordinal such that $P(D_m) = P(D^*)$, i.e., $n = \max_{0 \leq k \leq m} \{k \mid P(D_k) = P(D^*)\}$.

The first lemma shows that the states removed before the $n$th iteration should not exist in some trim optimally directed plant.

Lemma 2: Given a plant $G$, consider the notation of Algorithm 2 and that for $n$ stated above. If there exists an optimal director for $G$, then there exists an optimal director $D$ for $G$ such that $(X(G_1) - X(G_n)) \cap X^D = \emptyset$.

Proof: We prove by induction on the iteration ordinal $k$ of Algorithm 2 for any $0 \leq k \leq n$. It is clear that the base step trivially holds, i.e., if there exists an optimal director for $G$, then there exists an optimal director $D$ for $G$ such that $(X(G_1) - X(G_1)) \cap X^D = \emptyset$.

For induction hypothesis, assume that there exists an optimal director $D$ such that $(X(G_1) - X(G_k)) \cap X^D = \emptyset$ for any $0 \leq k < n$. We need to show there exists an optimal director $D'$ such that $(X(G_1) - X(G_{k+1})) \cap X^D = \emptyset$.

Assume for contradiction that there exists a state $x_k \in (X(G_1) - X(G_{k+1})) \cap X^D$ for any optimal $D$ such that $(X(G_1) - (X(G_k)) \cap X^D = \emptyset$ and $0 < k < n$. Then $x_k \in W_k \cap X^D$. So it follows (from Remark 6) that there exists a state $x_k' \in V_k \cap X^D$.

Note that $x_k'$ belongs to the argument of $\max_{x \in I_k} \rho_k(x)$. Also note that $P(D_n) \leq P(D_k)$ holds for any $0 < k < n$, so $\rho_k(x_k') = P(D_k) \geq P(D_n)$.

Since $G_k$ is a sub-plant of $G_1 = G$ and $(X(G_1) - X(G_k)) \cap X^D = \emptyset$, we have $\forall x \in X^D : d_{G_k}(x) = d_{G_k}(x)$.
Then it follows from Theorem 2 that $d_{G^0}(x'_k) = d_{G^0_k}(x'_k) \geq \rho_k(x'_k)$.

Therefore $P(D) \geq d_{G^0}(x'_k) \geq \rho_k(x'_k) = P(D) \geq P(D_n)$. Since $D$ is optimal, $P(D) \leq P(D_n)$ also holds, so it follows that $P(D) = P(D_n)$, which implies $D_n$ is optimal. Since $X(G_1) \supseteq X(G_k) \supseteq X(G_n)$ for any $0 < k < n$, we have $(X(G_1) - X(G_{k+1})) \cap X(G_n) = \emptyset$. It is clear that $X_{D_n} \subseteq X(G_n)$, then it follows that $(X(G_1) - X(G_{k+1})) \cap X_{D_n} = \emptyset$, which is a contradiction.

Hence if there exists an optimal director for $G$, then there exists an optimal director $D$ for $G$ such that $(X(G_1) - X(G_{n})) \cap X^D = \emptyset$.

The next lemma shows that certain states removed during and after the $n$th iteration should exist in some trim optimally directed plant.

**Lemma 3:** Given a plant $G$, consider the notation of Algorithm 2 and those for $m$ and $n$ stated above. If there exists an optimal director $D$ for $G$ such that $(X(G_1) - X(G_{n})) \cap X^D = \emptyset$ and $m \neq n$, then $W_m \cap X^D \neq \emptyset$.

**Proof:** We first prove by induction on the iteration ordinal $k$ of Algorithm 2 that $(X(G_1) - X(G_k)) \cap X^D \neq \emptyset$ for $n < k \leq m$.

We show the base step holds, i.e., $(X(G_1) - X(G_n)) \cap X^D \neq \emptyset$. Assume for contradiction that $(X(G_1) - X(G_n)) \cap X^D = \emptyset$.

If $x_0 \in V_m$, then $x_0$ belongs to the argument of max$_{x \in I_m} \rho_m(x)$. Also note that $P(D_n) < P(D_k)$ holds for any $n < k \leq m$. Then it follows from Theorem 2 that $P(D) \geq d_{G^0}(x_0) = d_{G^0_k}(x_0) \geq \rho_m(x_0) = P(D_m) > P(D_n)$, which is a contradiction to the optimality of $D$. If $x_0 \not\in V_m$, then $G_{m+1} = \emptyset$ and we have $W_m = X(G_m) - X(G_{m+1}) = X(G_m)$. Since $x_0 \in X(G_m) \cap X^D$, we have $X(G_m) \cap X^D = W_m \cap X^D \neq \emptyset$. Then it follows (from Remark 6) that $V_m \cap X^D \neq \emptyset$. Pick any state $x_m \in V_m \cap X^D$, then $x_m$ belongs to the argument of max$_{x \in I_m} \rho_m(x)$. It follows that $P(D) \geq d_{G^0}(x_m) = d_{G^0_m}(x_m) \geq \rho_m(x_m) = P(D_m) > P(D_n)$, which is also a contradiction to the optimality of $D$. Therefore $(X(G_1) - X(G_m)) \cap X^D \neq \emptyset$.

For induction hypothesis, assume that $(X(G_1) - X(G_k)) \cap X^D \neq \emptyset$ for any $n+1 < k \leq m$. We need to show $(X(G_1) - X(G_{k-1})) \cap X^D \neq \emptyset$.

Assume for contradiction that $(X(G_1) - X(G_{k-1})) \cap X^D = \emptyset$. So $\forall x \in X^D : d_{G^0}(x) = d_{G^0_{k-1}}(x)$. Since $(X(G_1) - X(G_k)) \cap X^D \neq \emptyset$, there exists a state $x_{k-1} \in (X(G_1) - X(G_k)) \cap X^D = W_{k-1} \cap X^D$. Then it follows (from Remark 6) that there exists a state $x_{k-1} \in V_{k-1} \cap X^D$. Note that $x_{k-1}$ belongs to the argument of max$_{x \in I_{k-1}} \rho_{k-1}(x)$. Then it follows from Theorem 2 that $P(D) \geq d_{G^0}(x_{k-1}) = d_{G^0_{k-1}}(x_{k-1}) \geq \rho_{k-1}(x_{k-1}) = P(D_{k-1}) > P(D_n)$, which is a contradiction to the optimality of $D$. Therefore $(X(G_1) - X(G_{k-1})) \cap X^D \neq \emptyset$.

So we have proved that $(X(G_1) - X(G_k)) \cap X^D \neq \emptyset$ for any $n < k \leq m$. Note that $(X(G_1) - X(G_n)) \cap X^D = \emptyset$. So we
have \((X(G_n) - X(G_{n+1})) \cap X^D \neq \emptyset\). Hence \(W_n \cap X^D \neq \emptyset\).

With Lemma 2 and 3 in hand, the correctness of Algorithm 2 is established as follows.

**Theorem 3:** Given a plant \(G\), consider the notation of Algorithm 2 and those for \(m\) and \(n\) stated above. If there exists an optimal director for \(G\), then the director \(D^*\) computed by Algorithm 2 is optimal.

**Proof:** If \(m = n\) and there exists an optimal director for \(G\), then it follows from Lemma 2 that there exists an optimal director \(D\) for \(G\) such that \((X(G_1) - X(G_n)) \cap X^D = \emptyset\). So \(\forall x \in X^D : d_{G(x)}(x) = d_{G(x)}(x)\).

If \(x_0 \in V_m\), then \(x_0\) belongs to the argument of \(\max_{x \in I_m} \rho_m(x)\). Then it follows from Theorem 2 that \(P(D) \geq d_{G(x)}(x) \geq p_m(x) = P(D_m) = P(D_n) = P(D^*)\). Since \(D\) is optimal, \(P(D) \leq P(D^*)\) also holds, so it follows that \(P(D) = P(D^*)\), which implies \(D^*\) is optimal.

If \(m \neq n\) and there exists an optimal director for \(G\), then it follows from Lemma 2 and 3 that there exists an optimal director \(D\) for \(G\) such that \((X(G_1) - X(G_n)) \cap X^D = \emptyset\) and \(W_n \cap X^D \neq \emptyset\). So \(\forall x \in X^D : d_{G(x)}(x) = d_{G(x)}(x)\). Also it follows (from Remark 6) that there exists a state \(x_m \in V_n \cap X^D\). Note that \(x_m\) belongs to the argument of \(\max_{x \in I_m} \rho_m(x)\). Then it follows from Theorem 2 that \(P(D) \geq d_{G(x)}(x) = d_{G(x)}(x) \geq p_m(x) = P(D_m) = P(D^*)\). Again since \(D\) is optimal, \(P(D) \leq P(D^*)\) also holds, so it follows that \(P(D) = P(D^*)\), which implies \(D^*\) is optimal.

It is clear that Algorithm 2 can be used to check the existence of an optimal director. So existence and synthesis of an optimal director for general plants are both polynomially solvable.

**V. APPLICATION EXAMPLE**

We provide an application example to demonstrate our result. The application is of train-traffic control over a set of track sections. As shown in Figure 5(a), nine sections of tracks labeled from 1 to 9 are separated by some traffic lights and switches. The traffic lights can be used to stop the traffic in the directions as indicated by the arrows above the lights, but have no effect for the traffic in the opposite directions. Suppose initially a train is in section 1. We are required to synthesize an optimal director to control the traffic lights and switches to ensure the train eventually reach section 6 or 9.

We model the movement of the train by a plant \(G\), shown in the Figure 5(b), with state “1” as the initial state, and states “6” and “9” as marked states. The cost associated with each edge of \(G\) is the amount of time taken by the train to go from the source section to the destination section of the edge. Also note that the way traffic lights and switches are placed, the events corresponding to transitions from section 4 to 8 and from section 6 to 7 are (uncontrollable) disturbance inputs, whereas all other transitions are controllable.

We apply Algorithm 2 on \(G\). After initiation, we obtain the directed plant with control cost of 10, shown in Figure 5(c).

**VI. CONCLUSION**

This paper develops an approach with polynomial complexity for the synthesis (and existence) of an optimal director for general plants. This is accomplished in two steps, the first of which finds a “locally optimal” director using a “greedy” search, and the second step iteratively refines the search space for the greedy search. In order to enhance its scope of applicability, future work can incorporate partial observability of events. Further, depending on the application, other notions of optimality may be defined and new algorithms for obtaining new types of optimal director may be developed.

**REFERENCES**


Fig. 5. Synthesis of an optimal directed controller for the application example


VII. APPENDIX

Algorithm 3: Input a plant $G$, the following steps, adapted from [2], generate a reduced trim plant $G'$ such that an optimal director for $G'$ exists if and only if an optimal director for $G$ exists.

1) For each $x \in X$, remove the transitions on $\sigma \in \Sigma_c(x)$ if $c(x, \sigma) = \infty$ and $x \in X_c \Rightarrow \exists \sigma' \in \Sigma_c(x) : c(x, \sigma') \neq \infty$. (Note that this step may result in new terminating states.)

2) $k = 0; \ X_k := \{ x \in X \mid \exists \sigma \in \Sigma_u(x) \text{ such that } c(x, \sigma) = \infty \} \cup \{ x \in X_e \mid \forall \sigma \in \Sigma_c(x) : c(x, \sigma) = \infty \} \cup \{ x \in X \mid \forall s \in \Sigma^* : \alpha(x, s) \notin X_m \}.$

3) $X_{k+1} := X_k \cup \{ x \in X - X_k \mid \exists \sigma \in \Sigma_u(x) : \alpha(x, \sigma) \in X_k \} \cup \{ x \in (X - X_k) \cap X_e \mid \forall \sigma \in \Sigma_c(x) : \alpha(x, \sigma) \in X_k \} \cup \{ x \in X - X_k \mid \forall \sigma \in \Sigma(x) : \alpha(x, \sigma) \in X_k \}.$

4) Repeat Step 3 with $k = k + 1$ until $X_{k+1} = X_k.$
5) Remove all the incoming transitions to the states of $X_k.$
6) Trim the plant and the resulting plant is $G'.$

Algorithm 4: Input a trim acyclic plant $G$, the following steps, taken from [2], computes an optimal director $D^*.$

Initiation:
Set $k = 1, \ \Omega_k = X_{tm}, \ \forall x \in \Omega_k : D^*(x) = \emptyset, \ \rho(x) = 0, \ \Pi(D^*) = 0, \ \forall x \in X_m : \lambda(x) = 0.$

Iteration:
1) Let $U$ be the set of all $1$-step $\Omega_k$-attractable states, and for each $x \in U$, compute the control action:
$$D^*(x) = \begin{cases} \{ x^* \} & x \in X_c \text{ or } \Sigma_u(x) = \emptyset \\ \emptyset & \text{otherwise} \end{cases}$$

where $x^* = (x, \sigma^*, x') \in E_u(x)$ is an edge that belongs to the argument of
$$\min_{e \in E_u(x)} \{ \max_{e' \in E_u(x)} \lambda(x') + c(e) \Pi(D^*(x')) \}$$
(Note that choice of $\sigma^*$ in definition of $D^*(x)$ need not be unique, indicating non-uniqueness of optimal director.)

2) For each $x \in U$ compute:
$$\rho_1(x) = \begin{cases} \lambda(x') + c(e^*) & x \in X_c \text{ or } \Sigma_u(x) = \emptyset \\ 0 & \text{otherwise} \end{cases}$$

$$\rho_2(x) = \begin{cases} \max_{e \in E_u(x)} \{ \lambda(x') + c(e) \} & e = (x, \sigma, x') \in E_u(x) \neq \emptyset \\ 0 & E_u(x) = \emptyset \end{cases}$$

$$\rho(x) = \max(\rho_1(x), \rho_2(x))$$

3) For each $x \in U$, compute the set of edges in the optimally directed graph and the cost of the optimal director rooted at state $x$:
$$E^{D^*}(x) = \{ (x, \sigma, x') \in E \mid \sigma \in D^*(x) \cup \Sigma_u(x) \}$$
$$\Pi(D^*) = \max_{x' \in \Omega_k} \{ \Pi(D^* \setminus x') \mid (x, \sigma, x') \in E^{D^*}(x) \}, \rho(x) \}$$

4) Set $\Omega_{k+1} = \Omega_k \cup U$, and for each $x \in U - X_m$, let $\lambda(x) = \rho(x).$
Termination:

- If $x_0 \in \Omega_{k+1}$, then stop and the optimally directed system is $G^{D^*} := (X^{D^*}, E^{D^*}, X^{D^*}_m)$, where $X^{D^*} := \Omega_{k+1}$ and $X^{D^*}_m := \Omega_{k+1} \cap X_m$.

- If $x_0 \not\in \Omega_{k+1}$, then set $k = k + 1$, and go to iteration step.

**Jing Huang** (S’02) received his B.Sc. and M.Sc. degrees from Xiamen University, Xiamen, China, in 1997 and 2000 respectively. He also received a M.Sc. degree from The University of Texas at Dallas, Richardson, TX in 2003. He is currently a Ph.D. candidate in the Department of Electrical and Computer Engineering at Iowa State University, Ames, IA. His research interests include modeling, specification, verification and control of event-driven, embedded, real-time, fault-tolerant and safety critical systems, formal verification and testing of hardware and software systems, sensor networks, formal methods, etc.

**Ratnesh Kumar** (F’07) received the B.Tech. degree in Electrical Engineering from the Indian Institute of Technology at Kanpur, India, in 1987, and the M.S. and the Ph.D. degree in Electrical and Computer Engineering from the University of Texas at Austin, Texas, in 1989 and 1991, respectively. From 1991-2002 he was on the faculty of University of Kentucky, and since 2002 he is on the faculty of the Iowa State University. He has held visiting position at the Institute of Systems Research at the University of Maryland at College Park, the Applied Research Laboratory at the Pennsylvania State University, the NASA Ames Research Center, and the Argonne National Laboratory—West. His primary research interest is Modeling, Verification, Control, and Diagnosis of Reactive/Event-driven, Real-time, and Hybrid Systems and their applications.

He was a recipient of the Microelectronics and Computer Development (MCD) Fellowship from the University of Texas at Austin, and was awarded the Lalit Narain Das Memorial Gold Medal for the Best EE Student and the Ratan Swapr Up Memorial Gold Medal for the Best All-rounder Student from the Indian Institute of Technology at Kanpur, India. He is a recipient of the NSF Research Initiation Award, NASA-ASEE summer faculty fellowship award, and coauthor of the book *Modeling and Control of Logical Discrete Event Systems*, Kluwer Academic Publishers, 1995. He serves on the program committee for the IEEE Control Systems Society, and the International Workshop on Discrete Event Systems. He is an associate editor of SIAM Journal on Control and Optimization, Journal of Discrete Event Dynamical Systems and IEEE Control Systems Society and is a past editor of IEEE Transactions on Robotics and Automation.