

On Optimal Attraction in Discrete-Event Processes

Yitzhak Brave[†] and Michael Heymann[‡]

September 1990

ABSTRACT

The concept of attraction has been shown to play a basic role in the synthesis of supervisors for stabilizing discrete-event processes. In this paper we define and investigate optimal attraction - optimal in the sense that the cost (or distance) of the convergence path is as small as possible with respect to some distance measure. We find conditions for the existence of supervisors achieving optimal attraction, and provide efficient algorithms for their synthesis.

Key words: discrete-event processes, supervisory control, stabilization, attraction, optimal supervisors.

1. Introduction

The concept of attraction (or stabilization) was introduced in [1], where it was shown to play an important role in the synthesis of supervisors for stabilizing discrete-event processes. Intuitively, the notion of attraction is concerned with the possibility of driving a process (modeled by a state machine or a directed graph) from arbitrary initial states to a prescribed subset of the state set and then keeping it there indefinitely. A similar concept was introduced independently in [2].

In this paper, we are interested in the problem of finding supervisors that achieves optimal attraction in the sense that the cost (or the distance) of the convergence path is as small as possible with respect to some distance measure. To this end, the digraph's edges (representing events) are assigned weights (lengths), and, roughly speaking, the cost is defined as that of the most expensive execution which may be selected by the process (under control) in its progress

[†]Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa 32000, Israel.

[‡]Department of Computer Science, Technion - Israel Institute of Technology Haifa 32000, Israel.

from an initial state to a target state. It turns out that, in general, there need not exist a supervisor that achieves minimal-cost attraction with respect to an arbitrary initial state. However, as will be shown later, under suitable conditions (depending on the process' structure and its edge weights) such an optimal supervisor exists, and we shall provide efficient algorithms for its synthesis.

The paper is organized as follows. In the remainder of this section we give some terminology and notation. In section 2, we give an informal description of the optimal attraction problem. Sections 3 and 4 deal with the existence and synthesis of optimal supervisors for processes whose edge lengths are positive, whereas the general case (i.e., when negative lengths are allowed) is discussed in section 5.

1.1 Processes and Supervisors

A detailed discussion of various aspects concerning attraction in discrete-event processes appears in [1]. We recall here only facts needed for the ensuing development. Let Σ be a finite alphabet. A *process* over Σ is modeled as a finite weighted directed graph (digraph) $G = (V, E)$ where V is a set of states (vertices) and $E \subseteq V \times \Sigma \times V$ is a set of edges. An edge of G is thus an ordered triple $e = (v, \sigma, u) \in E$ and it is said to be directed from v to u . Each edge $e \in E$ is assigned a length $l(e)$. If $(v, \sigma, u) \in E$ we say that v is a *predecessor* of u and u is a *successor* of v . For $v \in V$ and $\Omega \subseteq \Sigma$, let $\delta_G(v, \Omega)$ denote the set of all v 's successors that G can reach by executing a transition labeled $\sigma \in \Omega$. That is

$$\delta_G(v, \Omega) = \{ v' \in V \mid (v, \sigma, v') \in E \wedge \sigma \in \Omega \} . \quad (1.1)$$

For a singleton $\Omega = \{ \sigma \}$ we write $\delta_G(v, \sigma)$. The process G is assumed to be deterministic in the sense that $|\delta_G(v, \sigma)| \leq 1$ for every $v \in V$ and $\sigma \in \Sigma$ (here $|\cdot|$ denotes the cardinality of \cdot).

A *path* is a finite string of edges $s = (v_o, \sigma_1, v_1)(v_1, \sigma_2, v_2) \cdots (v_{n-1}, \sigma_n, v_n)$ with which is associated the unique (state) *trajectory* v_o, v_1, \cdots, v_n . A closed trajectory (i.e., $v_n = v_o$) in which no state (except the start and end states) appears more than once is called a *cycle*. A digraph without cycles is called *acyclic*. The length of a path s , denoted $l(s)$, is the sum of its edges' lengths.

A state v is *reachable* from a state u if there exists a path from u to v . A state v is said to be reachable from a subset of states $A \subseteq V$ if v is reachable from at least one state in A . The *reach* of A in G , denoted $r_G(A)$, is defined as the set of all states in G that are reachable from A . We say that a state v is connected to $A \subseteq V$ if there exists $u \in A$ such that u is reachable from v . The process G is called *A-connected* if each $v \in V$ is connected to A .

For a state $v \in V$, $G-v$ denotes the subprocess of G obtained by deleting v and all edges incident on v from G . If e is an edge in G , then $G-e$ is a subprocess of G obtained by deleting e (without its end states) from G .

Let $G_i = (V_i, E_i), i=1,2$ be two processes over $\Sigma_i, i=1,2$ respectively. The *concurrent composition* of G_1 and G_2 is denoted $G_1 \parallel G_2$, and defined as

$$G_1 \parallel G_2 = (V, E),$$

where $V = V_1 \times V_2$, $\Sigma = \Sigma_1 \cup \Sigma_2$ and $E \subseteq V \times \Sigma \times V$ satisfies the condition that

$$((v_1, v_2), \sigma, (u_1, u_2)) \in E \text{ iff } \begin{cases} (v_1, \sigma, u_1) \in E_1 \text{ and } (v_2, \sigma, u_2) \in E_2 & \text{if } \sigma \in \Sigma_1 \cap \Sigma_2 \\ (v_1, \sigma, u_1) \in E_1 \text{ and } v_2 = u_2 & \text{if } \sigma \in \Sigma_1 - \Sigma_2 \\ (v_2, \sigma, u_2) \in E_2 \text{ and } v_1 = u_1 & \text{if } \sigma \in \Sigma_2 - \Sigma_1 \end{cases} .$$

It can be shown that concurrent composition is an associative and commutative operator (up to isomorphism due to permutation of labels in the composite state space).

As in [3-5] we assume that $\Sigma = \Sigma_c \cup \Sigma_u$; controlled and uncontrolled alphabets (events). Clearly a similar classification is induced on E (i.e., $E = E_c \cup E_u$). A *supervisor* for G is a map $S: V \rightarrow 2^{\Sigma_c}$, and the *closed-loop* process, S/G , is defined as the subprocess (V, E^S) of G satisfying the condition that

$$(\forall e = (v, \sigma, u) \in E) \quad e \in E^S \text{ iff } \sigma \notin S(v). \quad (1.2)$$

In other words, at each state $v \in V$ the supervisor specifies a subset of controlled events that are disabled.

1.2 Attraction

Let $A, B \subseteq V$, $E' \subseteq E$. The subset A is E' -invariant if there is no edge in E' leading out of A . A is a *strong attractor* for B w.r.t. G , denoted $A \stackrel{G}{\leftarrow} B$ if the following conditions are satisfied:

- (a) A is E -invariant.
- (b) Each state $v \in r_G(B)$ is connected to A .
- (c) G has no cycles in $r_{G-A}(B)$.

The notion of strong attraction has the following interpretation: if $A \stackrel{G}{\leftarrow} B$ then G , initialized at state $v \in B$, always reaches A within a finite number of state transitions and remains in A .

We say that A is a *weak attractor* for B w.r.t. G , denoted $A \stackrel{G}{\leftarrow} B$, iff there exists a supervisor S such that $A \stackrel{S/G}{\leftarrow} B$. It was shown in [1] that for each $A \subseteq V$ there exists a unique maximal set weakly attracted by A . This maximal set is denoted $\Omega_G(A)$ and called the *region of weak attraction* of A w.r.t. G . If $\Omega_G(A) = V$, we say that A is a *global weak attractor*.

2. Motivation and Problem Definition

Fix $G = (V, E)$, $A \subseteq V$ such that $G-A$ is acyclic. For each $v \in V-A$, let $L_G(v, A)$ be the set of all paths of G of the form $s=(v_o, \sigma_1, v_1)(v_1, \sigma_2, v_2) \cdots (v_{n-1}, \sigma_n, v_n)$ such that $v_o = v$, $v_n \in A$ and $v_i \notin A$ for all $i=0, 1, \dots, n-1$. That is, $s \in L_G(v, A)$ iff s starts at v , ends in A and every state visited by s , except for its end state, is not in A . The *distance* of v from A w.r.t. G , denoted $d_G(v, A)$ is defined as the length of a path $s \in L_G(v, A)$ whose length is maximal, i.e.

$$d_G(v, A) = \max \{ l(s) \mid s \in L_G(v, A) \} . \quad (2.1)$$

Notice that $L_G(v, A)$ is finite since $G-A$ is acyclic. If $v \in A$, we define $d_G(v, A) = 0$, and if $L_G(v, A) = \emptyset$ we write $d_G(v, A) = \infty$.

Example 2.1 Let $G = (V, E)$ be the process described in Figure 2.1. Here, $V = \{ v_o, v_1, v_2, v_3, v_4 \}$ and $E = \{ (v_o, \phi, v_1), (v_1, \psi, v_o), \dots \}$. Associated with each edge e is a length $l(e)$; thus, e.g., $l((v_2, \gamma, v_o)) = 1$. With the former notation we have

$$L_G(v_3, A) = \{ (v_3, \alpha, v_1), (v_3, \beta, v_2)(v_2, \gamma, v_o), (v_3, \rho, v_4)(v_4, \delta, v_2)(v_2, \gamma, v_o) \}$$

and

$$d_G(v_3, A) = \max \{ 4, 2+1, 1+3+1 \} = 5$$

□

Example 2.2 Consider a small manufacturing system, depicted schematically in Fig. 2.2, consisting of two workstations (WS) and a Buffer (B). WS1 (see Fig. 2.3(a)) takes a workpiece (event α_1), and either successfully completes processing and passes the workpiece to the buffer (event β_1); or returns to its idle state (I_1) and discards the workpiece (event γ_1). WS2 (see Fig. 2.3(b)) either takes a workpiece from B and discards it (event γ_2); or takes a workpiece for processing (event α_2). In the latter case, the event β_2 indicates that WS2 has successfully completed processing. The one slot buffer (see Fig. 2.4) has three states: EMPTY(E), FULL(F) and ERROR(ER). The current content of the buffer is suitable for further processing by WS2 only if it has been passed to the buffer while being in its EMPTY state. So, state ER indicates that the current content of the buffer is faulty.

The concurrent operation of WS1, WS2 and B, denoted by the process $G=(V, E)$, is given by $G = WS1 \parallel B \parallel WS2$. The problem is to synthesize a supervisor S which guarantees that G will always reach the target set $A = \{ (I_1, E, I_2) \}$ in minimal cost (or distance); in fact, we are mainly concerned with the ‘recovery’ of G from error states (i.e.. states of the form (\cdot, ER, \cdot)). □

In general, we shall be interested in the following issues:

- (1) Does there exist a supervisor $S: V \rightarrow 2^{\Sigma_c}$ such that for each $v \in \Omega_G(A)$, $d_{S/G}(v, A)$ is minimal?

- (2) If the answer for (1) is positive, does there exist a supervisor S that solves it and is minimally restrictive in the sense that for every supervisor S' solving (1),

$$\forall v \in \Omega_G(A), \quad S(v) \subseteq S'(v)$$

- (3) When supervisors exist, find efficient algorithms.

We tackle these issues in the following section.

3. Existence of minimally restrictive optimal supervisors

In the sequel we assume (for simplicity) that every weak attractor is global. However the results are readily extended for general weak attractors.

Fix $A \subseteq V$ to be a global weak attractor, and define Ψ to be the set of all supervisors $S: V \rightarrow S^{\Sigma_c}$ such that A is a global strong attractor w.r.t. S/G (i.e., $A \stackrel{S/G}{\Leftarrow}$). Notice that Ψ is nonempty (since A is a global weak attractor) and finite (since V and Σ_c are finite sets). For each $v \in V$, let $m(v, A)$ denote the *minimal distance of v from A* , which can be obtained under control, i.e.,

$$m(v, A) = \min \{ d_{S/G}(v, A) \mid S \in \Psi \}. \quad (3.1)$$

The minimal distance $m(v, A)$ is well defined since Ψ is a nonempty finite set. A natural question that arises is whether there exists $S \in \Psi$ such that

$$(\forall v \in V) \quad d_{S/G}(v, A) = m(v, A). \quad (3.2)$$

Such a supervisor will be called *optimal*.

In general, given a global weak attractor A , we have the following attraction problems:

AP: Synthesize a supervisor $S \in \Psi$ (i.e. a supervisor S satisfying $A \stackrel{S/G}{\Leftarrow}$).

OAP: Synthesize (if possible) an optimal supervisor $S \in \Psi$ (i.e., $A \stackrel{S/G}{\Leftarrow}$ and S satisfies (3.2)).

Since A is a global weak attractor, AP is obviously solvable. A solution for AP was given in [1], and it is, in general, not unique. In what follows we show that OAP is solvable provided all edge lengths are positive, which will be the case throughout this and the next section.

For each $v \in V$ let S_v be a supervisor such that

$$A \stackrel{S_v/G}{\Leftarrow} \{v\} \quad \text{and} \quad d_{S_v/G}(v, A) = m(v, A). \quad (3.3)$$

That is, S_v achieves strong attraction of v from A , with minimal distance. In this case we say that S_v is *optimal relative to v* . Notice that it is not necessarily true that S_v is optimal relative to other states in V . Define the supervisor $S: V \rightarrow 2^{\Sigma_c}$ according to

$$(\forall v \in V) \quad S(v) = S_v(v) \quad , \quad (3.4)$$

where S_v satisfies (3.3). That is, S selects the 'local' evaluation of S_v at state v for each $v \in V$. The following lemmas state that A is a global strong attractor w.r.t. S/G , and that S is optimal.

Lemma 3.1: $A \stackrel{S/G}{\leftarrow}$.

Lemma 3.2 For each $v \in V$,

$$d_{S/G}(v, A) = m(v, A).$$

The following proposition is an immediate consequence of lemmas 3.1 and 3.2.

Proposition 3.3 Assuming all edge lengths are positive, OAP is solvable.

It turns out that, in general, OAP, as well as AP, has many solutions. If, however, one would prefer a minimally restrictive solution, in the sense that it disables as few events as possible, OAP has such a solution, whereas AP does not necessarily have. To illustrate the latter observation, let Φ be the set of all supervisors $S: V \rightarrow 2^{\Sigma^c}$, and let $\Psi^o \subseteq \Psi$ be the set of all supervisors that solve OAP (notice that proposition 3.3 implies that Ψ^o is nonempty). We define the operation \wedge (conjunction) on Φ as follows. For every $S_1, S_2 \in \Phi$ and every $v \in V$,

$$(S_1 \wedge S_2)(v) = S_1(v) \cap S_2(v).$$

Also define \preceq to be a partial order on Φ satisfying the condition that

$$S_1 \preceq S_2 \quad \text{iff} \quad (\forall v \in V) \quad S_1(v) \subseteq S_2(v).$$

It is clear from (1.2) that if $S_1 \preceq S_2$ then S_1 disables no more events than does S_2 . Furthermore, it can be shown that Φ is a complete semilattice, partially ordered by \preceq and with meet operation the conjunction of supervisors. The least element of Φ , $S_{\min}: V \rightarrow 2^{\Sigma^c}$, is given by

$$\forall v \in V, \quad S_{\min}(v) = \emptyset.$$

Definition 3.4 A least element of $\Psi(\Psi^o)$ will be called a *minimally restrictive* solution of AP (OAP).

The following example shows that AP need not have a minimally restrictive solution.

Example 3.1 Let G be the process depicted in Figure 3.1, let S_1 be a supervisor that disables γ at state u , and let S_2 be a supervisor that disables β at state v . Both S_1 and S_2 solve AP, but none of them is minimally restrictive in the sense of definition 3.4. In fact, S_1 , as well as S_2 , are minimal elements of Ψ (i.e., S_i satisfies the condition that for every $S \in \Psi$, $S \preceq S_i$ implies

$S = S_i$, $i=1,2$). Finally, notice that the conjunction supervisor $S_1 \wedge S_2$ is not a solution of AP. \square

As regards OAP, a minimally restrictive solution exists, as we show next. The following lemma states that if S_1 and S_2 are solutions of OAP then their conjunction is a solution of AP.

Lemma 3.5 Let S_1, S_2 such that for $i=1,2$ $A \stackrel{S_i/G}{\leftarrow}$ and

$$(\forall v \in V) \quad d_{S_i/G}(v, A) = m(v, A).$$

Then

$$A \stackrel{(S_1 \wedge S_2)/G}{\leftarrow}.$$

The next lemma claims that the conjunction of two solutions of OAP is optimal, i.e.

Lemma 3.6 Let S_1 and S_2 as in lemma 3.5. Then for every $v \in V$

$$d_{(S_1 \wedge S_2)/G}(v, A) = m(v, A).$$

The consequence of proposition 3.3 and lemmas 3.5 and 3.6 is

Proposition 3.7 The class Ψ^o of supervisors solving OAP is nonempty and closed under \wedge .

Notice that contrary to proposition 3.7, Ψ , the class of all supervisors solving AP, is not closed under \wedge , as was shown in example 3.1. By proposition 3.7, the finite class Ψ^o has a least element, whence OAP has a minimally restrictive solution. In example 3.1, S_2 , the supervisor that disables β at v and nothing elsewhere, is the minimally restrictive optimal supervisor.

4. Effective Algorithm for Solving OAP (positive edge weights)

In developing an effective algorithm for solving OAP we need the following definitions. Let $V' \subseteq V$. We say that $v \in V$ is V' -attractable if $v \notin V'$, v is a predecessor of a state in V' and every uncontrolled edge of G leaving v ends in V' , i.e., if $v \in V - V'$ and

$$(i) \quad V' \cap \delta_G(v, \Sigma) \neq \emptyset, \quad \text{and} \tag{4.1}$$

$$(ii) \quad \delta_G(v, \Sigma_u) \neq \emptyset \Rightarrow \delta_G(v, \Sigma_u) \subseteq V'. \tag{4.2}$$

Also, we say that $v \in V$ is (V', G) -directed if $v \in V - V'$ and $\delta_G(v, \Sigma) \subseteq V'$, (i.e. v is a predecessor of a state in V' and every edge of G leaving v ends in V'). It is easily seen that if v is V' -attractable, there exists a supervisor S such that v is $(V', S/G)$ -directed (i.e. $\delta_{S/G}(v, \Sigma) \subseteq V'$). If

$\beta, \rho \in \Sigma_c$ in example 2.1, v_2 and v_3 are A -attractable, whereas only v_2 is (A, G) -directed. Finally, let $E(v)$ denote the set of all edges (of G) leaving v . Then we have the following

Proposition 4.1 The following algorithm computes the minimal distances $m(v, A)$ and synthesizes a solution for OAP.

Algorithm 4.1

(1) Let $U_o = A$, let

$$\lambda(v) = \begin{cases} 0 & \text{if } v \in A \\ \infty & \text{if } v \in V-A \end{cases}$$

and let

$$S(v) = \begin{cases} \{ \sigma \in \Sigma_c \mid \delta_G(v, \sigma) \not\subseteq A \} & \text{if } v \in A \\ \emptyset & \text{if } v \in V-A \end{cases} \quad (4.3)$$

Iterate steps (2)-(4) until $U_{j+1} = U_j$.

(2) For each U_j -attractable state v compute

$$\rho(v) = \begin{cases} \min \{ \lambda(v') + l(e) \mid e = (v, \sigma, v') \in E(v), v' \in U_j \} & \text{if } E_u \cap E(v) = \emptyset \\ \max \{ \lambda(v') + l(e) \mid e = (v, \sigma, v') \in E_u \cap E(v) \} & \text{otherwise} \end{cases} \quad (4.4a)$$

$$(4.4b)$$

(3) Let V_j be the set of all states v whose $\rho(v)$ is minimal (with respect to all other U_j -attractable states).

(4) Let $U_{j+1} = U_j \cup \{ V_j \}$, and for all $v \in V_j$ set $\lambda(v) = \rho(v)$;

$$S(v) = \{ \sigma \in \Sigma_c \mid \exists e = (v, \sigma, v') \in E(v) \text{ s.t. } \lambda(v') + l(e) > \lambda(v) \} . \quad (4.5)$$

(5) Upon termination, S is the optimal solution of OAP, and $m(v, A) = \lambda(v)$, $\forall v \in V$. □

This algorithm terminates in at most $|V|$ iterations, say k iterations. In each iteration j , a U_j -attractable state v (with minimal value $\rho(v)$) is chosen (step (3)). Thus it can be shown ([1, section 5]) that upon termination $U_k = \Omega_G(A) = V$ (since A is assumed to be a global weak attractor). The following lemma states that the supervisor defined by (4.5) is a solution of AP.

Lemma 4.2 A is a global strong attractor w.r.t. S/G (i.e. $A \stackrel{S/G}{\leftarrow}$).

Next we show that S solves OAP and that the minimal distances are given by $\lambda(v)$, $v \in V$. To this end, assume, throughout the following lemmas, that for all $v' \in U_j$, $j \leq k$

$$\begin{aligned} m(v', A) &= \lambda(v') \\ &= d_{S/G}(v', A). \end{aligned}$$

where S is the supervisor specified in step (4) of the j -th iteration, and consider a state $v \in V_j$ (see step (3)). The following lemma states that S implements $\lambda(v)$, i.e., $\lambda(v)$ is the distance of v from A w.r.t. S/G .

Lemma 4.3

$$\lambda(v) = d_{S/G}(v, A).$$

Next we claim that for each U_j - attractable state w , $\rho(w)$ is a lower bound on the distance of any supervisor that connects w to U_j directly.

Lemma 4.4 Let w be a U_j -attractable state. Then

$$d_{S'/G}(w, A) \geq \rho(w)$$

for any S' solving AP such that

$$\delta_{S'/G}(w, \Sigma) \subseteq U_j.$$

Using lemmas 4.3 and 4.4, it can be shown that $\rho(v)$ is indeed the minimal distance of v from A , i.e.,

Lemma 4.5 Let $v \in V_j$ (see step (3)). Then $\rho(v) = m(v, A)$.

Now proposition 4.1 follows from lemmas 4.2 and 4.5. According to proposition 4.1, algorithm 4.1 provides a solution S for OAP. This solution, as we show below, is minimally restrictive. That is, if S' is another supervisor solving OAP then

$$(\forall v \in V) \quad S(v) \subseteq S'(v).$$

We shall need the following lemma. Let $\{\lambda_j\}_{j=0}^k$, $\lambda_0=0$, $k \leq |V|$, be the sequence consisting of the values $\lambda(v)$ of states v chosen in step (3) of algorithm 4.1. That is, if $v \in V_j$ then $\lambda_j := \lambda(v)$.

Lemma 4.6 The sequence $\{\lambda_j\}_{j=0}^k$ is monotonic non-decreasing, i.e.

$$\lambda_j \leq \lambda_{j+1}, \quad j=0, 1, \dots, k-1.$$

Based on lemma 4.6, the following proposition is easily proved.

Proposition 4.7 The supervisor S constructed by algorithm 4.1 is the minimally restrictive solution of OAP.

Let us now consider the complexity of algorithm 4.1. As we pointed out above, this algorithm terminates in at most $|V|$ iterations. In each iteration j , $\rho(v)$ is computed for each U_j -attractable state v (see step (2)); then a state v with minimal value $\rho(v)$ is chosen. In fact, the selection v in step (3) will not change even if v is chosen from the set of all predecessors of states in U_j (which includes the set of all U_j -attractable states). Furthermore, one can use the value $\rho_{j-1}(v)$ (i.e., the value $\rho(v)$ in the $(j-1)$ iteration) for computing the value $\rho_j(v)$ (i.e. the value $\rho(v)$ in the j -th iteration). Indeed, if v' has been chosen in step (3) of the $(j-1)$ iteration, $\rho_j(v)$ may differ from $\rho_{j-1}(v)$ only if there is an edge $e = (v, \sigma, v') \in E$. In that case

$$\rho_j(v) = \begin{cases} \min \{ \rho_{j-1}(v) , \lambda_{j-1}(v') + l(e) \} & \text{if } E_u \cap E(v) = \emptyset \\ \max \{ \rho_{j-1}(v) , \lambda_{j-1}(v') + l(e) \} & \text{otherwise} \end{cases}$$

We assumed here that G has no parallel edges since it makes no sense to have such edges. Since v' has at most $|V|$ predecessors, steps (3) and (2) can be completed in time $O(|V|)$. Thus, the whole algorithm is of $O(|V|^2)$ complexity.

Example 4.1 Consider the process G described in Figure 4.1. If algorithm 4.1 is applied, the selections in step (3) will be in the following order: v_3, v_1, v_2 and v_0 . The distances from v_3, v_1, v_2 and v_0 from A are, respectively, 2, 10, 11 and 12. The supervisor specified by (4.5) disables α at v_0 , β at v_1 , δ at v_2 , and γ at v_3 . \square

Example 4.2 Consider again the manufacturing system G of example 2.2. Figure 4.2 shows the closed-loop process S/G , where S is determined by (4.5) in algorithm 4.1. The numbers attached to the states are the values $\lambda(v)$ (see step (4)), indicating the minimal distances from v to A .

5. Optimal Attraction and Negative Edge Weights

So far we assumed that the weights associated with the edges of G are positive. However, this may not be the case in certain application (e.g., when G models a discrete event process in the area of economics). The question now of interest is whether previous sections results concerning minimal restrictive optimal supervisors remain valid if negative weights are allowed.

It turns out that in contrary to the case of positive weights, a solution for OAP need not exist. Consider the process of Figure 5.1. It is easily seen that $m(v_1, A) = 2$ and $m(v_2, A) = 3$. In fact, the supervisors S_1 and S_2 , described in Figure 5.2, implement these minimal distances, separately. However, although $S_1(S_2)$ implement minimal distance attraction from $v_1(v_2)$, there is no optimal supervisor achieving $m(v_1, A)$ and $m(v_2, A)$ simultaneously. It is worth noting that the conjunction supervisor $S_1 \wedge S_2$ does not solve OAP, as well as AP; in fact, $((S_1 \wedge S_2) / G) = G$. \square

The following proposition shows that the length of G 's cycles (the length of a cycle is the sum of its edges' length) is crucial in solving OAP.

Proposition 5.1 If the length of every cycle in $G-A$ is positive, OAP is solvable and has a minimally restrictive solution.

This restriction on the length of G 's cycles is not necessary, as it is shown in Figure 5.3. Figure 5.3(a) is the process G , whereas 5.3(b) is the closed-loop process S/G with S being the minimal restrictive optimal supervisor. This supervisor disables β at v_1 and α at v_2 .

Algorithm 4.1, which a natural extension of the algorithm in [1, section 5], is not applicable if negative edge lengths are allowed; lemmas 4.2-4.4 still hold, but lemma 4.5 does not. Next, an algorithm for synthesizing the minimally restrictive solution of OAP is described. it allows negative edge lengths, but does not allow a cycle whose length is nonpositive.

Algorithm 5.2

(1) Let

$$\lambda(v) = \begin{cases} 0 & \text{if } v \in A \\ \infty & \text{otherwise} \end{cases} \quad (5.1)$$

(2) As long as there is a state $v \notin A$ such that $\rho(v) < \lambda(v)$, where

$$\rho(v) = \begin{cases} \min \{ \lambda(v') + l(e) \mid e=(v, \sigma, v') \in E(v) \} & \text{if } E_u \cap E(v) = \emptyset \\ \max \{ \lambda(v') + l(e) \mid e=(v, \sigma, v') \in E_u \cap E(v) \} & \text{otherwise} \end{cases} \quad (5.2)$$

$$(5.3)$$

let $\lambda(v) = \rho(v)$.

(3) Upon termination, let a supervisor S be determined as

$$\forall v \in A, \quad S(v) = \{ \sigma \in \Sigma_c \mid \exists (v, \sigma, v') \in E \text{ s.t. } v' \notin A \} \quad (5.4)$$

$$\forall v \in V-A, \quad S(v) = \{ \sigma \in \Sigma_c \mid \exists e=(v, \sigma, v') \in E \text{ s.t. } \lambda(v) < \lambda(v') + l(e) \} \quad (5.5)$$

□

For our purposes $\infty + l = \infty$ for every $l \in \mathbf{R}$.

For showing termination of algorithm 5.2, we need the following lemma. Let A_j be the set of all states v whose value $\lambda(v)$ is finite at the end of the j -th iteration.

Lemma 5.3 For every $v \in A_j$ (i.e., $\lambda(v)$ is finite) there is a simple path from v to A whose length is $\lambda(v)$.

Since each value $\lambda(v)$ corresponds to at least one simple path from v to A , and since the number of simple paths in a finite digraph is finite, the number of values possible for $\lambda(v)$ is finite. Thus algorithm 5.2 must terminate.

There is a special relation between algorithms 5.2 and 4.1: the value $\rho(v)$ of a state $v \in V-A_j$ (where A_j is the current set of states w whose values $\lambda(w)$ are finite) is finite iff v is

A_j -attractable. This relation is explained as follows. If $E_u \cap E(v) = \emptyset$ (i.e., no uncontrolled transitions emanate v), $\rho(v)$ is finite iff (see (5.2)) v has a successor in A_j iff (see (4.1) and (4.2)) v is A_j -attractable. If, however, $E_u \cap E(v) \neq \emptyset$, $\rho(v)$ is finite iff (5.3) iff $\delta_G(v, \Sigma_u) \subseteq A_j$ iff v is A_j attractable. The consequence of this relation is that the sets A_j of algorithm 5.2 and the sets U_j of algorithm 4.1 (or more precisely, the sets U_j of the algorithm in [1, section 5]) are constructed according to the same rule, namely, augment a new state $v \in V - A_j$ iff v is A_j -attractable. Thus we have:

Lemma 5.4 Upon termination of algorithm 5.2, the set of all states v whose value $\lambda(v)$ is finite is the region of weak attraction, $\Omega_G(A)$.

Recall that A is assumed to be a global weak attractor (i.e., $\Omega_G(A) = V$), and thus, if algorithm 5.2 is applied to a process G , we end up with a finite value $\lambda(v)$ for every $v \in V$. Next we claim that S , the supervisor synthesized in step (3) of algorithm 5.2, indeed implements global strong attraction. That is,

Lemma 5.5 Let S be determined by (5.4) and (5.5). Then

$$A \xleftarrow{S/G} .$$

The next lemma states that S (as determined by (5.4) and (5.5)) is an optimal supervisor, and that each value $\lambda(v)$, $v \in V$, is the minimal distance from v to A .

Lemma 5.6 For every $v \in V$,

$$\lambda(v) = d_{S/G}(v, A),$$

and

$$\lambda(v) = m(v, A).$$

It follows by lemmas 5.5 and 5.6 that the supervisor S constructed by algorithm 5.2 is a solution of OAP. Moreover, it is minimally restrictive since (see (5.5)) enabling any transition $e = (v, \sigma, v')$, with $\lambda(v) < \lambda(v') + l(e)$, will violate the requirement of minimal distance attraction from v . Thus we have:

Proposition 5.7 Assuming $G - A$ has no nonpositive length cycles, the supervisor synthesized by algorithm 5.2 is the minimally restrictive solution of OAP.

For evaluating the complexity of algorithm 5.2 we assume that it is executed as follows. Order the states: $v_1, v_2, \dots, v_{|V-A|}$, where $v_i \in V - A$. Now perform step (2) by first checking v_1 , then v_2 , etc., and improving the values $\lambda(v_i)$ accordingly. The computation of $\rho(v)$, $\forall v \in V$, is of $O(|E|)$ complexity. After the first sweep, go through additional sweeps, until an entire sweep produces no improvement. By lemma 5.3, this process will terminate. Furthermore, if in S/G (where S is the supervisor synthesized by algorithm 5.2) a longest path from v to A consists of n edges, then by the end of the n -th sweep, v will have its final value; this can be proved by induction on n . Since $n \leq |V|$, the whole algorithm is of $O(|V| \cdot |E|)$ complexity.

Moreover, if by the $|V|$ -th sweep any improvement of a label takes place then $G-A$ contains a negative length cycle.

Example 5.1 Consider the process G of Figure 5.1. If algorithm 5.2 is executed as explained above, the values $\rho(v)$ and $\lambda(v)$ will be as described in Table 5.5. Upon termination, the supervisor S , determined by (5.5), disables α at v_0 , β and ρ at v_1 , δ at v_2 , and ϕ at v_3 . \square

6. Conclusion

The problem of synthesizing optimal supervisors for attraction (or stabilization) of discrete-event processes has been defined and investigated. Given a process G , a supervisor S stabilizing G and a state v , a performance measure has been chosen to be the cost (or distance) of the worst-case execution of the supervised process S/G , initialized at state v . Conditions guaranteeing the existence of optimal (and minimally restrictive) supervisors have been provided, as well as efficient algorithms (4.1 and 5.2) for their synthesis. The first algorithm is based on the synthesis procedure presented in [1], and it is suitable only for positive edge weights, whereas the second allows negative weights in the expense of more computations.

Verification and synthesis procedures for related problems can be derived from this work. For example, given a process G and a state v in the region of strong attraction, the distance from v to the set of target states can be computed by a version of algorithm 4.1. In fact, this problem can be solved by assuming that all events are uncontrolled and then using algorithm 4.1. Other examples are synthesizing supervisors achieving optimal attraction within a prespecified region of states (i.e., for each illegal initial state there is a subset of states which G may traverse in its way to the target states), or optimal attraction in processes with state weights in addition to edge weights. Solving the weighted attraction problem with respect to other optimality criteria (such as average-case execution or most probable execution) is an interesting topic for further research.

REFERENCES

- [1] Y. Brave and M. Heymann, "Stabilization of discrete-event processes", *Int. J. Control*, Vol. 51, No. 5, pp. 1101-1117, 1990.
- [2] C. M. Ozveren, A. S. Willsky and P. J. Antsaklis, "Stability and Stabilizability of discrete event dynamic systems", Preprint.
- [3] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete event processes", *SIAM J. on Control and Optimization*, 25(1), pp. 206-230, January 1987.
- [4] P. J. Ramadge and W. M. Wonham, "Modular feedback logic for discrete event systems", *SIAM J. Control and Optimization*, 25(5), pp. 1202-1218, September 1987.
- [5] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language", *SIAM J. Control and Optimization* 25(3), pp. 637-659, May 1987.