Towards Optimal Supervisory Control of Probabilistic Discrete Event Systems

Vera Pantelic and Mark Lawford

SQRL, Department of Computing and Sofware, Faculty of Engineering, McMaster University, Hamilton, ON, Canada L8S 4K1
(e-mail: pantelv at mcmaster dot ca, lawford at mcmaster dot ca)

Abstract: This paper considers optimal supervisory control of probabilistic discrete event systems (PDESs). PDESs are modeled as generators of probabilistic languages. The probabilistic supervisors employed enable/disable events with certain probabilities. We consider the case when there exists no probabilistic supervisor to match the behaviour of a plant to a probabilistic requirements specification. First, we define a notion of distance between two probabilistic generators. Then, given a plant and a desired probabilistic behaviour, we present an algorithm that minimizes the distance between the desired behaviour and the behaviour of the controlled plant achievable under probabilistic control.

Keywords: Supervisory control, stochastic systems, discrete event systems, optimal control.

1. INTRODUCTION

The control of different models of stochastic discrete event systems has been investigated in Mortazavian (1993), Borkar (1991), Kumar and Garg (1998), Chattopadhyay and Ray (2007), etc. Rabin's probabilistic automata are used in Mortazavian (1993) as the underlying model, while Borkar (1991) investigates the optimal control theory of Markov chains. The model of Garg (1992a,b) is used in Kumar and Garg (1998), Chattopadhyay and Ray (2007).

We adopt the supervisory control framework of PDESs as proposed in Lawford and Wonham (1993). PDESs are modeled as probabilistic generators from Garg (1992a,b). A PDES is represented as an automaton with transitions labeled with events and probabilities. As opposed to the model of Garg (1992a,b), our probabilistic automaton is deterministic in the following sense: for each state of the automaton and any event, there is at most one next state to which the automaton can move. The probabilities of all the events in a certain state add up to at most one. Further, deterministic supervisors for DES are generalized to *probabilistic supervisors*. The probabilistic supervisors are so named because they employ the control method of *random disablement*: after observing a string s, the probabilistic supervisor enables an event σ with a certain probability. Standard (deterministic) control can deterministically enable/disable controllable events. The supervisory control problem considered in Lawford and Wonham (1993) is to find, if possible, a supervisor under whose control the behaviour of a plant is identical to a given probabilistic specification. As shown in Lawford and Wonham (1993), a plant under probabilistic control can generate a much larger class of probabilistic languages than deterministic control. The necessary and sufficient conditions for the existence of a supervisor for a class of PDESs are given in Lawford and Wonham (1993). A formal proof of the necessity and sufficiency of the conditions and an algorithm for the calculation of the supervisor, if it exists, are presented in Postma and Lawford (2004), Pantelic et al. (2009).

Analogous to a problem in classical supervisory control theory, it can happen that, given a plant to be controlled and a probabilistic specification language, no probabilistic supervisor exists such that the plant under control generates the pre-specified probabilistic language. In this case, when the exact solution is not achievable, a designer tries to find a supervisor such that the plant generates the behavior closest to the desired behaviour. We consider the non-probabilistic language generated by the specification automaton to be a safety constraint in the standard supervisory control sense similar to Kumar and Garg (1998). We, therefore, generate the supremal controllable sublanguage of the intersection of the plant and specification as the maximal achievable legal non-probabilistic behaviour of the plant under control. Then, we minimize the distance between the achievable probabilistic behavior of the plant under control and the probabilistic behaviour of the specification (contrained to the supremal controllable sublanguage). The distance used is in a pseudometric on the states of probabilistic transition systems.

In Section 2 we present PDES as generators of probabilistic languages, and introduce the probabilistic control of PDES. The proposed pseudometric is presented in Section 3, and two algorithms for the calculation (approximation) of the distances in this pseudometric are presented. Section 4 presents the algorithm for finding the closest approximation to within a prespecified accuracy. Section 5 concludes with avenues for future work. In this paper, the proofs are omitted due to space restrictions. Complete proofs can be found in Pantelic and Lawford (2009).

2. PRELIMINARIES

In this section, we present PDES modeled as generators of probabilistic languages. Then, we introduce the probabilistic control of PDESs, the probabilistic supervisory problem, and the main results of Lawford and Wonham (1993), Postma and Lawford (2004).

2.1 Modeling PDES

The probabilistic DES can be modeled as a probabilistic generator $G = (Q, \Sigma, \delta, q_0, Q_m, p)$ (Lawford and Wonham (1993)), where Q is the nonempty set of states (at most countable), Σ is a finite alphabet whose elements we will refer to as event labels, $\delta : Q \times \Sigma \to Q$ is the (partial) transition function, $q_0 \in Q$ is the initial state, $Q_m \subseteq Q$ is the set of marking states, which represent the completed tasks, and $p : Q \times \Sigma \to [0, 1]$ is the statewise event probability distribution. In this paper, we consider only finite state PDESs (Q is a finite nonempty set). The transition function is traditionally extended by induction on the length of strings to $\delta : Q \times \Sigma^* \to Q$ in a natural way. For a state q, and a string s, the expression $\delta(q, s)!$ will denote that δ is defined for the string s in the state q.

The probability that the event $\sigma \in \Sigma$ is going to occur at the state $q \in Q$ is $p(q, \sigma)$. For the generator G to be well-defined, (i) $p(q, \sigma) = 0$ should hold if and only if $\delta(q, \sigma)$ is undefined, and (ii) $\forall q \; \sum_{\sigma \in \Sigma} p(q, \sigma) \leq 1$. The probabilistic generator G is nonterminating if, for every reachable state $q \in Q$, $\sum_{\sigma \in \Sigma} p(q, \sigma) = 1$. Conversely, G is terminating if there is at least one reachable state $q \in Q$ such that $\sum_{\sigma \in \Sigma} p(q, \sigma) < 1$. The probability that the system terminates at state q is $1 - \sum_{\sigma \in \Sigma} p(q, \sigma)$. Throughout the sequel, we will mostly consider nonterminating generators (if a plant is terminating, it can easily be transformed into a nonterminating one using the technique described in Lawford and Wonham (1993)).

The language $L(G)$ generated by a probabilistic DES automaton $G = (Q, \Sigma, \delta, q_0, Q_m, p)$ is $L(G) = \{s \in \Sigma^* \mid \delta(q_0, s)!\}$. The probabilistic language generated by G is defined as:

$$L_p(G)(\epsilon) = 1$$

$$L_p(G)(s\sigma) = \begin{cases} L_p(G)(s) \cdot p(\delta(q_0, s), \sigma) & \text{if } \delta(q_0, s)! \\ 0 & \text{otherwise} \end{cases}$$

Informally, $L_p(G)(s)$ is the probability that the string s is executed in G. Also, $L_p(G)(s) > 0$ iff $s \in L(G)$.

For each state $q \in Q$, we define the function $\rho_q : \Sigma \times Q \to [0, 1]$ such that for any $q' \in Q$, $\sigma \in \Sigma$, we have $\rho_q(\sigma, q') = p(q, \sigma)$ if $q' = \delta(q, \sigma)$, and 0 otherwise. The function ρ_q is a probability distribution on the set $\Sigma \times Q$. Also, for a state q, we define *the set of possible events* to be $Pos(q) := \{\sigma \in \Sigma | p(q, \sigma) > 0\}$.

2.2 Probabilistic Supervisors: Existence and Synthesis

As in classical supervisory theory, the set Σ is partitioned into Σ_c and Σ_u, the sets of controllable and uncontrollable events, respectively. Deterministic supervisors for DES are generalized to *probabilistic supervisors*. The control technique used is called *random disablement*. Instead of deterministically enabling or disabling controllable events, probabilistic supervisors enable them with certain probabilities. This means that, upon reaching a certain state q, the control pattern is chosen according to supervisor's probability distributions of controllable events. Consequently, the controller does not always enable the same events when in the state q.

For a PDES $G = (Q, \Sigma, \delta, q_0, Q_m, p)$, a *probabilistic supervisor* is a function $V_p : L(G) \times \Sigma \to [0, 1]$ such that

$$(\forall s \in L(G))(\forall \sigma \in \Sigma) V_p(s, \sigma) = \begin{cases} 1 & \text{if } \sigma \in \Sigma_u \\ x_{s,\sigma} \in [0, 1] & \text{otherwise.} \end{cases}$$

Therefore, after observing a string s, the supervisor enables the event σ with probability $V_p(s, \sigma)$. After a set of controllable events to be enabled has been decided upon (uncontrollable events are always enabled), the system acts as if supervised by a deterministic supervisor. Given sets A, B, we will denote power set of A by $\mathcal{P}(A)$, and the set difference of A and B by $A \backslash B$. Let $q \in Q$ be the state of the plant after $s \in L(G)$ has been observed. The probability that the event $\alpha \in \Sigma$ will occur after string s has been observed is equal to:

$$P(\alpha \text{ in } V_p/G|s) = \sum_{\Theta \in \mathcal{P}(Pos(q) \cap \Sigma_c)} P(\alpha | V_p \text{ enables } \Theta \text{ after } s) \cdot P(V_p \text{ enables } \Theta | s)$$

where

$$P(\alpha | V_p \text{ enables } \Theta \text{ after } s) = \begin{cases} \dfrac{p(q, \alpha)}{\displaystyle\sum_{\sigma \in \Theta \cup \Sigma_u} p(q, \sigma)} & \text{if } \alpha \in \Theta \cup \Sigma_u \\ 0 & \text{otherwise} \end{cases}$$

$$P(V_p \text{ enables } \Theta | s) = \prod_{\sigma \in \Theta} V_p(s, \sigma) \quad \cdot \prod_{\sigma \in (Pos(q) \cap \Sigma_c) \backslash \Theta} (1 - V_p(s, \sigma))$$

The goal is to match the behaviour of the controlled plant with a given probabilistic specification language. We call this problem the *Probabilistic Supervisory Control Problem (PSCP)*. More formally, given a plant PDES G_1 and a specification PDES G_2, we want to find, if possible, a probabilistic supervisor V_p such that $L_p(V_p/G_1) = L_p(G_2)$. An example of probabilistic generators representing a plant and a requirements specification is shown in Fig. 1. Controllable events are marked with a bar on their edges.

We now present the conditions for the existence of a probabilistic supervisor for PSCP (that were first presented in Lawford and Wonham (1993)).

Theorem 1. Let $G_1 = (Q, \Sigma, \delta_1, q_0, Q_m, p_1)$ and $G_2 = (R, \Sigma, \delta_2, r_0, R_m, p_2)$ be two nonterminating PDESs with disjoint state sets Q and R. There exists a probabilistic supervisor V_p such that $L_p(V_p/G_1) = L_p(G_2)$ iff for all $s \in L(G_2)$ there exists $q \in Q$ such that $\delta_1(q_0, s) = q$ and, letting $r = \delta_2(r_0, s)$, the following two conditions hold:

(i) $Pos(q) \cap \Sigma_u = Pos(r) \cap \Sigma_u$, and for all $\sigma \in Pos(q) \cap \Sigma_u$,

$$\frac{p_1(q, \sigma)}{\displaystyle\sum_{\alpha \in \Sigma_u} p_1(q, \alpha)} = \frac{p_2(r, \sigma)}{\displaystyle\sum_{\alpha \in \Sigma_u} p_2(r, \alpha)}$$

(ii) $Pos(r) \cap \Sigma_c \subseteq Pos(q) \cap \Sigma_c$, and, if $Pos(q) \cap \Sigma_u \neq \emptyset$, then for all $\sigma \in Pos(q) \cap \Sigma_c$,

$$\frac{p_2(r, \sigma)}{p_1(q, \sigma)} \sum_{\alpha \in \Sigma_u} p_1(q, \alpha) + \sum_{\alpha \in Pos(q) \cap \Sigma_c} p_2(r, \alpha) \leq 1.$$

If the conditions of Theorem 1 are satisfied, the supervisor can be synthesized via fixpoint iteration as presented in Postma and Lawford (2004), Pantelic et al. (2009).

The results to be presented are for prefix closed probabilistic specification languages so, in the sequel, we simplify the probabilistic generator $G = (Q, \Sigma, \delta, q_0, Q_m, p)$ to $G = (Q, \Sigma, \delta, q_0, p)$.

3. THE METRIC FOR THE CLOSEST APPROXIMATION

In this section, we roughly introduce research done on metrics on states of probabilistic systems, and present the chosen metric.

3.1 Literature review

Probabilistic bisimulation is commonly used to define an equivalence relation between probabilistic systems. However, probabilistic bisimulation is hardly a robust relation: roughly speaking, two states of probabilistic systems are bisimilar if and only if they have the same transitions with exactly the same probabilities to states in the same equivalence classes. The formal definition follows and represents a modified version of the definition of bisimulation given in Barrett and Lafortune (1997).

Definition 1. Let $G = (Q, \Sigma, \delta, q_0, p)$ be a PDES. Probabilistic bisimulation on Q is the binary relation $\equiv$ such that for any $q_1 \equiv q_2$ and $\sigma \in \Sigma$, the following holds:

(1) For every q_1' such that $\delta(q_1, \sigma) = q_1'$, there is q_2' such that $\delta(q_2, \sigma) = q_2'$, $p(q_1, \sigma) = p(q_2, \sigma)$, and $q_1' \equiv q_2'$.
(2) For every q_2' such that $\delta(q_2, \sigma) = q_2'$, there is q_1' such that $\delta(q_1, \sigma) = q_1'$, $p(q_1, \sigma) = p(q_2, \sigma)$, and $q_1' \equiv q_2'$.

States q_1 and q_2 are probabilistic bisimilar if there exists a probabilistic bisimulation $\equiv$ such that $q_1 \equiv q_2$.

As a more flexible way to compare probabilistic systems, a notion of *pseudometric* is introduced. A *pseudometric on a set of states* Q is a function $d : Q \times Q \to \mathbb{R}$ that defines a distance between two elements of Q, and satisfies the following conditions: $d(x, y) \geq 0$, $d(x, x) = 0$, $d(x, y) = d(y, x)$, and $d(x, z) \leq d(x, y) + d(y, z)$, for any $x, y, z \in Q$. If all distances are not greater than 1, the pseudometric is *1-bounded*. In the sequel, we will use the terms metric and pseudometric interchangeably.

The work of Deng et al. (2006) introduces a pseudometric on states for a large class of probabilistic automata, including reactive and generative probabilistic automata. The pseudometric is based on the Kantorovich metric on distributions (also known as the Hutchinson metric). The metric is characterized as the greatest fixed point of a function. Two states are at distance 0 in this metric if and only if they are probabilistic bisimilar. For reactive systems, the work of Deng et al. (2006) is closely related to Desharnais et al. (2002), van Breugel and Worrell (2001). This is the metric we are to use in the solution of our problem. The metric intuitively matches our notion of the distance between PDESs, and accounts for all differences between corresponding transition probabilities. Furthermore, as the metric is suggested for a large class of systems, it allows for an extension of our work to e.g., nondeterministic systems. Also, as it turns out, there is a simple algorithm to compute distances in this metric for our generative, deterministic model. For a more detailed discussion on metrics, see Pantelic and Lawford (2009).

3.2 The metric

First, we introduce some notation. Let $G = (Q, \Sigma, \delta, q_0, p)$ be a nonterminating PDES, where $Q = \{q_0, q_1, \ldots q_{N-1}\}$. This is the system we shall be using throughout the sequel. Let $q_q, q_r \in Q$, and let ρ_{q_q} and ρ_{q_r} be the distributions on $\Sigma \times Q$ induced by the states q_q and q_r, respectively. Assume $0 \leq i, j \leq N-1$, $\Psi = Pos(q_q) \cup Pos(q_r)$, $\sigma \in \Sigma$. For notational convenience, we will write $\rho_{\sigma,i}$ instead of $\rho_{q_q}(\sigma, q_i)$, and, similarly, $\rho'_{\sigma,j}$ instead of $\rho_{q_r}(\sigma, q_j)$. Next, we present the slightly changed pseudometric of Deng et al. (2006) suggested for a large class of automata which includes our generator.

First, in Desharnais et al. (2002) and Deng et al. (2006), the class $\mathcal{M}$ of 1-bounded pseudometrics on states is defined with the ordering

$$d_1 \preceq d_2 \text{ if } \forall s, t \; d_1(s, t) \geq d_2(s, t). \tag{1}$$

Further, it is proved that $(\mathcal{M}, \preceq)$ is a complete lattice.

Next, let $d \in \mathcal{M}$, and let the constant $e \in (0, 1]$ be a *discount factor* that determines the degree to which the difference in the probabilities of farther transitions is discounted: the smaller the value of e, the greater the discount on future transitions. We assume that the total mass of ρ_{q_q} is greater or equal to the total mass of ρ_{q_r}, $\sum_{\substack{\sigma\in\Psi \\ 0\leq i\leq N-1}} \rho_{\sigma,i} \geq \sum_{\substack{\sigma\in\Psi \\ 0\leq i\leq N-1}} \rho'_{\sigma,i}$. This assumption is not needed for nonterminating automata. Then, the distance between the distributions ρ_{q_q} and ρ_{q_r}, $d(\rho_{q_q}, \rho_{q_r})$, (note a slight abuse of notation) is given as:

$$\text{Maximize} \sum_{\substack{\sigma\in\Psi \\ 0\leq i\leq N-1}} a_{\sigma,i}\rho_{\sigma,i} - \sum_{\substack{\sigma\in\Psi \\ 0\leq i\leq N-1}} a_{\sigma,i}\rho'_{\sigma,i} \tag{2}$$

$$\begin{aligned} \text{subject to} \quad & 0 \leq a_{\sigma,i} \leq 1, && \sigma \in \Psi,\ 0 \leq i \leq N-1 \\ & a_{\sigma,i} - a_{\alpha,j} \leq c^{\sigma\alpha}_{ij}, && \sigma, \alpha \in \Psi,\ 0 \leq i, j \leq N-1 \\ & \text{where} && \\ & c^{\sigma\alpha}_{ij} = \begin{cases} e \cdot d(q_i, q_j) & \text{if } \sigma = \alpha \\ 1 & \text{otherwise} \end{cases} && \end{aligned}$$

If the total mass of ρ_{q_q} is less than the total mass of ρ_{q_r}, $d(\rho_{q_q}, \rho_{q_r})$ is defined to be $d(\rho_{q_r}, \rho_{q_q})$. This extension to distributions is also a 1-bounded pseudometric, and is consistent with the ordering (1) (see Desharnais et al. (2002), van Breugel and Worrell (2001)).

The pseudometric on states, d_{max}, is, then, given as the greatest fixed-point of the function $\mathcal{D}$ on $\mathcal{M}$ (here we give a simplified version of that in Deng et al. (2006)):

$$\mathcal{D}(d)(q_q, q_r) = d(\rho_{q_q}, \rho_{q_r}), \quad d \in \mathcal{M}, q_q, q_r \in Q \tag{3}$$

Compared to Deng et al. (2006), the pseudometric on distributions (2) is changed so that the distances between states in our pseudometric are by the factor $1/e$ larger than these in pseudometric of Deng et al. (2006) (see Pantelic and Lawford (2009)). The proofs that the function defined by (3) is monotone on $\mathcal{M}$, and that it has the greatest fixed point follow straightforwardly from Desharnais et al. (2002).

Next, let $i(q_q, \sigma) = i$ such that $q_i = \delta(q_q, \sigma)$ if $\delta(q_q, \sigma)!$, and $i(q_q, \sigma) = 0$, otherwise. Similarly, $j(q_r, \sigma) = j$ such that $q_j = \delta(q_r, \sigma)$ if $\delta(q_r, \sigma)!$, and $j(q_r, \sigma) = 0$, otherwise.

For readability purposes, we will write i instead of $i(q_q, \sigma)$, and j instead of $j(q_r, \sigma)$. The function $\mathcal{D}(d)$ for our model can be shown to be (see Pantelic and Lawford (2009)):

$$\mathcal{D}(d)(q_q, q_r) = \sum_{\sigma \in \Psi} max(\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}, c_{ij}\rho_{\sigma,i}) \quad (4)$$

where $c_{ij} = e \cdot d(q_i, q_j)$. We arbitrarily choose $i(q_q, \sigma)$ to be 0 (similarly for $j(q_r, \sigma)$) when $\delta(q_q, \sigma)$ is not defined although we could have chosen any other $i \in \{1, \ldots, N-1\}$. This is because when $\delta(q_q, \sigma)!$ does not hold, then $\rho_{\sigma, i(q_q,\sigma)} = 0$ for any $i(q_q, \sigma) \in \{0, \ldots, N-1\}$.

The pseudometric d_{max} is defined on the states of a single PDES. The distance between two PDESs (with disjoint sets of states) is the distance between their initial states in a new PDES that represents the union of the PDESs as defined in Section 4.1.

3.3 Calculating the Pseudometric: Algorithms

For $e \in (0, 1)$, we will prove that the function $\mathcal{D}$ has only one fixed point, d^*, and, consequently, $d_{max} = d^*$. Then, we suggest two algorithms for calculating the distances in our metric d_{max}.

First, we introduce some useful definitions and results from linear algebra. A real $n \times n$ matrix $A = (a_{ij})$ defines a linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$, and we will write $A \in L(R^n)$ to denote either the matrix or linear function, as we shall make no distinction between the two. Also, the absolute value of column vector $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ will be denoted by $|x|$, and defined as $|x| = (|x_1|, \ldots, |x_n|)^T$.

Now, let $d \in \mathcal{M}$. Next, we define the function $\mathcal{V}$ on $\mathcal{M}$:

$$\mathcal{V}(d) = (d(q_0, q_0), d(q_0, q_1), \ldots, d(q_{N-1}, q_{N-1}))^T.$$

Note that the vector $\mathcal{V}(d)$ could be further cut down, as $d(s, s) = 0$ and $d(s, t) = d(t, s)$ for any $s, t \in Q$. However, for ease of presentation, we will not decrease the size of the vector. Therefore, $\mathcal{V}(d) = (\mathcal{V}_1(d), \mathcal{V}_2(d), \cdots, \mathcal{V}_{N^2}(d))^T$, where $\mathcal{V}_k(d)$ for $k \in \{1, \ldots, N^2\}$ is given as:

$$\mathcal{V}_k(d) = d(q_i, q_j), \; i = k \; div \; (N+1), \; j = (k-1) \; mod \; N.$$

Now, we redefine the function $\mathcal{D}$ in a natural way as $\mathcal{D}(\mathcal{V}(d)) = (\mathcal{D}_1(\mathcal{V}(d)), \ldots, \mathcal{D}_{N^2}(\mathcal{V}(d)))^T$, where for any $k \in \{1, \ldots, N^2\}$:

$$\mathcal{D}_k(\mathcal{V}(d)) = d(\rho_{q_i}, \rho_{q_j}), \;\; i = k \; div \; (N+1), \qquad (5)$$
$$j = (k-1) \; mod \; N.$$

Further, let $D_0 = \{\mathcal{V}(d) | d \in \mathcal{M}\}$.

Theorem 2. For any $\mathfrak{d}^0 \in D_0$, the sequence

$$\mathfrak{d}^{n+1} = \mathcal{D}(\mathfrak{d}^n), \;\; n = 0, 1, \ldots$$

converges to the only fixed point of $\mathcal{D}$ in D_0, $\mathfrak{d}^* = \mathcal{V}(d^*)$, and the error of convergence is given componentwise as:

$$|\mathfrak{d}^n_k - \mathfrak{d}^*_k| \leq (1-e)^{-1}e^n, \;\; n = 1, 2, \ldots$$

Now, using the presented analysis, we suggest the following two algorithms for the calculation of the distances between the states of PDESs in the chosen pseudometric.

Algorithm 1 Theorem 2 proves that the system of equations

$$\mathfrak{d} = \mathcal{D}(\mathfrak{d}) \qquad (6)$$

has a unique solution. The equations are linear. Therefore, the system (6) can be rewritten into the standard form $A\mathfrak{d} = b$, where A is a $N^2 \times N^2$ matrix and b is a column vector of dimension N^2. Therefore, the distances in our pseudometric can be calculated by solving this system of linear equations. The distances found are exact solutions (if we disregard the round-off error).

Algorithm 2 Theorem 2 suggests an iterative algorithm to approximate distances between the states of a probabilistic generator. The algorithm is a straightforward modification of that of van Breugel and Worrell (2001) that calculates distances in a pseudometric suggested for a different kind of probabilistic system and is derived by using terminal coalgebras. Let $d^0(q_q, q_r) = 0$ for any two states $q_q, q_r \in Q$. As before, let ρ_{q_q} and ρ_{q_r} be the distributions induced by the states q_q and q_r, respectively. Assume $\Psi = Pos(q_q) \cup Pos(q_r)$. The n-th iteration of the algorithm calculates the pseudometric d^n $(n \geq 1)$, where, for any $q_q, q_r \in Q$:

$$d^n(q_q, q_r) = \sum_{\sigma \in \Psi} max(\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}, c_{ij}\rho_{\sigma,i}) \qquad (7)$$

where $c_{ij} = e \cdot d^{n-1}(q_i, q_j)$, and $i = i(q_q, \sigma)$ and $j = j(q_r, \sigma)$ are defined as in (4). The accuracy of the solution found in n-th iteration is $(1-e)^{-1}e^n$.

The iterative method can be useful for systems with large K, where the direct method can be rather expensive. Furthermore, the mathematical apparatus used to reach the iterative method will be reused in the solution of the closest approximation problem.

4. FINDING THE CLOSEST APPROXIMATION

Next, we first characterize the closest approximation and, then, give the algorithm that calculates it with a certain accuracy. All the results in the sequel hold for $e \in (0, 1)$.

4.1 Characterizing The Closest Approximation

First, we repeat the formulation of the nearest approximation problem. Assume that the plant is given as PDES $G_p = (Q_p, \Sigma, \delta_p, q_{p0}, p_p)$, and the requirements specification is given as $G_r = (Q_r, \Sigma, \delta_r, q_{r0}, p_r)$. If there is no probabilistic supervisor V_p such that $L_p(V_p/G_p) = L_p(G_r)$, we seek the optimal (closest) solution and characterize it as follows. The conditions (i) and (ii) of the Theorem 1 for the existence of probabilistic supervisor consist of two parts. The first part of both conditions corresponds to controllability as used in classical supervisory theory (namely, the condition $Pos(q) \cap \Sigma_u = Pos(r) \cap \Sigma_u$ of (i), and $Pos(r) \cap \Sigma_c \subseteq Pos(q) \cap \Sigma_c$ of (ii)). The remaining equations and inequalities correspond to the conditions for probability matching. Hence, before we start looking for the closest approximation in the sense of probability matching, we resort to the classical supervisory theory of supremal controllable languages. We first find $L(G_p) \cap L(G_r)$, and then the supremal controllable sublanguage of $L(G_p) \cap L(G_r)$ (with respect to G_p), K. Then, the DES that represents this language K, further equipped with p_p distribution (appropriately normalized) becomes the modified plant PDES G_1, and the same DES (corresponding to the supremal controllable language K) equipped with the distribution p_r appropriately normalized becomes the desired behaviour PDES G_2. Formally, let (reachable and deadlock-free) DES $G = (Q, \Sigma, \delta, q_0)$ represent

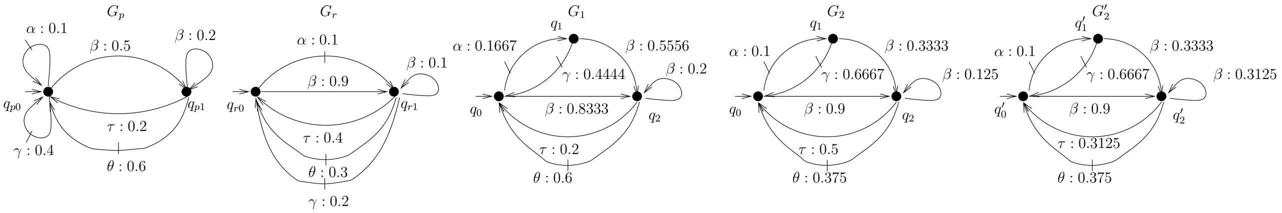

Fig. 1. Plant G_p, requirements specification G_r, PDESs G_1, G_2, and optimal approximation G_2'

the supremal controllable language K. We define PDESs $G_1 = (Q, \Sigma, \delta, q_0, p)$, and $G_2 = (Q, \Sigma, \delta, q_0, p_2)$, where the distributions $p, p_2 : Q \times \Sigma \to [0, 1]$, for any $q \in Q, \sigma \in \Sigma$, are defined as:

$$p(q, \sigma) = \frac{p_p(q_p, \sigma)}{\sum\limits_{\sigma \in \{\sigma \in \Sigma | \delta(q,\sigma)!\}} p_p(q_p, \sigma)}, \quad p_2(q, \sigma) = \frac{p_r(q_r, \sigma)}{\sum\limits_{\sigma \in \{\sigma \in \Sigma | \delta(q,\sigma)!\}} p_r(q_r, \sigma)}$$

where $q_p = \delta_p(q_{p_0}, s)$ for $s \in L(G)$ such that $q = \delta(q_0, s)$, and $q_r = \delta_r(q_{r_0}, s)$ for $s \in L(G)$ such that $q = \delta(q_0, s)$. Note that p and p_2 are well-defined as we do not perform state minimization on automaton representing the supremal controllable language K.

Next, we check probability matching equations and inequalities from Theorem 1. If they are not satisfied (there is no probabilistic supervisor V_p such that $L_p(V_p/G_1) = L_p(G_2)$), we seek $G_2' = (Q', \Sigma, \delta', q_0', p')$ such that there exists a probabilistic supervisor V_p so that $L_p(V_p/G_p) = L_p(G_2')$ holds, and G_2' is closest to G_2 in the chosen metric. Without loss of generality, we assume that $Q \cap Q' = \emptyset$. Also, without loss of generality, we assume that the nonprobabilistic automata underlying G_2 and G_2' are isomorphic (with labeling of events being preserved). Therefore, the automata underlying PDESs G_2 and G_2' are identical up to renaming of states. This assumption is not restrictive as there cannot be any string in the desired system that does not belong to $L(G_2)$. Our goal is to find the probabilities of G_2' as the closest approximation such that the distance between the initial states of G_2 and G_2' is minimized. As our metric is defined on the states of a system, in order to define distances between the states of G_2 and G_2', we can consider the union PDES $G_u = (Q \cup Q', \Sigma, \delta_u, q_0, p_u)$ defined as:

$$\delta_u(q, \sigma) = \delta(q, \sigma), p_u(q, \sigma) = p(q, \sigma) \text{ if } q \in Q \text{ and } \sigma \in \Sigma,$$
$$\delta_u(q, \sigma) = \delta'(q, \sigma), p_u(q, \sigma) = p'(q, \sigma) \text{ if } q \in Q' \text{ and } \sigma \in \Sigma.$$

Then, $\mathcal{M}$ is the set of 1-bounded pseudometrics on the states of the joined system with the same ordering as in (1). Further, note that nonprobabilistic automata underlying G_1 and G_2 are also isomorphic (with the isomorphism function being the identity function). We next note that, considering the isomorphism between nonprobabilistic versions of G_2 and G_2', we will be interested only in the distances between (probability measures on) states $q \in Q$ of G_2 and $q' = f(q) \in Q'$ of G_2', where $f : Q \to Q'$ is the isomorphism between nonprobabilistic automata underlying G_2 and G_2'. We assume that, after the occurrence of string $s \in L(G_1)$, the PDES G_1 is in state q ($\delta(q_0, s) = q$). Then, G_2 and its closest approximation G_2' are in states q and q', respectively, and $q' = f(q)$. Next, we define a class $\mathcal{A}$ of partial functions $b : Q \times Q' \to [0, 1]$, such that $\forall q \in Q, q' = f(q) \in Q'$ $b(q, q') = d(q, q')$, where $d \in \mathcal{M}$. Therefore, the class $\mathcal{A}$ is the class of all 1-bounded pseudometrics with domain reduced to $Q \times Q'$, and only distances between $q \in Q$ and $q' = f(q) \in Q'$ defined. Also, we introduce the reversed ordering on $\mathcal{A}$ to match the one in (1): $d_1 \preceq' d_2$ if $\forall q \in Q$ $\forall q' = f(q) \in Q'$ $d_1(q, q') \geq d_2(q, q')$.

Let $d \in \mathcal{A}$. Let $Q = \{q_0, q_1, \ldots q_{N-1}\}$, and $Q' = \{q_0', q_1', \ldots q_{N-1}'\}$, where $q_i' = f(q_i)$, $i = 0, \ldots, N-1$. Further, we are now able to substantially simplify the notation used in the previous section. Let ρ_q be the probability distribution induced by the state $q \in Q$ of G_2 and let $\rho'_{q'}$ be the probability distribution induced by the state $q' \in Q'$. Assume $\Psi = Pos(q)$, $\Psi_u = Pos(q) \cap \Sigma_u$, and $\Psi_c = Pos(q) \cap \Sigma_c$. Also, for $\sigma \in \Psi$, we will write i instead of $i(q, \sigma) \in \{0, \ldots, N-1\}$, ρ_σ instead of $\rho_q(\sigma, q_i)$, ρ'_σ instead of $\rho'_{q'}(\sigma, q_i')$ and p_σ instead of $p(q, \sigma)$. It can be shown (see Pantelic and Lawford (2009)) that relevant minimal achievable distances in our pseudometric correspond to the greatest fixed point of the function $\mathcal{P} : \mathcal{A} \to \mathcal{A}$ defined as ($q \in Q$ and $q' = f(q) \in Q'$):

$$\mathcal{P}(d)(q, q') = \underset{\rho'_\sigma}{\text{Minimize}} \sum_{\sigma \in \Psi} max(\rho_\sigma - \rho'_\sigma + c_i \rho'_\sigma, c_i \rho_\sigma), \tag{8}$$

where, for each $\sigma \in \Psi, c_i = e \cdot d(q_i, q_i')$ s.t. $q_i = \delta(q, \sigma)$

subject to

$$\frac{p_\sigma}{\sum\limits_{\alpha \in \Psi_u} p_\alpha} = \frac{\rho'_\sigma}{\sum\limits_{\alpha \in \Psi_u} \rho'_\alpha}, \qquad \sigma \in \Psi_u, \tag{9}$$

$$\frac{\sum_{\alpha \in \Psi_u} p_\alpha}{p_\sigma} \rho'_\sigma + \sum_{\alpha \in \Psi_c} \rho'_\alpha \leq 1, \quad \sigma \in \Psi_c, \tag{10}$$

$$0 \leq \rho'_\sigma \leq 1, \qquad \sigma \in \Psi, \tag{11}$$

$$\sum_{\alpha \in \Psi} \rho'_\alpha = 1 \tag{12}$$

Hence, for $q' \in Q'$, distribution $\rho'_{q'}$ is given by the values of decision variables $\rho'_\sigma, \sigma \in \Psi$, for which the minimum in $\mathcal{P}(d)(q, q')$ is reached under the conditions for the existence of probabilistic supervisor given by the constraints (9), (10), (11), (12).

4.2 Minimizing the Distance: Algorithm

We suggest an iterative algorithm to calculate minimum achievable distance (i.e. the only fixed point of the function $\mathcal{P}$) up to a desired accuracy and probability distribution of the achievable behaviour of the system when this distance is achieved. We use the notation from the Section 4.1. Also, we follow the proof pattern used in the Section 3.3. However, as mentioned before, only relevant distances are the ones between $q \in Q$ and $q' = f(q) \in Q'$. Let $d \in \mathcal{A}$. Further, let us define $\hat{\mathcal{V}}(d) = (\hat{\mathcal{V}}_1(d), \ldots, \hat{\mathcal{V}}_N(d))^T$ as:

$$\hat{\mathcal{V}}(d) = (d(q_0, q_0'), d(q_1, q_1'), \ldots, d(q_{N-1}, q_{N-1}'))^T.$$

Therefore, for $k = 1, \ldots, N$, $\hat{\mathcal{V}}_k(d) = d(q_{k-1}, q_{k-1}')$. Further, let $P_0 = \{\hat{\mathcal{V}}(d) | d \in \mathcal{A}\}$. We redefine the function $\mathcal{P}$

in a natural way as $\mathcal{P}(\hat{\mathcal{V}}(d)) = (\mathcal{P}_1(\hat{\mathcal{V}}(d)), \dots, \mathcal{P}_N(\hat{\mathcal{V}}(d)))^T$, where $\mathcal{P}_k(\hat{\mathcal{V}}(d)) = \mathcal{P}(d)(q_{k-1}, q'_{k-1})$ for any $k \in \{1, \dots, N\}$.

Theorem 3. For any $\hat{\mathfrak{d}}^0 \in P_0$, the sequence

$$\hat{\mathfrak{d}}^{n+1} = \mathcal{P}(\hat{\mathfrak{d}}^n), \;\; n = 0, 1, \dots$$

converges to the only fixed point of $\mathcal{P}$ in P_0, $\hat{\mathfrak{d}}^*$, and the error of convergence is given componentwise as:

$$|\hat{\mathfrak{d}}^n_k - \hat{\mathfrak{d}}^*_k| \leq (1-e)^{-1} e^n, \;\; n = 1, 2, \dots$$

The objective function in (8) is nonlinear, but transformable into a linear one by introducing variables y_σ. We now present the iterative algorithm for finding the fixed point of function $\mathcal{P}$. Let $d^0(q, q') = 0$ for all $q \in Q, q' = f(q) \in Q'$. The distance $d^n(q, q')$ between the states of $q \in Q$ and $q' = f(q) \in Q'$ in the n-th iteration ($n \geq 1$) is given as:

$$\begin{aligned}
&\text{Minimize} \sum_{\sigma \in \Psi} y_\sigma \qquad (13)\\
&\text{subject to}\\
&\rho_\sigma - \rho'_\sigma + c_i \rho'_\sigma \leq y_\sigma, \qquad \sigma \in \Psi\\
&c_i \rho_\sigma \leq y_\sigma, \qquad \sigma \in \Psi\\
&\text{where, for } \sigma \in \Psi, \; c_i = e \cdot d^{n-1}(q_i, q'_i) \text{ s.t. } q_i = \delta(q, \sigma)\\
&\frac{p_\sigma}{\sum\limits_{\alpha \in \Psi_u} p_\alpha} = \frac{\rho'_\sigma}{\sum\limits_{\alpha \in \Psi_u} \rho'_\alpha}, \qquad \sigma \in \Psi_u,\\
&\frac{\sum_{\alpha \in \Psi_u} p_\alpha}{p_\sigma} \rho'_\sigma + \sum_{\alpha \in \Psi_c} \rho'_\alpha \leq 1, \quad \sigma \in \Psi_c,\\
&0 \leq \rho'_\sigma \leq 1, \qquad \sigma \in \Psi,\\
&\sum_{\alpha \in \Psi} \rho'_\alpha = 1
\end{aligned}$$

After the n-th iteration, the values of decision variables ρ'_σ that represent the unknown transition probabilities, are such that that the distance between the (initial states of) systems G_2 and G'_2 is within $(1-e)^{-1}e^n$ of the minimal achievable distance between the two systems (in our pseudometric). Further, for each of the state of G_2 (typically, the number of states of G_2 is much smaller than $|Q_p| \cdot |Q_r|$), simplex method can be used to efficiently solve the linear programming problem (13). The worst-case time complexity of simplex method is exponential in the number of decision variables. In our case, the number of decision variables is twice the number of events possible from the state q. As this number is typically small in practical applications, this exponential complexity does not generally present a limitation of the algorithm. Furthermore, the number of iterations needed to reach the accuracy of ϵ is $\lceil \log_e(\epsilon(1-e)) \rceil$.

4.3 Example

For a plant G_p as depicted in Fig. 1, there does not exist a probabilistic supervisor V_p such that $G(V_p/G_p) = G_r$. First, PDESs G_1 and G_2 are found as suggested in Section 4.1. Then, using our iterative algorithm with 20 iterations, we find that the closest behaviour (for $e = 0.5$) achievable with probabilistic control is as given in Fig. 1.

5. CONCLUSIONS

Although the rate of convergence of the algorithm to the minimal distance is known, we would also like to investigate how unknown probabilities change as the distance converges. Also, the question of uniqueness of the closest approximation remains open as well as probabilistic control with marking.

REFERENCES

Barrett, G. and Lafortune, S. (1997). Using bisimulation to solve discrete event control problems. In *Proceedings of the 1997 American Control Conference*, 2337–2341. Albuquerque, NM.

Borkar, V.S. (1991). *Topics in controlled Markov chains*. Wiley, New York.

Chattopadhyay and Ray, A. (2007). Language-measure-theoretic optimal control of probabilistic finite-state systems. *International Journal of Control*, 80(8), 1271–1290.

Deng, Y., Chothia, T., Palamidessi, C., and Pang, J. (2006). Metrics for action-labelled quantitative transition systems. *Electronic Notes in Theoretical Computer Science*, 153(2), 79–96. Also appeared in *Proceedings of the 3rd Workshop on Quantitative Aspects of Programming Languages*.

Desharnais, J., Jagadeesan, R., Gupta, V., and Panangaden, P. (2002). The metric analogue of weak bisimulation for probabilistic processes. In *LICS '02: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science*, 413–422. IEEE Computer Society, Washington, DC, USA.

Garg, V. (1992a). An algebraic approach to modeling probabilistic discrete event systems. In *Proceedings of 31st IEEE Conference on Decision and Control*, 2348–2353. Tucson, AZ, USA.

Garg, V. (1992b). Probabilistic languages for modeling of DEDS. In *Proceedings of 26th Conference on Information Sciences and Systems*, volume 1, 198–203. Princeton, NJ.

Kumar, R. and Garg, V. (1998). Control of stochastic discrete event systems: Existence. In *Proceedings of 1998 International Workshop on Discrete Event Systems*, 24–29. Cagliari, Italy.

Lawford, M. and Wonham, W. (1993). Supervisory control of probabilistic discrete event systems. In *Proceedings of the 36th IEEE Midwest Symposium on Circuits and Systems*, volume 1, 327–331. IEEE.

Mortazavian, H. (1993). Controlled stochastic languages. In *Proceedings of 31st Annual Allerton Conference on Communications, Control, and Computing*, 938–947. Urbana, Illinois.

Pantelic, V. and Lawford, M. (2009). Optimal supervisory control of probabilistic discrete event systems. Technical Report 55, Software Quality Research Lab, McMaster University, Hamilton, ON, Canada.

Pantelic, V., Postma, S., and Lawford, M. (2009). Supervisory control of probabilistic discrete event systems. *IEEE Transactions on Automatic Control*. Accepted subject to revision.

Postma, S. and Lawford, M. (2004). Computation of probabilistic supervisory controllers for model matching. In *Proceedings of Allerton Conference on Communications, Control, and Computing*.

van Breugel, F. and Worrell, J. (2001). An algorithm for quantitative verification of probabilistic transition systems. In K.G. Larsen and M. Nielsen (eds.), *CONCUR*, volume 2154 of *Lecture Notes in Computer Science*, 336–350. Springer.