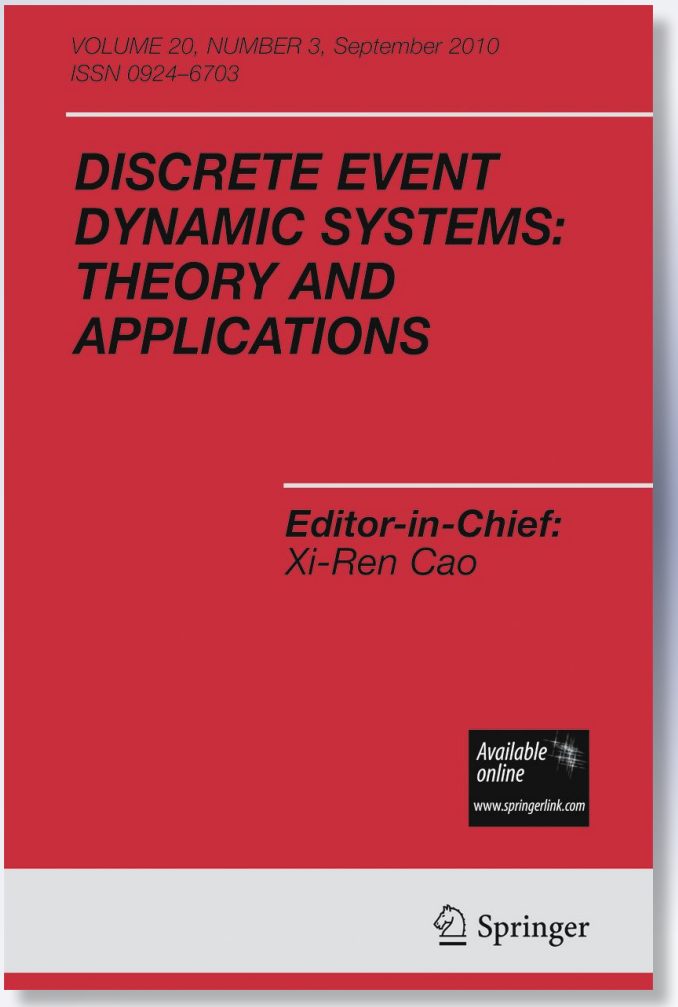
A pseudometric in supervisory control of probabilistic discrete event systems

Vera Pantelic & Mark Lawford

Discrete Event Dynamic Systems
Theory and Applications

ISSN 0924-6703
Volume 22
Number 4

Discrete Event Dyn Syst (2012)
22:479-510
DOI 10.1007/s10626-011-0126-7



Springer

Discrete Event Dyn Syst (2012) 22:479–510
DOI 10.1007/s10626-011-0126-7

A pseudometric in supervisory control of probabilistic discrete event systems

Vera Pantelic · Mark Lawford

Received: 11 March 2011 / Accepted: 6 December 2011 / Published online: 8 January 2012

Abstract The focus of this paper is the pseudometric used as a key concept in our previous work on optimal supervisory control of probabilistic discrete event systems. The pseudometric is employed to measure the behavioural similarity between probabilistic systems, and initially was defined as a greatest fixed point of a monotone function. This paper further characterizes the pseudometric. First, it gives a logical characterization of the pseudometric so that the distance between two systems is measured by a formula that distinguishes between the systems the most. A trace characterization of the pseudometric is then derived from the logical characterization, characterizing the similarity between systems from a language perspective. Further, the solution of the problem of approximation of a given probabilistic generator with another generator of a prespecified structure is suggested such that the new model is as close as possible to the original one in the pseudometric. The significance of the approximation is then discussed, especially with respect to previous work on optimal supervisory control of probabilistic discrete event systems.

Keywords Supervisory control · Probabilistic systems · Pseudometric · Optimal control

1 Introduction

A supervisory control framework for *probabilistic discrete event systems (PDES)* was proposed in Lawford and Wonham (1993). The framework represents a straightforward probabilistic extension of the framework of standard Supervisory Control

V. Pantelic (✉) · M. Lawford
Department of Computing and Software, Faculty of Engineering, McMaster University, 1280 Main Street West, Hamilton, ON, Canada L8S 4K1
e-mail: pantelv@mcmaster.ca

M. Lawford
e-mail: lawford@mcmaster.ca

Theory (SCT). PDES are modeled as *probabilistic generators* inspired by Garg (1992a, b). Probabilistic generators are a generalization of generators used in standard SCT to model discrete event systems (DES): each transition is labeled not only with an event but also with a probability that represents the probability of the occurrence of the transition. The probabilities of all the events in a state add up to at most one. Further, deterministic supervisors for DES are generalized to *probabilistic supervisors*: after observing a string s, the probabilistic supervisor enables an event σ with a certain probability. Although far more intricate than deterministic control, both from a theoretical and a practical point of view, probabilistic control is also much more powerful: Lawford and Wonham (1993) show that a plant under probabilistic control can generate a much larger class of probabilistic languages than deterministic control. Further, the classical Supervisory Control Problem is generalized to the *Probabilistic Supervisory Control Problem (PSCP)* (Lawford and Wonham 1993). The PSCP is to find, if possible, a supervisor under whose control the behaviour of a plant is identical to a given probabilistic specification. Necessary and sufficient conditions for the existence of a supervisor for the PSCP are given in Lawford and Wonham (1993) and Pantelic et al. (2009). 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), and Pantelic et al. (2009). Further, 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 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 behaviour closest to the desired behaviour (Pantelic and Lawford 2009, 2012). The problem is referred to as the *Optimal Probabilistic Supervisory Control Problem (OPSCP)*. The nonprobabilistic behaviour of the requirements specification is considered to be a safety constraint in the standard supervisory control sense similar to Kumar and Garg (1998). Therefore, the supremal controllable sublanguage of the specification with respect to the plant is generated as the maximal achievable legal nonprobabilistic behaviour of the plant under control. Then, the transition probabilities of the controlled plant are determined such that it is behaviourally the most similar to the requirements specification (whose nonprobabilistic behaviour is reduced to the mentioned supremal controllable sublanguage). The behavioural similarity is measured using a *pseudometric* on states of probabilistic generators. Therefore, a controlled plant at the minimal distance (in the chosen pseudometric) from the modified requirements specification is found: this controlled plant is referred to as a closest approximation.

The pseudometric is based on the pseudometric introduced in Deng et al. (2006). It measures behavioural similarity between two states: the smaller the distance, the greater similarity between the states. The pseudometric subsumes probabilistic bisimulation: two states are at distance 0 in the pseudometric if and only if they are probabilistic bisimilar. The pseudometric has a discount factor $e \in (0, 1]$: the smaller the factor, the greater the discount on differences between systems farther in the future than those in the near future. This pseudometric is inspired by the Kantorovich metric (Kantorovich 1942) which is used in transport problems, and more recently has been used by Hutchinson in his theory of fractals (Hutchinson 1981). The metric is also known as Wasserstein metric (Wasserstein 1969), earthmover's metric,

transport metric etc. While extensively used in business, economics, scheduling problems etc., the metric has only recently found applications in computer science (Deng and Du 2009). The work of Deng et al. (2006) is closely related to Desharnais et al. (1999, 2002, 2004), van Breugel and Worrell (2001, 2005, 2006), van Breugel et al. (2005, 2007), Ferns et al. (2004, 2005, 2006) which consider reactive systems.

For $e \in (0, 1)$, there is a simple algorithm to compute distances in our pseudometric for our generative, deterministic model (see Pantelic and Lawford 2009, 2012; with detailed proofs given in Pantelic 2011). Also, the pseudometric intuitively matches our notion of the distance between PDES and accounts for all differences between corresponding transition probabilities, as opposed to e.g., that of Giacalone et al. (1990) that, roughly speaking, considers only the maximum of the differences between the corresponding probabilities. Furthermore, as the pseudometric is applicable to a large class of systems, it allows for an extension of our work to e.g., nondeterministic systems.

While our initial interest in the pseudometric lies in the context of supervisory control theory of probabilistic discrete event systems, the pseudometric is interesting in its own right. We are not interested in topological aspects of the pseudometric (convergence, continuity, etc.), but rather in its use as a tool to measure the behavioural similarity of systems in our framework. As Giacalone et al. (1990), Desharnais et al. (1999, 2002) (to name a few) pointed out, probabilistic bisimulation is not robust as it requires the exact matching of the values of probabilities of corresponding transitions. It is too sensitive to small changes in probabilities: a slight change of probabilities makes bisimilar systems nonbisimilar. Similarly, two systems with only slightly different probabilities of corresponding transitions would be as different as two systems with disjoint event sets (Deng et al. 2006). Further, as the values of probabilities are often only approximations, using either probabilistic bisimulation or reasoning in a boolean-valued logic is not sensible (van Breugel and Worrell 2005). The notion of a pseudometric is hence used to approximate the notion of equivalence. It provides for a notion of "approximately intersubstitutable" system instead of rigid "exactly intersubstitutable" system (Desharnais et al. 1999), although we do not explore compositional reasoning in this paper.

This paper further characterizes the pseudometric as to deepen the understanding of it as an approximation tool in our framework. The characterization also provides an additional, a posteriori motivation for the choice of the pseudometric in the solution of the control problem of Pantelic and Lawford (2009, 2012). First, the pseudometric is characterized using a real-valued logic. In the aforementioned bulk of research closely related to the pseudometric of Deng et al. (2006), Desharnais et al. (1999) were the first to suggest a pseudometric via a real-valued logic that is motivated by the well-known result that the Hennessy-Milner logic is complete for bisimulation Arnold (1994). More concretely, Desharnais et al. (1999) use the ideas of Kozen (1985) to generalize a logic so that reasoning about probabilistic systems is supported. Let $\mathcal{F}$ be a set of functions such that a function $f \in \mathcal{F}$ evaluated at a state takes a truth value in the interval [0, 1], instead of {0, 1}. Then, the distance between two states is defined as a pseudometric:

$$d(q_q, q_r) = sup\{|f(q_q) - f(q_r)| f \in \mathcal{F}\}. \quad (1)$$

Similarly, Desharnais et al. (2002), van Breugel and Worrell (2005) and van Breugel et al. (2007) suggest pseudometrics via real-valued logics for different reactive

models. Our *logical characterization* is the most similar to that of Desharnais et al. (2002). However, the logic itself is different than that of Desharnais et al. (2002) as our models are generative. Also, the main part of the characterization proof is, to the best of our knowledge, novel. The idea of the logical characterization is that the distance between two systems is measured by a real-valued formula that distinguishes between the systems the most (as in Eq. 1).

Further, in this paper, this logical characterization is used to derive a *trace characterization*. While trace characterization has not been of concern in any of the aforementioned related literature, it is of importance in our work, given the language-oriented aspect of supervisory control theory. The trace characterization answers the following question: "If two systems are similar in our pseudometric, how similar are the discounted probabilities of the strings generated by the systems?" The probability of (the occurrence of) a string is discounted by discount factor e for each event in the string.

In the control theory of PDES, Chattopadhyay and Ray (2008) introduce a pseudometric in a symbolic pattern recognition application to measure the distance between the original model and one with a prespecified structure, where the latter has the same long term distribution over the states as the original one. In this paper, the problem of a similar probabilistic model transformation is discussed in our setting. The problem is referred to as the *Probabilistic Model Fitting Problem*. A probabilistic generator is approximated by another one with a prespecified structure such that the distance between the two is minimal in our pseudometric. The significance of model fitting for model reduction and control-related applications is then discussed. Then, a transformation used in the solution of the probabilistic model fitting problem is used to solve a modified version of the OPSCP: instead of minimizing the distance between the controlled plant and the requirements specification restricted to the supremal controllable sublanguage, the distance between the controlled plant and the original requirements specification is minimized.

Our research should find an application in the field of robotics as probabilistic generators have been used extensively to model systems in the control of robot systems (Li et al. 1998; Mallapragada et al. 2009; Chattopadhyay et al. 2009). Also, the Kantorovich metric has been used in a number of applications (Deng and Du 2009). The most promising area in regard to our research is the field of bionformatics, where the metric has been increasingly used (Thorsley and Klavins 2010; Koeppl et al. 2010; Deng and Du 2009). Further, one of the routes to explore is the use of our research in the generation of test cases (adversaries) for MDPs. More precisely, a probabilistic generator can be viewed as a supervisor for MDPs (see Pantelic 2011). On the other hand, a probabilistic supervisor as defined in our framework can be represented as an MDP (see Pantelic 2011). This duality between the plant to be controlled and the probabilistic supervisor performing the control might provide interesting connections between probabilistic model checking and supervisory control theory.

In Section 2, the probabilistic control of PDES is reviewed. Section 3 presents the logical characterization of the pseudometric. The trace characterization stems from the logical one and is presented in Section 4. The probabilistic model fitting problem, its solution, and its applications are introduced in Section 5. Section 6 solves the modified OPSCP. Section 7 concludes with avenues for future work.

This paper is an extended version of the conference paper (Pantelic and Lawford 2010). The conference version has been extended with detailed proofs and the solution of the modified OPSCP.

2 Preliminaries

In this section, PDES modeled as generators of probabilistic languages are presented, and probabilistic control is introduced. The pseudometric is next defined. Finally, the problem statements and solutions for the PSCP and the OPSCP are presented.

2.1 Modeling PDES

Following Lawford and Wonham (1993) and Pantelic et al. (2009), a probabilistic DES is modeled as a *probabilistic generator* defined as follows.

Definition 1 A *probabilistic generator* G is a tuple $G = (Q, \Sigma, \delta, q_0, p)$, where Q is the nonempty finite set of states, 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, and $p : Q \times \Sigma \to [0, 1]$ is the statewise event probability distribution, i.e. for any $q \in Q$, $\sum_{\sigma \in \Sigma} p(q, \sigma) = 1$. The probability that the generator will execute event $\sigma \in \Sigma$ at state $q \in Q$ is $p(q, \sigma)$. For generator G to be well-defined, $p(q, \sigma) = 0$ holds if and only if $\delta(q, \sigma)$ is undefined.

Remark 1 Relaxing the condition $\sum_{\sigma \in \Sigma} p(q, \sigma) = 1$ to $\sum_{\sigma \in \Sigma} p(q, \sigma) \leq 1$ would allow for modeling termination. The probability that the system terminates at state q would then be $1 - \sum_{\sigma \in \Sigma} p(q, \sigma)$. However, since a terminating PDES can easily be transformed into a probabilistic generator of Definition 1 using the technique described in Lawford and Wonham (1993), we find the model of Definition 1 general enough for our purposes.

The state 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 string s in state q. Note that the definition of PDES does not contain marking states since the probabilistic specification languages considered in this paper are prefix closed languages. The language $L(G)$ generated by G 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 $\Sigma \times Q$ induced by q. Also, for a state q, we define the *set of possible events* to be $Pos(q) := \{\sigma \in \Sigma | p(q, \sigma) > 0\}$, or, equivalently, $Pos(q) := \{\sigma \in \Sigma | \delta(q, \sigma)!\}$.

Next, the *synchronous product* of the (nonprobabilistic) discrete event systems that underlie PDES is defined in a standard manner. For a probabilistic generator $G = (Q, \Sigma, \delta, q_0, p)$, the (nonprobabilistic) DES that underlies G will be denoted G^{np}, i.e., $G^{np} = (Q, \Sigma, \delta, q_0)$ throughout this paper. Let G_1^{np} and G_2^{np} be the nonprobabilistic generators (DES) underlying $G_1 = (Q_1, \Sigma, \delta_1, q_{0_1}, p_1)$ and $G_2 = (Q_2, \Sigma, \delta_2, q_{0_2}, p_2)$, respectively, i.e., $G_1^{np} = (Q_1, \Sigma_1, \delta_1, q_{0_1})$ and $G_2^{np} = (Q_2, \Sigma, \delta_2, q_{0_2})$.

Definition 2 The *synchronous product* of $G_1^{np} = (Q_1, \Sigma, \delta_1, q_{0_1})$ and $G_2^{np} = (Q_2, \Sigma, \delta_2, q_{0_2})$, denoted $G_1^{np} \parallel G_2^{np}$, is the reachable sub-DES of DES $G_a = (Q_a, \Sigma, \delta, q_0)$, where $Q_a = Q_1 \times Q_2$, $q_0 = (q_{0_1}, q_{0_2})$, and, for any $\sigma \in \Sigma$, $q_i \in Q_i$, $i = 1, 2$, it holds that $\delta((q_1, q_2), \sigma) = (\delta_1(q_1, \sigma), \delta_2(q_2, \sigma))$ whenever $\delta_1(q_1, \sigma)!$ and $\delta_2(q_2, \sigma)!$.

While the synchronous product of nonprobabilistic DES as defined by Definition 2 is straightforward in supervisory control theory, the definition of the synchronous product of probabilistic discrete event systems requires more careful consideration. However, it is not needed for the results of this paper.

2.2 Probabilistic control

As in classical supervisory control 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. 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.

Let $x : L(G) \to [0, 1]^{\Sigma_c}$. For a PDES $G = (Q, \Sigma, \delta, q_0, p)$, a *probabilistic supervisor* is a function $V_p : L(G) \to [0, 1]^{\Sigma}$ such that

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

Therefore, after observing a string $s \in L(G)$ (all the events are assumed to be observable), the supervisor enables event σ with probability $V_p(s)(\sigma)$. More precisely, for event σ, the supervisor performs a Bernoulli trial with possible outcomes *enable* (that has the probability $V_p(s)(\sigma)$), and *disable* (with probability $1 - V_p(s)(\sigma)$), and, depending on the outcome of the trial, decides whether to enable or disable the event. After (independent) Bernoulli trials have been performed for all controllable events, control pattern Θ is determined as a set of controllable events such that a controllable event belongs to Θ if and only if its corresponding Bernoulli trial resulted in outcome *enable*. After Θ has been decided upon, the system acts as if supervised by a deterministic supervisor. Given sets A, B, we will denote the 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 plant G under the control of the supervisor V_p will be denoted V_p/G. The probability that the event $\alpha \in \Sigma$ will occur in the controlled plant V_p/G after string s has been observed is equal to:

$$P(\alpha \text{ in } V_p/G|s) = \sum_{\Theta \in \mathcal{P}(Pos(q) \cap \Sigma_c)} P(\alpha|V_p \text{ enables } \Theta \text{ after } s) \cdot P(V_p \text{ enables } \Theta|s) \quad (2)$$

where

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

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

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.

2.3 PSCP

The formulation of the PSCP is given first, and, then, its solution is presented.

2.3.1 PSCP: formulation

The problem was first presented in Lawford and Wonham (1993). The goal is to match the behaviour of the controlled plant with a given probabilistic specification language. The problem is called the *Probabilistic Supervisory Control Problem (PSCP)*. More formally:

Given a plant PDES G_p and a specification PDES G_r, find, if possible, a probabilistic supervisor V_p such that $L_p(V_p/G_p) = L_p(G_r)$.

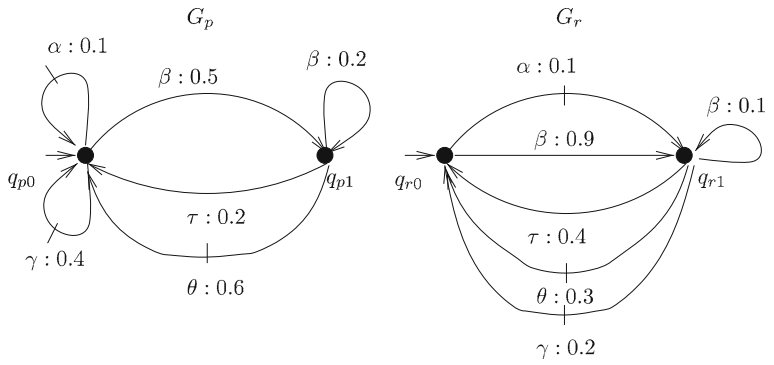


Fig. 1 Plant G_p, and requirements specification G_r

2.3.2 PSCP: solution

We present the conditions for the existence of a probabilistic supervisor for the PSCP from Lawford and Wonham (1993) and Pantelic et al. (2009).

Theorem 1 *Let $G_p = (Q_p, \Sigma, \delta_p, q_{p_0}, p_p)$ and $G_r = (Q_r, \Sigma, \delta_r, q_{r_0}, p_r)$ be two PDES with disjoint state sets Q_p and Q_r. Then, let G_p^{np} and G_r^{np} be the nonprobabilistic generators underlying G_p and G_r, respectively, i.e. $G_p^{np} = (Q_p, \Sigma, \delta_p, q_{p_0})$ and $G_r^{np} = (Q_r, \Sigma, \delta_r, q_{r_0})$. Also, let $G_s = (Q_s, \Sigma, \delta_s, q_{0_s})$ be the synchronous product of generators G_p^{np} and G_r^{np}, $G_s = G_p^{np} \parallel G_r^{np}$. There exists a probabilistic supervisor V_p such that $L_p(V_p/G_p) = L_p(G_r)$ iff for all $(q, r) \in Q_s$, the following two conditions hold:*

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

$$\frac{p_p(q,\sigma)}{\sum_{\alpha\in\Sigma_u} p_p(q,\alpha)} = \frac{p_r(r,\sigma)}{\sum_{\alpha\in\Sigma_u} p_r(r,\alpha)}$$

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

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

Conditions (i) and (ii) together are necessary and sufficient for the existence of a probabilistic supervisor. 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. For each uncontrollable event possible from a state in a plant, the equation to be checked reflects the fact that the ratio of probabilities of uncontrollable events remains the same under supervision. This comes from the fact that after a control pattern has been chosen, the probabilities of disabled events in the plant are redistributed over enabled events in proportion to their probabilities. Any possible uncontrollable events are always enabled, hence the ratios of their probabilities remain unchanged. An inequality for each possible controllable event σ is derived from the the upper bound on the probability of the occurrence of σ in the supervised plant, that is reached when the controllable event is always enabled.

When the conditions are satisfied, a solution to the PSCP exists. The probabilistic supervisor can then be computed by the fixed point iteration algorithm as presented in Postma and Lawford (2004) and Pantelic et al. (2009). For the example from Fig. 1, the probabilistic supervisor for the PSCP is given in Fig. 2.

Also, we would like to note that the initial plant, the supervisor, and the controlled plant satisfy the Markov property as can be easily inferred.

2.4 Definition of the pseudometric

Probabilistic bisimulation, introduced in Larsen and Skou (1991), is commonly used to define an equivalence relation between probabilistic systems. However, probabilistic bisimulation is not a robust relation: the probabilities of corresponding

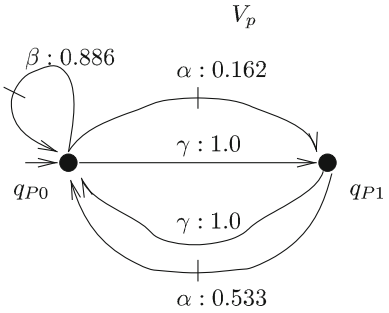


Fig. 2 Probabilistic supervisor V_p such that $L(V_p/G_p) = L(G_r)$ for G_p and G_r from Fig. 1

transitions must match exactly. 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 $x, y, z \in Q$. A pseudometric generalizes a metric in that two distinct points are allowed to be at the distance 0. If all distances are less than or equal to 1, the pseudometric is *1-bounded*.

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. Two states are at distance 0 in this pseudometric if and only if they are probabilistic bisimilar. Here, the pseudometric is presented only for probabilistic generators.

Let $G = (Q, \Sigma, \delta, q_0, p)$ be a PDES, where $Q = \{q_0, q_1, \ldots q_{N-1}\}$.

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 $(d_1, d_2 \in \mathcal{M})$

$$d_1 \preceq d_2 \text{ if } \forall q_q, q_r \in Q \; d_1(q_q, q_r) \geq d_2(q_q, q_r). \tag{3}$$

Further, it is proved that $(\mathcal{M}, \preceq)$ is a complete lattice. The ordering in Eq. 3 is reversed for the purpose of characterizing bisimilarity as the greatest fixed point of a function.

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 future transitions is discounted: the smaller the value of e, the greater the discount on future transitions. 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. 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 $j(q_r, \sigma)$. Further, we will write $\rho_{\sigma,i}$ instead of $\rho_{q_q}(\sigma, q_i)$, and, similarly, $\rho'_{\sigma,j}$ instead of $\rho_{q_r}(\sigma, q_j)$. Then, the pseudometric on states d_{fp} is given as the greatest fixed point of the function $\mathcal{D}$ on $\mathcal{M}$, that, in the special case of probabilistic generators, can be shown to be (see Pantelic and Lawford 2009, 2012):

$$\begin{aligned}\mathcal{D}(d)(q_q, q_r) &= \sum_{\sigma \in \Sigma} \max(\rho_{\sigma,i} - \rho'_{\sigma,j} + e\rho'_{\sigma,j} d(q_i, q_j), e\rho_{\sigma,i} d(q_i, q_j)) \\ &= \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} \geq \rho'_{\sigma,j}\}} \left(\rho_{\sigma,i} - \rho'_{\sigma,j} + e\rho'_{\sigma,j} d(q_i, q_j)\right) + \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} < \rho'_{\sigma,j}\}} e\rho_{\sigma,i} d(q_i, q_j)\end{aligned} \tag{4}$$

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 distances between the states in d_{fp} are larger by the factor $1/e$ than the distances in pseudometric defined in Deng et al. (2006). This has been done so that the distances are in the range $[0, 1]$, instead of $[0, e]$.

Remark 2 According to Tarski's fixed point theorem, since $\mathcal{D}$ is a monotone function on a complete lattice, it has a greatest fixed point. Furthermore, this greatest fixed point can be reached through an iterative process that starts from the greatest element. As the number of transitions from a state of a probabilistic generator is finite, the greatest fixed point of the function $\mathcal{D}$ is reached after at most ω iterations (Deng et al. 2006; Desharnais et al. 2002) (equivalently, the closure ordinal of $\mathcal{D}$ is ω, where ω is the first infinite ordinal). Therefore, the pseudometric d_{fp} can be reached through the following iterative process.

Definition 3 The distance function d^0_{fp} is defined as:

$$d^0_{fp} = 0,$$

and the distance function d^{n+1}_{fp}, $n \in \mathbb{N}$, is given as:

$$d^{n+1}_{fp} = \mathcal{D}(d^n_{fp}), \tag{5}$$

where $\mathcal{D}$ is given in Eq. 4.

Remark 3 An important feature of d_{fp} is to be noted: pseudometric d_{fp} is defined on any two states of a single PDES, not on two states that belong to different PDES. In order to define the distance between two PDES (with disjoint sets of states) as the distance between their initial states, a new PDES is created that represents the union of the two PDES, with the initial state arbitrarily chosen between the initial states of the two PDES. The union will not be formalized as it does not change the distance between the states.

Remark 4 Discount factor e is very important in the definition of pseudometric d_{fp}. An excellent discussion on the significance of discount in different fields has been presented in de Alfaro et al. (2003). E.g., in economics, it has been a key concept as it models inflation. From an engineering point of view presented in de Alfaro et al. (2003), the near future might feel more important than the distant future (e.g., a bug in the near future is more urgent than the one in the distant future; entering a bad state, if not preventable, might be desired to be postponed, etc.). Also, the concept of discount has been widely applied in game theory, and optimal control (Blackwell 1962). Discount is also important from the computational point of view. Specifically, when it comes to pseudometrics, van Breugel and Worrell (2001, 2006), Ferns et al. (2004, 2005, 2006) all present polynomial algorithms that approximate (with a specified accuracy) the distances in respective pseudometrics (also based on the Kantorovich metric with a discount factor). The algorithms are applicable

for $e \in (0, 1)$. In general, for Kantorovich-like pseudometrics, efficient algorithms typically exist that approximate the distances for $e \in (0, 1)$, while no such algorithms are known to exist for $e = 1$. It is not surprising, then, that for $e \in (0, 1)$, there is a simple algorithm to compute distances in pseudometric d_{fp} for our generative, deterministic model (see Pantelic and Lawford 2009 and 2012).

2.5 OPSCP

In the case when the conditions for the existence of a solution to the probabilistic supervisory control problem are not satisfied, we search for a suitable approximation. The problem is referred to as the *Optimal Probabilistic Supervisory Control Problem* (OPSCP). The problem and the solution were introduced in Pantelic and Lawford (2009, 2012), with detailed proofs presented in Pantelic (2011).

2.5.1 OPSCP: formulation

Optimal Probabilistic Supervisory Control Problem (OPSCP): Let $G_p = (Q_p, \Sigma, \delta_p, q_{p_0}, p_p)$ be a plant PDES, and let $G_r = (Q_r, \Sigma, \delta_r, q_{r_0}, p_r)$ be a requirements specification represented as a PDES. If there is no probabilistic supervisor V_p such that $L_p(V_p/G_p) = L_p(G_r)$ (i.e., the conditions of Theorem 1 fail), find, if it exists, V_p such that

1. $L(V_p/G_p) \subseteq L(G_r)$ and supervisor V_p is maximally permissive in the nonprobabilistic sense (i.e., $L(V_p/G_p)$ is the supremal controllable sublanguage of $L(G_r)$ with the respect to G_p).
2. The probabilistic behaviour of the controlled plant is "as close as possible" to the probabilistic behaviour of the requirements specification restricted to the supremal controllable sublanguage of $L(G_r)$ with the respect to G_p.

Let $G = V_p/G_p = (S, \Sigma, \delta, s_0, p)$ be a closest approximation.

The first criterion is straightforward. The requirement G_r represents a safety constraint: the controlled plant is not allowed to generate strings not in $L(G_r)$ even with the smallest of probabilities. Further, the criterion of maximal permissiveness is a standard one for optimality of supervisory control. The second criterion, on the other hand, is probabilistic: the distance in pseudometric d_{fp} between the initial states of the probabilistic generators G and an appropriately modified G_r is chosen as a measure of probabilistic similarity. The requirements specification G_r is modified such that its (nonprobabilistic) language is the supremal controllable sublanguage of $L(G_r)$ with respect to G_p. Consequently, the probabilities of the specification are revised so that the probabilities of the events inadmissible for not satisfying the first criterion are redistributed over the admissible ones. The reasons for using the modified specification are the following:

– After realizing that only a subset of the desired nonprobabilistic behaviour is achievable, the designer may see no reason in insisting on probabilities suggested for the behaviour that cannot be achieved. We assume that the designer wants to, for each state, distribute the probabilities of the events not possible anymore over the remaining events so that the new probabilities are proportional to the old ones. However, the designer might want to rebalance the probabilities any way it suites him/her.

– It might be the case that the designer prefers to leave the specification intact. Then, the problem to solve would be a modified OPSCP with criterion (2) changed so that the difference between the controlled plant and the original specification is minimized. As will be shown in Section 6, with some preprocessing, the algorithm that solves the original OPSCP (which is reproduced in Section 2.5.2) can be reused to solve the modified OPSCP.

2.5.2 Solution of the OPSCP

The solution of the OPSCP uses the separation of concerns from its formulation: the classical supervisory theory of supremal controllable sublanguages is used first, and, then, a closest approximation in the probabilistic sense is calculated.

First, the classical controllability conditions that correspond to the first parts of conditions (i) and (ii) of Theorem 1 are checked while constructing $L(G_p) \cap L(G_r)$. Then, if these conditions are not satisfied, the goal is to find K, the deadlock-free supremal controllable sublanguage of $L(G_p) \cap L(G_r)$ (with respect to G_p). The language K is required to be deadlock-free since termination is not allowed (as assumed in Section 2.1). Then, the DES that represents this language K, further equipped with distribution p_p (appropriately normalized) becomes the modified plant PDES G_1. Also, a DES corresponding to language K equipped with the distribution p_r appropriately normalized, becomes the desired behaviour PDES G_2. Formally, let the reachable and deadlock-free DES $G_{1k} = (T, \Sigma, \zeta, t_0)$ represent language K. Generator PDES $G_1 = (T, \Sigma, \zeta, t_0, p_1)$ is defined, where the distribution $p_1 : T \times \Sigma \to [0, 1]$, for any $q \in T, \sigma \in \Sigma$, is defined as:

$$p_1(q, \sigma) = \frac{p_p(q_p, \sigma)}{\sum_{\sigma \in \{\sigma \in \Sigma \mid \zeta(q,\sigma)!\}} p_p(q_p, \sigma)}$$

where $q_p = \delta_p(q_{p_0}, s)$ for any $s \in K$ such that $q = \zeta(t_0, s)$.

Similarly, let $G_{2k} = (Q, \Sigma, \delta, q_0)$ be a DES isomorphic to G_{1k} up to renaming of states, and, without loss of generality, assume $T \cap Q = \emptyset$. Obviously, the nonprobabilistic language generated by G_{2k} is K, too. Similarly, we define a PDES $G_2 = (Q, \Sigma, \delta, q_0, p)$ where the distribution $p : Q \times \Sigma \to [0, 1]$, for any $q \in Q, \sigma \in \Sigma$, is defined as:

$$p(q, \sigma) = \frac{p_r(q_r, \sigma)}{\sum_{\sigma \in \{\sigma \in \Sigma \mid \delta(q,\sigma)!\}} p_r(q_r, \sigma)}$$

where $q_r = \delta_r(q_{r_0}, s)$ for any $s \in K$ such that $q = \delta(q_0, s)$. Note that p_1 and p are well-defined as no state minimization on the automaton representing language K is performed. E.g., if the minimization were performed, it could happen that, for $s_1, s_2 \in K, q_1, q_2 \in Q_p$ such that $\zeta(t_0, s_1) = \zeta(t_0, s_2)$, $q_1 = \delta_p(q_{p_0}, s_1)$, and $q_2 = \delta_p(q_{p_0}, s_2)$, it holds that $q_1 \neq q_2$. Then, in general, $p_p(q_1, \sigma) \neq p_p(q_2, \sigma)$ for $\sigma \in \Sigma$, and p_1 would not be well-defined.

Now, the probability matching equations and inequalities from Theorem 1 are checked. If they are not satisfied (i.e., there is no probabilistic supervisor V_p such

that $L_p(V_p/G_1) = L_p(G_2)$), the goal is to find $G_2' = (Q', \Sigma, \delta', q_0', p')$ such that there exists a probabilistic supervisor V_p so that $L_p(V_p/G_1) = L_p(G_2')$ holds, and G_2' is closest to G_2 in pseudometric d_{fp}. Without loss of generality, it is assumed that $Q \cap Q' = \emptyset$. Also, without loss of generality, it is assumed that the nonprobabilistic automata underlying G_2 and G_2' are isomorphic (with labeling of events being preserved). Therefore, the nonprobabilistic automata underlying G_2 and G_2' (G_2^{np} and $G_2'^{np}$, respectively) 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 K, and, therefore, since $K = L(G_2)$, there cannot be any string in the desired system that does not belong to $L(G_2)$. This comes from the fact that $L(G_2)$ is the reachable and deadlock-free supremal controllable sublanguage: if any string not in $L(G_2)$ would be allowed in the controlled plant, either the safety or nontermination requirement would not be met.

Let f be the isomorphism between G_2^{np} and $G_2'^{np}$. Also, let h be the isomorphism between G_2^{np} and G_1^{np}. G_2' is approximated as follows.

Theorem 2 *Let* $T = \{t_0, t_1, \ldots t_{N-1}\}$, $Q = \{q_0, q_1, \ldots q_{N-1}\}$, *and* $Q' = \{q_0', q_1', \ldots q_{N-1}'\}$, *where* $q_i' = f(q_i)$, $t_i = h(q_i)$, $i = 0, \ldots, N-1$. *Let* $0 \le i \le N-1$, $\Psi(q_i) = Pos(q_i)$, $\Psi_u(q_i) = Pos(q_i) \cap \Sigma_u$, *and* $\Psi_c(q_i) = Pos(q_i) \cap \Sigma_c$. *Let* ρ_q *be the probability distribution induced by the state* $q \in Q$ *of PDES* G_2 *and let* $\rho'_{q'}$ *be the probability distribution induced by the state* $q' \in Q'$ *of PDES* G_2'. *Also, we will write* j *for* $j(i, \sigma)$, *then* $\rho_{q_i,\sigma}$ *instead of* $\rho_{q_i}(\sigma, q_k)$, *and* $\rho'_{q_i',\sigma}$ *instead of* $\rho'_{q_i'}(\sigma, q_k')$, $k = 0, 1, \ldots N-1$. *Let* $d^0(q_i, q_i') = 0$, $i = 0, 1, \ldots N-1$. *The distance* $d^n(q_i, q_i')$ *in the n-th iteration* ($n > 0$) *is given as:*

$$\text{Minimize} \sum_{\sigma \in \Psi(q_i)} y_{q_i,\sigma} \tag{6}$$

subject to

$$\rho_{q_i,\sigma} - \rho'_{q_i',\sigma} + c_j \rho'_{q_i',\sigma} \le y_{q_i,\sigma}, \qquad \sigma \in \Psi(q_i)$$

$$c_j \rho_{q_i,\sigma} \le y_{q_i,\sigma}, \qquad \sigma \in \Psi(q_i)$$

$$\text{where } c_j = e \cdot d^{n-1}(q_j, q_j') \text{ s.t. } q_j = \delta(q_i, \sigma),$$

$$p_1(t_i, \sigma) \sum_{\alpha \in \Psi_u(q_i)} \rho'_{q_i,\alpha} = \rho'_{q_i',\sigma} \sum_{\alpha \in \Psi_u(q_i)} p_1(t_i, \alpha), \quad \sigma \in \Psi_u(q_i),$$

$$\frac{\sum_{\alpha \in \Psi_u} p_1(t_i, \alpha)}{p_1(t_i, \sigma)} \rho'_{q_i',\sigma} + \sum_{\alpha \in \Psi_c(q_i)} \rho'_{q_i',\alpha} \le 1, \quad \sigma \in \Psi_c(q_i),$$

$$\sum_{\alpha \in \Psi(q_i)} \rho'_{q_i',\alpha} = 1,$$

$$\rho'_{q_i',\sigma} \ge 0, \qquad \sigma \in \Psi(q_i).$$

After the n-th iteration, the values of decision variables $\rho'_{q_i',\sigma}$ *that represent the unknown transition probabilities, are such that the distance between the (initial states*

of) systems G_2 and G_2' is within e^n of the minimal achievable distance between the two systems (in pseudometric d_{fp}).

Therefore, the algorithm first finds K, the supremal controllable sublanguage of $L(G_r)$ with respect to G_p. Then, probabilistic generator G_2 that represents a modified requirements specification is constructed such that its underlying graph generates exactly the sublanguage K, while the probabilities are appropriately normalized (or, in general, modified as a user wishes). Likewise, probabilistic generator G_2' that represents a closest approximation is constructed such that its underlying graph generates exactly sublanguage K, while the probabilities of G_2' are yet to be determined. Then, the distance in d_{fp} between these two generators representing the controlled plant, and the modified probabilistic requirement is now minimized such that the probabilistic controllability conditions of Theorem 1 are satisfied. An iterative algorithm is given to approximate the probabilities of the controlled plant G_2'. More precisely, as the underlying graphs of the two generators are isomorphic, in each iteration, the distance is minimized by minimizing the distance between each pair of isomorphic states. The algorithm iterates until a prespecified accuracy is reached.

Note that the aforementioned results hold for $e \in (0, 1)$. The detailed proof of the Theorem 2 can be found in Pantelic (2011).

3 Logical characterization

The pseudometric with the fixed point characterization as presented in Section 2.4 is now given a logical characterization, along the lines of Desharnais et al. (2002). The idea behind the logical characterization is that the distance between two systems is measured by a logical formula that distinguishes between the systems the most. If the systems are probabilistic bisimilar, there should not be a formula that distinguishes between the systems.

As before, let $G = (Q, \Sigma, \delta, q_0, p)$ be a probabilistic generator, where $Q = \{q_0, q_1, \ldots q_{N-1}\}$, and discount factor $e \in (0, 1]$.

Definition 4 The logic $\mathcal{L}$ is defined as follows:

$$\phi ::= \mathbf{1} \mid \langle\sigma\rangle\phi \mid \bigvee_{\sigma\in\Theta} \langle\sigma\rangle\phi \mid 1 - \phi \mid \phi \ominus p,$$

where p is a rational number in $[0, 1]$, $\sigma \in \Sigma$, and $\Theta \subseteq \Sigma$.

The formula ϕ evaluated at a state $q \in Q$, denoted $\phi(q)$, is a measure of how much ϕ is satisfied in state q. The semantics of the logic $\mathcal{L}$ is given next.

Definition 5 Let $q \in Q$, and ρ_q be the probability distribution on $\Sigma \times Q$ induced by state q. Let $\phi \in \mathcal{L}$, and $\psi : \Sigma \to \mathcal{L}$. The notation ψ_σ will be used for $\psi(\sigma)$, $\sigma \in \Sigma$. Then:

$$\begin{aligned}
\mathbf{1}(q) &= 1 \\
\langle\sigma\rangle\phi(q) &= e\rho_q(\sigma, q_{i(q,\sigma)})\phi(q_{i(q,\sigma)}) \\
\bigvee_{\sigma\in\Theta} \langle\sigma\rangle\psi_\sigma(q) &= \sum_{\sigma\in\Theta} e\rho_q(\sigma, q_{i(q,\sigma)})\psi_\sigma(q_{i(q,\sigma)}) \\
(1-\phi)(q) &= 1 - \phi(q) \\
(\phi \ominus p)(q) &= max(\phi(q) - p, 0)
\end{aligned}$$

where $\sigma \in \Sigma$, and, as before, $i(q,\sigma) = i$ such that $q_i = \delta(q,\sigma)$ if $\delta(q,\sigma)!$, and $i(q,\sigma) = 0$, otherwise.

The presented logic represents a probabilistic modification of Hennessy-Milner logic (Hennessy and Milner 1985). The formula $\mathbf{1}$ corresponds to the constant *true*, $\langle\sigma\rangle\phi$ is the next operator, $1-\phi$ corresponds to negation, and $\phi \ominus p$ provides for the testing of the value of ϕ (Desharnais et al. 2002). The logic only supports disjunctions of the form $\bigvee\langle\sigma\rangle\phi$; extending it to $\bigvee\phi$ would require a more complicated formalization that is unnecessary for the main result to be presented.

The pseudometric d_L is defined next. The distance between two states is measured by a formula that differentiates them the most.

Definition 6 For every $q_q, q_r \in Q$, the pseudometric d_L is defined as:

$$d_L(q_q, q_r) = \sup_{\phi\in\mathcal{L}}\{|\phi(q_q) - \phi(q_r)|\}.$$

It is easy to verify that d_L is indeed a pseudometric.

In this logical setting, the smaller the factor e is, the more discounted the difference is for complex formulae.

An example is given in Fig. 3. States q_0 and q_0' are at the distance $0.35e + 0.65e^2$ in the pseudometric d_L, witnessed by formula $\phi = \bigvee_{\sigma\in\{\alpha,\beta\}}\langle\sigma\rangle\phi_\sigma$, where $\phi_\alpha = 1 - \langle\gamma\rangle\mathbf{1}$,

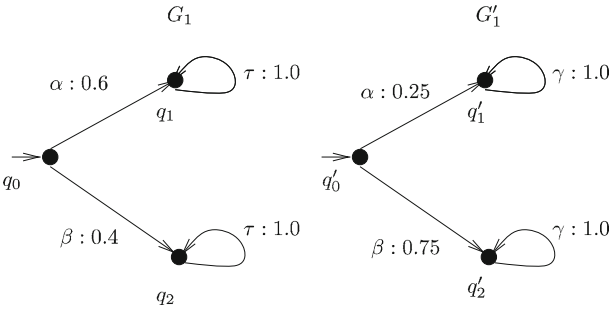


Fig. 3 The distance between G_1 and G_1' (between states q_0 and q_0') in d_L is $0.35e + 0.65e^2$ and is witnessed by formula $\phi = \bigvee_{\sigma\in\{\alpha,\beta\}}\langle\sigma\rangle\phi_\sigma$, where $\phi_\alpha = 1 - \langle\gamma\rangle\mathbf{1}$, and $\phi_\beta = \langle\tau\rangle\mathbf{1}$, i.e., $d_L(q_0, q_0') = |\phi(q_0) - \phi(q_0')|$. The distance between q_1 and q_1' (also, q_1 and q_2') is e, and is witnessed by $\phi = \langle\tau\rangle\mathbf{1}$

and $\phi_\beta = \langle\tau\rangle\mathbf{1}$. Further, states q_1 and q_1' (also, q_1 and q_2') are at the distance e as witnessed by formula $\phi = \langle\tau\rangle\mathbf{1}$.

The goal is to show that pseudometric d_{fp} is equal to pseudometric d_L up to constant e.

Lemma 1 *Let $q_q, q_r \in Q$. For a function $\psi : \Sigma \to \mathcal{L}$, the shorthand notation ψ_σ will be used for $\psi(\sigma)$. Then:*

$$d_L(q_q, q_r) = \sup_{\psi_\sigma \in \mathcal{L}} \left\{ \left| \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle\psi_\sigma(q_q) - \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle\psi_\sigma(q_r) \right| \right\}.$$

Proof The idea of the proof is similar to that of Desharnais et al. (2002), Lemma 4.4. As before, for a function $\varphi : \Sigma \to \mathcal{L}$, the shorthand notation φ_σ will be used for $\varphi(\sigma)$. It should be proven that there exist $\varphi_\sigma \in \mathcal{L}, \sigma \in \Sigma$, such that

$$\left| \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle\varphi_\sigma(q_q) - \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle\varphi_\sigma(q_r) \right| \geq |\phi(q_q) - \phi(q_r)|,$$

for any $\phi \in \mathcal{L}$. Induction on the structure of ϕ is used. The base case ($\phi = \mathbf{1}$) is satisfied. Next, the case when $\phi = \langle\alpha\rangle\phi'$, $\phi' \in \mathcal{L}$, is investigated. It should be shown that

$$\left| \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle\varphi_\sigma(q_q) - \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle\varphi_\sigma(q_r) \right| \geq |\langle\alpha\rangle\phi'(q_q) - \langle\alpha\rangle\phi'(q_r)|.$$

If, for $\sigma \neq \alpha$, $\varphi_\sigma = 1 - \mathbf{1} = 0$, and $\varphi_\sigma = \phi'$ for $\sigma = \alpha$, the inequality is obviously satisfied. The case when $\phi = \bigvee_{\sigma\in\Theta}\langle\sigma\rangle\varphi_\sigma$, for $\Theta \subseteq \Sigma$, is proven in the same manner.

The functions $\phi = 1 - \phi'$ and $\phi = \phi' \ominus p$ are non-expansive (easily shown), so

$$\begin{aligned} |\phi(q_q) - \phi(q_r)| &\leq |\phi'(q_q) - \phi'(q_r)| \\ &\leq \left| \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle\varphi_\sigma(q_q) - \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle\varphi_\sigma(q_r) \right| \end{aligned}$$

by the induction hypothesis on ϕ'. □

The following two definitions will be used for the proof of the main result. First, the depth of a formula $\phi \in \mathcal{L}$ is defined (in a manner similar to that of Desharnais et al. 2002).

Definition 7 The depth of a formula of logic $\mathcal{L}$ is defined as:

$$
\begin{aligned}
depth(\mathbf{1}) &= 0,\\
depth(\langle\sigma\rangle\phi) &= depth(\phi)+1,\\
depth\left(\bigvee_{\sigma\in\Theta}\langle\sigma\rangle\psi_\sigma(q)\right) &= max\{depth(\psi_\sigma)|\sigma\in\Theta\}+1,\\
depth(1-\phi) &= depth(\phi),\\
depth(\phi\ominus p) &= depth(\phi).
\end{aligned}
$$

Now, the formula $\phi^n_{q_q,q_r}$ is introduced.

Definition 8 Let $q_q, q_r \in Q$. The notation adopted for Eq. 4 is used here. Then, formula $\phi^0_{q_q,q_r}$ is defined as

$$\phi^0_{q_q,q_r} = \mathbf{1},$$

and, for $n \in \mathbb{N}$, formula $\phi^{n+1}_{q_q,q_r}$ is defined as

$$\phi^{n+1}_{q_q,q_r} = \bigvee_{\sigma\in\Sigma}\langle\sigma\rangle\psi^n_{\sigma,q_q,q_r}, \text{ where}$$

$$
\psi^n_{\sigma,q_q,q_r} = \begin{cases} 1-((1-\phi^n_{q_i,q_j})\ominus(1-\phi^n_{q_i,q_j}(q_i))), & \text{if } \rho_{\sigma,i}\geq\rho'_{\sigma,j}\\ \phi^n_{q_i,q_j}\ominus\phi^n_{q_i,q_j}(q_j), & \text{otherwise.}\end{cases}
$$

The main result relating the two pseudometrics is presented next. It states that d_L and d_{fp} are equal up to constant e.

Theorem 3 $d_L = ed_{fp}$

Proof The proof consists of two parts. In the first part, it is proven that, for every q_q, q_r, there exists $\phi \in \mathcal{L}$ such that $\phi(q_q) - \phi(q_r) = ed_{fp}(q_q, q_r)$. Consequently, $d_L(q_q, q_r) \geq ed_{fp}(q_q, q_r)$. In the second part, inequality $d_L(q_q, q_r) \leq ed_{fp}(q_q, q_r)$ is proven.

First, let us prove that for every q_q, q_r, there exists $\phi \in \mathcal{L}$ such that $\phi(q_q) - \phi(q_r) = ed_{fp}(q_q, q_r)$. Given Definition 3, it is sufficient to prove that $\phi^n_{q_q,q_r}(q_q) - \phi^n_{q_q,q_r}(q_r) = ed^n_{fp}(q_q, q_r)$, for every $n \in \mathbb{N}$, where $\phi^n_{q_q,q_r}$ is given as in Definition 8. The proof is by induction. The base case is satisfied, since $\phi^0_{q_q,q_r}(q_q) = \phi^0_{q_q,q_r}(q_r) = 1$, and $d^0_{fp}(q_q, q_r) = 0$ according to Definition 3. Now assume that for some $n \in \mathbb{N}$, we have for every $q_q, q_r \in Q$:

$$\phi^n_{q_q,q_r}(q_q) - \phi^n_{q_q,q_r}(q_r) = ed^n_{fp}(q_q, q_r).$$

Also, 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. Also, for notational convenience, we will write $\rho_{\sigma,i}$ instead of $\rho_{q_q}(\sigma, q_i)$, and, similarly, $\rho'_{\sigma,j}$ instead of $\rho_{q_r}(\sigma, q_j)$ for any i, j such that $0 \leq i, j \leq N-1$. Then, for $\sigma \in \Sigma$, 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, let $j(q_r, \sigma) = j$ 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 $j(q_r, \sigma)$. Then:

$$
\begin{aligned}
&\phi^{n+1}_{q_q,q_r}(q_q) - \phi^{n+1}_{q_q,q_r}(q_r) \\
&\quad = \left(\sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} \geq \rho'_{\sigma,j}\}} e\rho_{\sigma,i} + \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} < \rho'_{\sigma,j}\}} e\rho_{\sigma,i} e d^n_{fp}(q_i, q_j) \right) \\
&\qquad - \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} \geq \rho'_{\sigma,j}\}} e\rho'_{\sigma,j}(1 - e d^n_{fp}(q_i, q_j)) \\
&\quad \text{(by the definition of } \phi^{n+1}_{q_q,q_r} \text{ and the induction hypothesis)} \\
&\quad = \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} \geq \rho'_{\sigma,j}\}} \left(e(\rho_{\sigma,i} - \rho'_{\sigma,j}) + e^2 \rho'_{\sigma,j} d^n_{fp}(q_i, q_j) \right) \\
&\qquad + \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} < \rho'_{\sigma,j}\}} e^2 \rho_{\sigma,i} d^n_{fp}(q_i, q_j) \\
&\quad = e \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} \geq \rho'_{\sigma,j}\}} \left(\rho_{\sigma,i} - \rho_{\sigma,j} + e\rho'_{\sigma,j} d^n_{fp}(q_i, q_j) \right) \\
&\qquad + e \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} < \rho'_{\sigma,j}\}} e\rho_{\sigma,i} d^n_{fp}(q_i, q_j) \\
&\quad = e d^{n+1}_{fp}(q_q, q_r) \text{ (follows from Eq. 5)}
\end{aligned}
$$

Next, the induction on the depth of formula is used to prove that $d_L(q_q, q_r) \leq e d_{fp}(q_q, q_r)$ by proving that $d^n_L(q_q, q_r) \leq e d^n_{fp}(q_q, q_r)$ for any $n \in \mathbb{N}$, where

$$
d^n_L(q_q, q_r) = \sup_{\phi \in \mathcal{L}} \{ |\phi(q_q) - \phi(q_r)| \;\; | depth(\phi) \leq n \}.
$$

The base case is satisfied as $d^0_L(q_q, q_r) = d^0_{fp}(q_q, q_r) = 0$. For $n \in \mathbb{N}$, assume:

$$
d^n_L(q_q, q_r) \leq e d^n_{fp}(q_q, q_r).
$$

Then, according to Lemma 1, and the definition of function *depth*:

$$
\begin{aligned}
& d_L^{n+1}(q_q, q_r) \\
& \quad = \sup_{\phi_\sigma^n \in \mathcal{L}} \left\{ \left| \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle \phi_\sigma^n(q_q) - \bigvee_{\sigma\in\Sigma} \langle\sigma\rangle \phi_\sigma^n(q_r) \right| \right\} \\
& \quad = e \cdot \sup_{\phi_\sigma^n \in \mathcal{L}} \left\{ \sum_{\sigma\in\Sigma} \rho_{\sigma,i} \phi_\sigma^n(q_i) - \sum_{\sigma\in\Sigma} \rho'_{\sigma,j} \phi_\sigma^n(q_j), \right. \\
& \qquad\qquad \left. \sum_{\sigma\in\Sigma} \rho'_{\sigma,j} \phi_\sigma^n(q_j) - \sum_{\sigma\in\Sigma} \rho_{\sigma,i} \phi_\sigma^n(q_i) \right\} \\
& \quad = e \cdot \sup_{\phi_\sigma^n \in \mathcal{L}} \left\{ \sum_{\substack{\sigma\in\Sigma \\ 0\le i\le N-1}} \rho_{\sigma,i} \phi_\sigma^n(q_i) - \sum_{\substack{\sigma\in\Sigma \\ 0\le j\le N-1}} \rho'_{\sigma,j} \phi_\sigma^n(q_j), \right. \\
& \qquad\qquad \left. \sum_{\substack{\sigma\in\Sigma \\ 0\le j\le N-1}} \rho'_{\sigma,j} \phi_\sigma^n(q_j) - \sum_{\substack{\sigma\in\Sigma \\ 0\le i\le N-1}} \rho_{\sigma,i} \phi_\sigma^n(q_i) \right\} \\
& \qquad \text{(as } G \text{ is deterministic)}
\end{aligned}
$$

where, for any $\sigma, \alpha \in \Sigma$, $|\phi_\sigma^n(q_i) - \phi_\alpha^n(q_j)| \le d_L^n(q_i, q_j) \le e d_{fp}^n(q_i, q_j)$ (by induction hypothesis). In Pantelic and Lawford (2009), the function in Equation (2) of that paper is a pseudometric (therefore, symmetry holds), and, for $a_{\sigma,i}$ (of that equation) equal to $\phi_\sigma^n(q_i)$, the constraints of that equation are satisfied, so $d_L^{n+1}(q_q, q_r) \le e d_{fp}^{n+1}(q_q, q_r)$. □

Therefore, Theorem 3 proves that pseudometric d_{fp} can be characterized in terms of logic $\mathcal{L}$. Hence, pseudometric d_{fp} measures the extent to which two states (systems) differ in satisfying properties in logic $\mathcal{L}$.

4 From logic to traces

In this section, the logical characterization from the previous section is used for the derivation of the trace characterization of the pseudometric. The section answers the question of how the pseudometric relates to the difference in the (discounted) probabilities of traces generated by systems.

First, $L_p(G)(s)$ is modified to define the *discounted probability of a string s* in G, denoted $P_d(G)(s)$.

Definition 9 Let $P_d(G) : L(G) \to [0, 1]$ be defined as:

$$
\begin{aligned}
& P_d(G)(\epsilon) = 1 \\
& P_d(G)(s\sigma) = \begin{cases} e \cdot P_d(G)(s) \cdot p(\delta(q_0, s), \sigma), & \text{if } \delta(q_0, s)! \\ 0, & \text{otherwise} \end{cases}
\end{aligned}
$$

where $s \in L(G)$, $\sigma \in \Sigma$. Then, $P_d(G)(s)$ is the *discounted probability of a string s* in G.

Informally, the discounted probability of a string is the probability of occurrence of the string discounted by factor e for every event in the string, i.e. $P_d(G)(s) = e^{|s|} L_p(G)(s)$.

Let G_1 and G_2 be two probabilistic generators. An important result states that there is not a string whose discounted probabilities differ by more than the distance d_L between the corresponding generators.

Theorem 4

$$d_L(G_1, G_2) \geq \sup_{s \in \Sigma^*} \{|P_d(G_1)(s) - P_d(G_2)(s)|\} \tag{7}$$

Proof Let t be the string for which the supremum in Eq. 7 is reached. The formula corresponding to this distance is easily constructed. Assume that $t = \sigma_1 \sigma_2 \ldots \sigma_n$. Then, the formula is given as $\phi = \langle \sigma_1 \rangle \langle \sigma_2 \rangle \ldots \langle \sigma_n \rangle \mathbf{1}$. □

Further, it can be shown that distance in the pseudometric d_L between the two systems is also greater than the difference in discounted probabilities of a set of strings such that none of the strings is a substring of another. Let $\Gamma \subseteq \Sigma^*$, such that no string in Γ is a prefix of another string in Γ. Then:

Theorem 5

$$d_L(G_1, G_2) \geq \sup_{\Gamma \subseteq \Sigma^*} \left\{ \left| \sum_{s \in \Gamma} P_d(G_1)(s) - \sum_{s \in \Gamma} P_d(G_2)(s) \right| \right\}$$

Proof Similar to Theorem 4, by using the disjunction formula. □

Similarly, the correspondence between the discounted probability of strings and formulae in $\mathcal{L}$ can be made for the remaining formulae of Definition 5. Therefore, the pseudometric measures not only the difference in probabilities of strings in two languages (discounted for their lengths), but also the difference in discounted probabilities of a certain set of strings, or some more complicated properties of strings, e.g., whether the discounted probability of a string is greater than a prespecified value.

5 Probabilistic model fitting

We next turn to the *probabilistic model fitting problem*: a non-control problem whose solution will turn out to have control-related implications. The problem is to how to represent a given PDES with a generator of a prespecified graph such that the new representation is at the minimal distance from the old one in pseudometric d_{fp}.

First, the section introduces the probabilistic model fitting problem. Then, the solution of the problem is presented. Next, some applications of model fitting are presented.

Also, it is assumed that $e \in (0, 1]$, unless otherwise stated.

5.1 Probabilistic model fitting: problem and solution

The Probabilistic Model Fitting Problem: Let $G_1 = (Q_1, \Sigma, \delta_1, q_{01}, p_1)$ be a probabilistic generator. Given a nonprobabilistic generator $G_2^{np} = (Q_2, \Sigma, \delta_2, r_0)$ such that $G_1^{np} \parallel G_2^{np}$ is isomorphic to G_2^{np}, find the statewise event probability distribution p_2 such that probabilistic generator $G_2 = (Q_2, \Sigma, \delta_2, r_0, p_2)$ is at the minimal distance from G_1 in pseudometric d_{fp}.

It should be noted that no minimization is done in the construction of the synchronous product of (nonprobabilistic) generators as defined by Definition 2 in Section 2.1.

The idea of solving the problem is as follows. The generator G_1 is to be modified to make G_2^{np} isomorphic (identical up to renaming of states) to a subautomaton of modified G_1^{np}, while the probabilistic language of G_1 is preserved. Then, the distance between G_1 and G_2 is minimized by minimizing the distance between the modified G_1, and G_2. This is allowed as the two distances are the same, since G_1 and its modified version are probabilistic bisimilar:

Lemma 2 *Let G_1 and G_2 be two probabilistic generators. Then, if $L_p(G_1) = L_p(G_2)$, then $d_{fp}(G_1, G_2) = 0$.*

Proof Since $L_p(G_1) = L_p(G_2)$, G_1 and G_2 are probabilistic trace equivalent in the sense of Jou and Smolka (1990). As G_1 and G_2 are deterministic, probabilistic trace equivalence implies probabilistic bisimulation equivalence. Therefore, $d_{fp}(G_1, G_2) = 0$. □

Next, as previously stated, we seek to represent $L_p(G_1)$ with an automaton G_{1a} such that G_2^{np} is isomorphic to a subautomaton of G_{1a}^{np}. Figure 4 illustrates an example. The part of G_{1a} drawn by a solid line corresponds to the subautomaton of G_{1a}^{np} isomorphic to G_2^{np}. In general, the automaton G_{1a} will represent a non-minimal realization of $L_p(G_1)$ (in the sense that it might have more states than G_1, but $L_p(G_1) = L_p(G_{1a})$). Generator G_{1a} can be constructed in the following manner.

1. Self-loop each state of G_2^{np} with events not possible from that state. Formally, $G_{2a}^{np} = (Q_2, \Sigma, \delta_{2a}, r_0)$, where, for $q \in Q_2, \sigma \in \Sigma$:

$$\delta_{2a}(q, \sigma) = \begin{cases} \delta_2(q, \sigma), & \text{if } \delta_2(q, \sigma)! \\ q, & \text{otherwise.} \end{cases}$$

2. Next, let $G_{1a}^{np} = (Q_{1a}, \Sigma, \delta_{1a}, q_0) = G_1^{np} \parallel G_{2a}^{np}$.
3. The probabilistic version of G_{1a}^{np} is $G_{1a} = (Q_{1a}, \Sigma, \delta_{1a}, q_0, p_{1a})$, such that, for all $q \in Q_{1a}, \sigma \in \Sigma$:

$$p_{1a}(q, \sigma) = p_1(r, \sigma),$$

where $r = \delta_1(q_{01}, s)$ for any $s \in L(G_{1a})$ such that $q = \delta_{1a}(q_0, s)$.

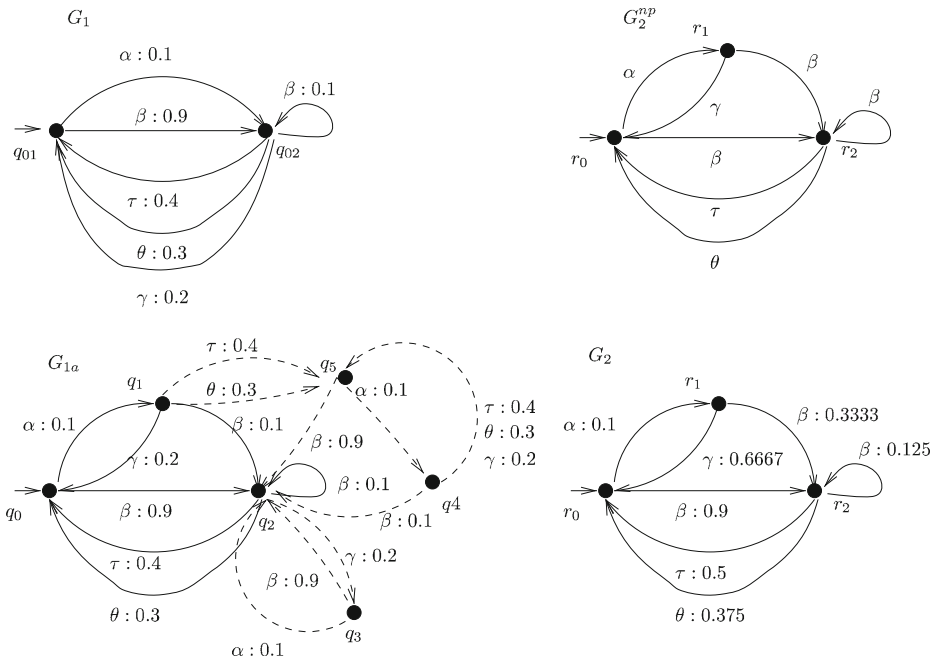


Fig. 4 Model fitting: an example. G_1 is the probabilistic generator that is to be represented using a probabilistic generator G_{1a} with an underlying subgraph isomorphic to G_2^{np}. Therefore, G_{1a} is such that $L_p(G_{1a}) = L_p(G_1)$, and the subgraph of G_{1a} depicted by the solid arcs is isomorphic to G_2^{np}. G_2 is one of the possble solutions to the model fitting problem as given by Theorem 6

Lemma 3 $L_p(G_1) = L_p(G_{1a})$.

Proof Follows from the construction of G_{1a}. □

Now, let $f : Q_2 \to Q_{1a}$ be an embedding (a monomorphism) of G_2^{np} into G_{1a}^{np}, i.e.:

1. $f(r_0) = q_0$,
2. $\forall q \in Q_2 \forall \sigma \in Pos(q)\ \ (f(\delta_2(q,\sigma)) = \delta_{1a}(f(q),\sigma))$.

The function f always exists and is unique. This fact follows from the construction of G_{1a} and the determinism of generators.

Without loss of generality, it is assumed that, $Q_{1a} = \{q_0, \ldots, q_{M-1}\}$, $Q_2 = \{r_0, \ldots, r_{N-1}\}$, and $M \geq N > 0$, $d \in \mathcal{M}$, $q \in Q_2$, where $\mathcal{M}$ is the set of 1-bounded pseudometrics on the states of the system that represents the union of G_{1a} and G_2 (see Remark 3) with the same ordering as in Eq. 3. Next, $i(f(q),\sigma) = i$ such that $q_i = \delta_{1a}(f(q),\sigma)$ if $\delta_{1a}(f(q),\sigma)!$, and $i(f(q),\sigma) = 0$, otherwise. Let $j(q,\sigma) = j$ such that $r_j = \delta_2(q,\sigma)$ if $\delta_2(q,\sigma)!$, and $j(q,\sigma) = 0$, otherwise. For readability purposes, we

will write i instead of $i(f(q), \sigma)$, and j instead of $j(q, \sigma)$. The distance between G_{1a} and G_2 is $d_{fp}(q_0, r_0)$. Also, $f(r_0) = q_0$, and

$$\begin{aligned}
&\mathcal{D}(d)(f(q), q) \\
&= \sum_{\sigma \in \Sigma} max\left(\rho_{\sigma,i} - \rho'_{\sigma,j} + e\rho'_{\sigma,j} d(q_i, r_j), e\rho_{\sigma,i} d(q_i, r_j)\right) \\
&= \sum_{\sigma \in Pos(f(q)) \setminus Pos(q)} \rho_{\sigma,i} \\
&\quad + \sum_{\sigma \in Pos(q)} max\left(\rho_{\sigma,i} - \rho'_{\sigma,j} + e\rho'_{\sigma,j} d(f(r_j), r_j), e\rho_{\sigma,i} d(f(q_j), q_j)\right) \qquad (8) \\
&\quad \text{(since } f(r_j) = q_i, \text{ by the definition of } f)
\end{aligned}$$

where $\rho_{f(q)}$ and ρ_q are the distributions on $\Sigma \times Q$ induced by the states $f(q)$ and q, respectively, and $\rho_{\sigma,i}$ is written instead of $\rho_{f(q)}(\sigma, q_i)$, and, similarly, $\rho'_{\sigma,j}$ instead of $\rho_q(\sigma, r_j)$.

Remark 5 Based on Eq. 8, it can be concluded that, for $q \in Q_2$, the distance between state $f(q) \in Q_{1a}$ and state q depends only on distances between $f(t)$ and t, $t \in Q_2$. E.g., in Fig. 4, the distance between G_{1a} and G_2 depends only on distances between states of pairs (q_0, r_0), (q_1, r_1), and (q_2, r_2); states q_3, q_4, q_5 are irrelevant.

Therefore, in order to calculate the distance between G_{1a} and G_2, only the distances $d_{fp}(f(q), q)$, $q \in Q_2$, are of interest. Hence, the distance between G_{1a} and G_2, for a fixed p_2, can be found by at most ω iterations given in Definition 3, where the domain of d^n_{fp} is restricted to $Q_{1a} \times Q_2$ and only distances between $f(q) \in Q_{1a}$ and $q \in Q_2$ are defined.

This reasoning leads to the solution of the probabilistic model fitting problem as presented next.

Theorem 6 *Let $G_1 = (Q_1, \Sigma, \delta_1, q_{01}, p_1)$ be a probabilistic generator. For a given $G_2^{np} = (Q_2, \Sigma, \delta_2, r_0)$ such that $G_1^{np} \parallel G_2^{np}$ is isomorphic to G_2^{np}, if the statewise event probability distribution p_2 satisfies, for all $r \in Q_2$, $\sigma \in Pos(r)$:*

$$p_2(r, \sigma) \geq p_1(q, \sigma), \qquad (9)$$

where $q = \delta_1(q_{01}, s)$ for any $s \in L(G_2)$ such that $r = \delta_2(r_0, s)$, then $G_2 = (Q_2, \Sigma, \delta_2, r_0, p_2)$ is at the minimal distance from G_1 in the pseudometric d_{fp}.

Proof Let $G_2 = (Q_2, \Sigma, \delta_2, r_0, p_2)$, where p_2 satisfies Eq. 9. Also, let $G'_2 = (Q_2, \Sigma, \delta_2, r_0, p'_2)$ be a probabilistic generator with an arbitrary probability distribution p'_2, and let G_{1a} be the generator constructed from G_1 and G_2^{np} as before in this section. We use induction to show that $d_{fp}(G_{1a}, G'_2) \geq d_{fp}(G_{1a}, G_2)$ by showing that $d^n_{fp}(G_{1a}, G'_2) \geq d^n_{fp}(G_{1a}, G_2)$, $n \in \mathbb{N}$. For $q \in Q_2$, let $d^n_{fp}(f(q), q)$ be the distance between the states $f(q)$ of G_{1a} and q of G_2, and $d'^n_{fp}(f(q), q)$ be the distance between $f(q)$ of G_{1a} and q of G'_2. The base case is trivially satisfied. Next, assume that, for each $q \in Q_2$, $d'^n_{fp}(f(q), q) \geq d^n_{fp}(f(q), q)$. The functions i and j are defined as for

Eq. 8, and, for $q \in Q_2$, $k(q,\sigma) = k$ such that $r_k = \delta_2(q,\sigma)$ if $\delta_2(q,\sigma)!$, and $k(r,\sigma) = 0$, otherwise. The shorthand notation k will be used. For $q \in Q_2$, let $\rho_{f(q)}$, ν_q and ν'_q be the distributions induced by the states $f(q)$ of G_{1a}, q of G_2 and q of G'_2, respectively. Also, for $q \in Q_2$, let $\rho_{\sigma,i}$ be used instead of $\rho_{f(q)}(\sigma, q_i)$, and, similarly, $\nu_{\sigma,j}$ instead of $\nu_q(\sigma, r_j)$ and $\nu'_{\sigma,k}$ instead of $\nu'_q(\sigma, r_k)$. Then:

$$
\begin{aligned}
& d'^{n+1}_{fp}(f(q), q) \\
&\quad = \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,i} \geq \nu'_{\sigma,k}\}} \left(\rho_{\sigma,i} - \nu'_{\sigma,k} + e\nu'_{\sigma,k} d'^{n}_{fp}(q_i, r_k) \right) + \sum_{\sigma \in \{\sigma \in \Sigma \mid \rho_{\sigma,q_i} < \nu'_{\sigma,k}\}} e\rho_{\sigma,i} d'^{n}_{fp}(q_i, r_k)
\end{aligned}
$$

(follows from Eqs. 4 and 5)

$$
\begin{aligned}
&= \sum_{\sigma \in Pos(f(q)) \setminus Pos(q)} \rho_{\sigma,i} + \sum_{\sigma \in \{\sigma \in Pos(q) \mid \rho_{\sigma,i} \geq \nu'_{\sigma,k}\}} \left(\rho_{\sigma,i} - \nu'_{\sigma,k} + e\nu'_{\sigma,k} d'^{n}_{fp}(q_i, r_k) \right) \\
&\quad + \sum_{\sigma \in \{\sigma \in Pos(q) \mid \rho_{\sigma,i} < \nu'_{\sigma,k}\}} e\rho_{\sigma,i} d'^{n}_{fp}(q_i, r_k) \\
&\geq \sum_{\sigma \in Pos(f(q)) \setminus Pos(q)} \rho_{\sigma,i} + \sum_{\sigma \in Pos(q)} e\rho_{\sigma,i} d'^{n}_{fp}(q_i, r_k) \\
&\geq \sum_{\sigma \in Pos(f(q)) \setminus Pos(q)} \rho_{\sigma,i} + \sum_{\sigma \in Pos(q)} e\rho_{\sigma,i} d^{n}_{fp}(q_i, r_k)
\end{aligned}
$$

(because of induction hypothesis, since $q_i = f(r_k)$)

$$
\begin{aligned}
&= \sum_{\sigma \in Pos(f(q)) \setminus Pos(q)} \rho_{\sigma,i} \\
&\quad + \sum_{\sigma \in Pos(q)} max \left(\rho_{\sigma,i} - \nu_{\sigma,j} + e\nu_{\sigma,j} d^{n}_{fp}(q_i, r_j), e\rho_{\sigma,i} d^{n}_{fp}(q_i, r_j) \right)
\end{aligned}
$$

(since $\nu_{\sigma,j} \geq \rho_{\sigma,i}$ for every $\sigma \in Pos(q)$, as follows from Eq. 9)

$$
= d^{n+1}_{fp}(f(q), q)
$$

□

Therefore, the new model is not unique: as long as the probabilities of the events possible in the new model increase or stay the same, the new model is at the minimal distance from the original one. For the example from Fig. 4, one of the possible solutions is represented by the generator at the bottom right corner of the figure. In another possible solution, the probabilities of occurrence of β and γ at the state r_1 would be 0.2 and 0.8, respectively. Therefore, the fitting can be performed by any redistribution of the probabilities of events that are not possible anymore over the possible ones. Hence, model fitting can accommodate some further requirements on p_2.

5.2 Some applications of model fitting

Other than the obvious use of the presented fitting to simplify and reduce the state space of probabilistic systems, the fitting has much more significant control implications.

As mentioned before, it is possible to choose probabilities of events in the new system to a certain extent: as long as they are greater than or equal to the original ones. However, some of the further requirements on p_2 cannot be accommodated by Theorem 6 (e.g., an obvious one would be that the probability of an event still possible in the new system be smaller than in the original system). If the restrictions are given on probabilities of events, statewise, a straightforward modification of the OPSCP algorithm of Section 2.5.2 for $e \in (0, 1)$ would suffice. An example of such an additional requirement would be that the probability of a certain event from a state is less than a specified value, that is, in turn, smaller than the original one.

Further, in the solution of the OPSCP problem presented in Section 2.5.1, in order for the first, maximal permissiveness requirement as presented in Section 2.5.1 to be satisfied, the supremal controllable sublanguage of $L(G_r)$ with respect to G_p is generated. Then, the distance between the controlled plant, and the probabilistic requirement now restricted to the supremal controllable sublanguage, with normalized probabilities, is minimized. Intuitively, after satisfying the nonprobabilistic requirement, and before the probabilistic part is handled, it makes sense for a designer to modify the original requirement so that its nonprobabilistic behaviour matches the one achievable. Then, the probabilities are revised accordingly: probabilities of the events that are inadmissible because they do not satisfy the nonprobabilistic requirement, are redistributed over the admissible ones. The redistribution is such that the probability of an event in the new system is proportional to its original probability. Theorem 6 proves that this normalization is justified in a strict mathematical sense, as the new model that is normalized is at the minimal possible distance from the original one in the pseudometric d_{fp}. However, a revised specification is going to be at the minimal possible distance from the original one, as long as the probabilities of its remaining events are greater than or equal to the original ones: a designer has a freedom to choose how to redistribute the probabilities over the events that are still possible.

Further, the transformation of G_1 into G_{1a} presented here can be used in a modification of the OPSCP algorithm to solve the OPSCP (as presented in Section 2.5.1) with requirement 2) changed so that the controlled plant is "as close as possible" to the unmodified requirement G_r. This modification is shown in detail in the next section.

6 Model fitting and OPSCP: problem revisited

In this section, the OPSCP is revisited. First, in Section 6.1, the OPSCP is modified, and the solution of the problem is presented. Next, in Section 6.2, a complexity analysis of the solution is offered, and an example is presented.

6.1 Revisiting OPSCP

In Section 2.5, after satisfying the nonprobabilistic criterion (the first part of the OPSCP of Section 2.5.1), a designer revises the requirements specification before satisfying the probabilistic criterion (the second part of the OPSCP of Section 2.5.1). In this section, after satisfying the nonprobabilistic requirement, the distance between the achievable behaviour of the plant under control and the original requirements specification is minimized.

Modified Optimal Probabilistic Supervisory Control Problem (Modified OPSCP) Let $G_p = (Q_p, \Sigma, \delta_p, q_{p_0}, p_p)$ be a plant PDES, and let $G_r = (Q_r, \Sigma, \delta_r, q_{r_0}, p_r)$ be a requirements specification represented as a PDES. If there is no probabilistic supervisor V_p such that $L_p(V_p/G_p) = L_p(G_r)$ (i.e., the conditions of Theorem 1 fail), find, if it exists, V_p such that

1. $L(V_p/G_p) \subseteq L(G_r)$ and supervisor V_p is maximally permissive in the nonprobabilistic sense (i.e., $L(V_p/G_p)$ is the supremal controllable sublanguage of $L(G_r)$ with respect to G_p).
2. The probabilistic behaviour of the controlled plant is "as close as possible" to the probabilistic behaviour of the requirements specification.

Let $e \in (0, 1)$. Formally, let the reachable and deadlock-free DES $G_{1k} = (T, \Sigma, \zeta, t_0)$ represent language K, the supremal controllable sublanguage as defined in Section 2.5.2. Then, $G_1 = (T, \Sigma, \zeta, t_0, p_1)$ is defined in the same manner as in Section 2.5.2—it is the probabilistic automaton corresponding to the restriction of the plant G_p to K. Next, the requirement is not normalized as before, but, instead, the language $L_p(G_r)$ is represented using the generator $G_2 = (Q, \Sigma, \delta, q_0, p)$, such that a subautomaton of G_2^{np} is isomorphic to G_{1k}; hence, isomorphic to G_1^{np}, too (see Fig. 5 for an illustration). The part of G_2 drawn by a solid line corresponds to the subautomaton of G_2^{np} isomorphic to G_1^{np}. As before, we should find p' in $G_2' = (Q', \Sigma, \delta', q_0', p')$, such that $L_p(V_p/G_1) = L_p(G_2')$ holds, and G_2' is closest to G_2 in our chosen metric. Also, $G_2'^{np}$ is such that it is isomorphic to G_1^{np}. This comes from the fact that there cannot be any string in the desired system that does not belong to K, and, therefore, there cannot be any string in the desired system that does not belong to $L(G_1)$ (as explained in Section 2.5.2). This comes from the fact that K is the reachable and deadlock-free supremal controllable sublanguage: if any string not in K would be allowed in the controlled plant, either the safety or nontermination requirement would not be met. It follows from Lemma 3 that minimizing the distance between the G_r and G_2' is the same as minimizing the distance between G_2 and G_2'. Also, generator G_2 can be constructed from G_r and G_1 in the same manner as G_{1a} was constructed from G_2 and G_1 in Section 5, and, according to the results of Section 5, the construction is possible, as $G_r^{np} \parallel G_1^{np}$ is isomorphic to G_1^{np}. Now, given the definitions of G_2 and G_2', there exists an embedding $f : Q' \to Q$ of $G_2'^{np}$ to G_2^{np}, i.e.:

1. $f(q_0') = q_0$,

2. $\forall q \in Q' \; \forall \sigma \in Pos(q) \; (f(\delta'(q, \sigma)) = \delta(f(q), \sigma))$.

We assume that $T = \{t_0, t_1, \ldots t_{N-1}\}$, $Q = \{q_0, q_1, \ldots q_{M-1}\}$, and $Q' = \{q'_0, q'_1, \ldots q'_{N-1}\}$, where $q_i = f(q'_i)$, $t_i = h(q'_i)$, $i = 0, \ldots, N-1$, where $M \geq N > 0$, and h is the isomorphism between $G_2^{'np}$ and G_1^{np}. Let ρ_{q_i} be the probability distribution induced by the state $q_i \in Q$ of PDES G_2 and let $\rho'_{q'_i}$ be the probability distribution induced by the state $q'_i \in Q'$ of PDES G'_2. Also, we will write j for $j(i, \sigma)$, then $\rho_{q_i,\sigma}$ instead of $\rho_{q_i}(\sigma, q_k)$, and $\rho'_{q'_i,\sigma}$ instead of $\rho'_{q'_i}(\sigma, q'_k)$, $k = 0, 1, \ldots, N-1$. Let $\mathcal{A}$ be the class of all 1-bounded pseudometrics on the states of the system that represents the union of G_2 and G'_2, with domain reduced to $Q \times Q'$, and only distances between $q = f(q') \in Q$ and $q' \in Q'$ defined. Let $d \in \mathcal{A}$, $0 \leq i \leq N-1$, $\Psi(q_i) = Pos(q_i)$, $\Psi(q'_i) = Pos(q'_i)$, $\Psi_u(q'_i) = Pos(q'_i) \cap \Sigma_u$, and $\Psi_c(q'_i) = Pos(q'_i) \cap \Sigma_c$.

Let $c_j = e \cdot d(q_j, q'_j)$ such that $q_j = \delta(q_i, \sigma)$. Note that, since,

$$\mathcal{D}(d)(q_i, q'_i) = \sum_{\sigma \in \Psi(q_i) \setminus \Psi(q'_i)} \rho_{q_i,\sigma} + \sum_{\sigma \in \Psi(q'_i)} max\left(\rho_{q_i,\sigma} - \rho'_{q'_i,\sigma} + c_j \rho'_{q'_i,\sigma}, c_j \rho_{q_i,\sigma}\right), \quad (10)$$

the distance between G_2 and G'_2 is going to depend only on distances between the isomorphic states. E.g., in Fig. 5, the distance between G_2 and G'_2 depends only on distances between states of pairs (q_0, q'_0), (q_1, q'_1), and (q_2, q'_2); states q_3, q_4, q_5 are irrelevant.

Theorem 7 *Let $d^0(q_i, q'_i) = 0$. The distance $d^n(q_i, q'_i)$ in the n-th iteration ($n > 0$) is given as:*

$$\textit{Minimize} \sum_{\sigma \in \Psi(q_i) \setminus \Psi(q'_i)} \rho_{q_i,\sigma} + \sum_{\sigma \in \Psi(q'_i)} y_{q_i,\sigma} \quad (11)$$

subject to

$$\rho_{q_i,\sigma} - \rho'_{q'_i,\sigma} + c_j \rho'_{q'_i,\sigma} \leq y_{q_i,\sigma}, \qquad \sigma \in \Psi(q'_i),$$

$$c_j \rho_{q_i,\sigma} \leq y_{q_i,\sigma}, \qquad \sigma \in \Psi(q'_i),$$

$$\textit{where } c_j = e \cdot d^{n-1}(q_j, q'_j) \textit{ s.t. } q_j = \delta(q_i, \sigma),$$

$$p_1(t_i, \sigma) \sum_{\alpha \in \Psi_u(q_i)} \rho'_{q_i,\alpha} = \rho'_{q'_i,\sigma} \sum_{\alpha \in \Psi_u(q_i)} p_1(t_i, \alpha), \quad \sigma \in \Psi_u(q'_i), \quad (12)$$

$$\frac{\sum_{\alpha \in \Psi_u} p_1(t_i, \alpha)}{p_1(t_i, \sigma)} \rho'_{q'_i,\sigma} + \sum_{\alpha \in \Psi_c(q'_i)} \rho'_{q'_i,\alpha} \leq 1, \quad \sigma \in \Psi_c(q'_i),$$

$$0 \leq \rho'_{q'_i,\sigma} \leq 1, \qquad \sigma \in \Psi(q'_i),$$

$$\sum_{\alpha \in \Psi(q'_i)} \rho'_{q'_i,\alpha} = 1.$$

After the n-th iteration, the values of decision variables $\rho'_{q'_i,\sigma}$ that represent the unknown transition probabilities, are such that the distance between the (initial states of) systems G_2 and G'_2 is within e^n of the minimal achievable distance between the two systems (in pseudometric d_{fp}).

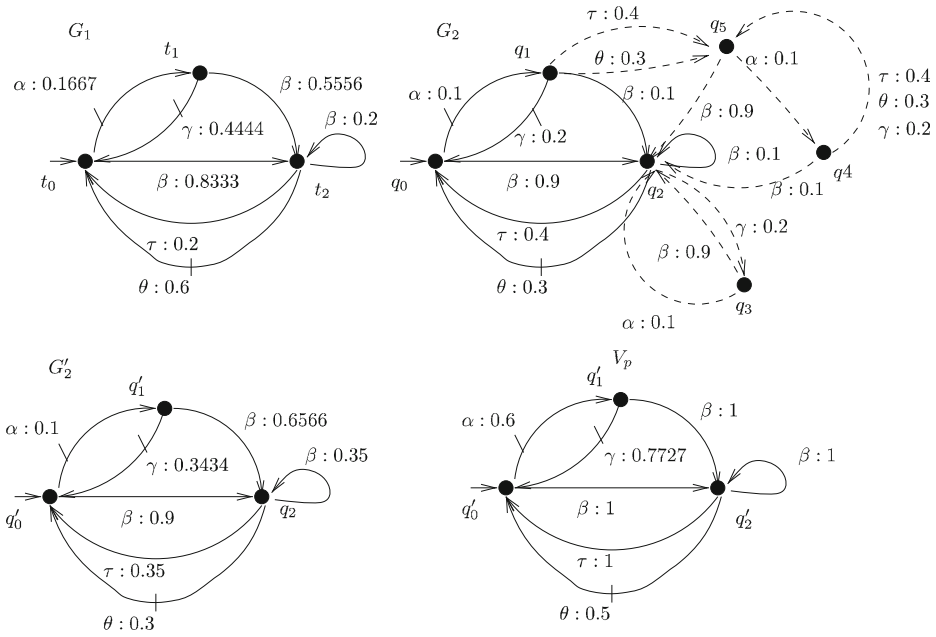


Fig. 5 Illustrating the modified OPSCP algorithm for the example from Fig. 1. The nonprobabilistic languages generated by automata G_1, G_2' and the solid part of automaton G_2, correspond to K, the supremal controllable sublanguage of $L(G_r)$ with respect to G_p. G_1 is the normalized plant, G_2 is such that $L_p(G_2) = L_p(G_r)$, and, at first, the probabilities of G_2' are unknown. These probabilities are found by minimizing the distance between G_2 and G_2', i.e., minimizing the distance between states q_0 and q_0', q_1 and q_1', q_2 and q_2'. When G_2' is calculated, the probabilistic supervisor V_p such that $L_p(V_p/G_p) = L_p(G_2')$ is calculated by using the algorithm of Postma and Lawford (2004) and Pantelic et al. (2009)

Proof The proof follows from Eq. 10 and the proof of Theorem 2. □

In summary, the algorithm first finds language K, the supremal controllable sublanguage of $L(G_r)$ with respect to G_p. Probabilistic generator G_2' that represents a closest approximation is such that its underlying graph $G_2'^{np}$ generates exactly sublanguage K, while the probabilities of G_2' are yet to be determined. The probabilistic language $L_p(G_r)$ of the requirements specification G_r can be exactly represented by a probabilistic generator G_2 whose underlying nonprobabilistic automaton, G_2^{np}, has a subautomaton that is isomorphic to the automaton that represents the supremal controllable sublanguage of the controlled plant, K. Then, the probabilities of the controlled plant G_2' are approximated by minimizing the distance in d_{fp} (under the probabilistic controllability conditions of Theorem 1) between these two generators representing the controlled plant G_2', and the probabilistic requirement G_2. An iterative algorithm is used to minimize the distance. More precisely, using the reasoning of Remark 5, the distance between the requirement and the controlled plant depends only on the distances between isomorphic states of the subautomaton of G_2^{np} and G_2'. So, in each iteration, the distance is minimized by minimizing the distance between each pair of the isomorphic states. The algorithm iterates until a prespecified accuracy is reached.

Again, note that the aforementioned results hold for $e \in (0, 1)$.

6.2 Complexity of the algorithm and an example

Either the simplex method or an interior point method can be used to solve the linear programming problem 11. While the worst-case time complexity of the simplex method is exponential in the number of events possible from state q_i, the worst-case time complexity of an interior method is polynomial. Still, the simplex method is typically very efficient, especially since the number of events possible from a state is typically small.

Let the prespecified accuracy be $\epsilon \in (0, 1)$, and let $m = max\{|Pos(q)||q \in T\}$. Then, in the case that the simplex method is used, the worst-case time complexity of the algorithm is $O(|Q_p| \cdot |Q_r| \cdot (|\Sigma| + k^m \cdot \lceil \log_e \epsilon \rceil))$, $k \geq 2$ (for more details, see Pantelic and Lawford 2012). Factor $\lceil \log_e \epsilon \rceil$ corresponds to the number of iterations sufficient to reach accuracy ϵ. This factor is the same as in the algorithms of Ferns et al. (2004, 2005, 2006), while the number of iterations in the algorithm of van Breugel and Worrell (2001, 2006) is $\lceil \log_e \epsilon/2 \rceil$. These algorithms calculate distances in pseudometrics related to d_{fp} (for more discussion on the topic, see Pantelic and Lawford 2012).

For the example from Fig. 1, 10 iterations of the algorithm are sufficient to find a closest approximation, with the accuracy $\epsilon = 0.001$, and $e = 0.5$. It took 0.25 s on a 2.6 GHz dual core Opteron processor with 8 GB of RAM running Red Hat Enterprise Linux Server 5.5. The algorithm is partially shown in Fig. 5.

7 Conclusions

The paper presents a pseudometric as a tool to measure the behavioural similarity between probabilistic generators. The pseudometric has been adopted from the theoretical computer science community, and used in Pantelic and Lawford (2009, 2012) to solve an optimal supervisory control problem (namely, the OPSCP).

This paper gives a logical characterization of the pseudometric that offers a better insight into the core of the pseudometric from both logic and language standpoints. Further, the pseudometric is used in the probabilistic model fitting problem: a probabilistic system represented by a probabilistic generator is represented using another probabilistic generator of prespecified structure, such that that the new representation is at the minimal distance from the original one. The fitting has a number of applications. One of the transformations used in the solution of the fitting problem is reused in the solution of the modified OPSCP problem.

Operators on probabilistic generators remain to be defined (prefixing, choice operators, parallel composition, etc.). The desired property of non-expansiveness of operators with the respect to the pseudometric merits further study.

References

Arnold A (1994) Finite transition systems. Prentice Hall
Blackwell D (1962) Discrete dynamic programming. Ann Math Stat 33(2):719–726

Chattopadhyay I, Ray A (2008) Structural transformations of probabilistic finite state machines. Int J Control 81(5):820–835

Chattopadhyay I, Mallapragada G, Ray A (2009) $\nu^\star$: a robot path planning algorithm based on renormalized measure of probabilistic regular languages. Int J Control 82(5):849–867

de Alfaro L, Henzinger TA, Majumdar R (2003) Discounting the future in systems theory. In: Baeten JCM, Lenstra JK, Parrow J, Woeginger GJ (eds) Proceedings of international colloquium on automata, languages and programming. Lecture Notes in Computer Science, vol 2719. Springer, pp 1022–1037

Deng Y, Du W (2009) The Kantorovich metric in computer science: a brief survey. Electron Notes Theor Comput Sci 253:73–82

Deng Y, Chothia T, Palamidessi C, Pang J (2006) Metrics for action-labelled quantitative transition systems. Electron Notes Theor Comput Sci 153(2):79–96

Desharnais J, Gupta V, Jagadeesan R, Panangaden P (1999) Metrics for labeled Markov systems. In: Baeten JCM, Mauw S (eds) Proceedings of the 10th international conference on concurrency theory. Lecture Notes in Computer Science, vol 1664. Springer, pp 258–273

Desharnais J, Jagadeesan R, Gupta V, Panangaden P (2002) The metric analogue of weak bisimulation for probabilistic processes. In: Proceedings of the 17th annual IEEE symposium on logic in computer science, IEEE Computer Society, Washington, DC, USA, pp 413–422

Desharnais J, Gupta V, Jagadeesan R, Panangaden P (2004) Metrics for labelled Markov processes. Theor Comp Sci 318(3):323–354

Ferns N, Panangaden P, Precup D (2004) Metrics for finite Markov decision processes. In: Proceedings of the 20th conference on uncertainty in artificial intelligence (UAI-04). Banff, Canada, AUAI Press, Arlington, Virginia, pp 162–169, 07–11 July 2004

Ferns N, Panangaden P, Precup D (2005) Metrics for Markov Decision Processes with infinite state spaces. In: Proceedings of the 21st conference in uncertainty in artificial intelligence. Edinburgh, Scotland, AUAI Press, Cambridge, MA, USA, pp 201–208, 26–29 July 2005

Ferns N, Castro PS, Precup D, Panangaden P (2006) Methods for computing state similarity in Markov decision processes. In: Proceedings of the 22nd conference on uncertainty in artificial intelligence, AUAI Press, Cambridge, MA, USA, pp 174–181, 13–16 July 2006

Garg V (1992a) An algebraic approach to modeling probabilistic discrete event systems. In: Proceedings of 31st IEEE conference on decision and control, Tucson, AZ, USA, pp 2348–2353

Garg V (1992b) Probabilistic languages for modeling of DEDS. In: Proceedings of 26th conference on information sciences and systems, vol 1. Princeton, NJ, pp 198–203

Giacalone A, Jou C, Smolka S (1990) Algebraic reasoning for probabilistic concurrent systems. In: Broy M, Jones CB (eds) Proceedings of the working conference on programming concepts and methods, North-Holland, Sea of Gallilee, Israel, pp 443–458

Hennessy M, Milner R (1985) Algebraic laws for nondeterminism and concurrency. J ACM 32(1):137–161

Hutchinson JE (1981) Fractals and self-similarity. Indiana Univ Math J 30(5):713–747

Jou CC, Smolka SA (1990) Equivalences, congruences, and complete axiomatizations for probabilistic processes. In: Baeten JCM, Klop JW (eds) Proceedings of international conference on concurrency theory. Lecture notes in computer science, vol 458. Springer, pp 367–383

Kantorovich L (1942) On the transfer of masses (in Russian). Dokl Akad Nauk 37(2):227–229; translated in Manage Sci, 5:(1–4) (1959)

Koeppl H, Setti G, Pelet S, Mangia M, Petrov T, Peter M (2010) Probability metrics to calibrate stochastic chemical kinetics. In: Proceedings of 2010 IEEE international symposium on circuits and systems (ISCAS), pp 541–544

Kozen D (1985) A probabilistic PDL. J Comput Cyst Sci 30(2):162–178

Kumar R, Garg V (1998) Control of stochastic discrete event systems: existence. In: Proceedings of 1998 international workshop on discrete event systems, Cagliari, Italy, pp 24–29

Larsen KG, Skou A (1991) Bisimulation through probabilistic testing. Inf Comput 94(1):1–28

Lawford M, Wonham W (1993) Supervisory control of probabilistic discrete event systems. In: Proceedings of the 36th IEEE Midwest symposium on circuits and systems, IEEE, vol 1, pp 327–331

Li Y, Lin F, Lin ZH (1998) Supervisory control of probabilistic discrete event systems with recovery. IEEE Trans Automat Contr 44(10):1971–1975

Mallapragada G, Chattopadhyay I, Ray A (2009) Autonomous robot navigation using optimal control of probabilistic regular languages. Int J Control 82(1):13–26

Pantelic V (2011) Probabilistic supervisory control of probabilistic discrete event systems. PhD thesis, McMaster University, Hamilton, ON, Canada

Pantelic V, Lawford M (2009) Towards optimal supervisory control of probabilistic discrete event systems. In: Proceedings of 2nd IFAC workshop on dependable control of discrete systems (DCDS 2009). Bari, Italy, pp 85–90

Pantelic V, Lawford M (2010) Use of a metric in supervisory control of probabilistic discrete event systems. In: Proceedings of the 10th international workshop on discrete event systems (WODES 2010), pp 227–232, 30 August–1 September

Pantelic V, Lawford M (2012) Optimal supervisory control of probabilistic discrete event systems. IEEE Trans Automat Contr (in press)

Pantelic V, Postma S, Lawford M (2009) Probabilistic supervisory control of probabilistic discrete event systems. IEEE Trans Automat Contr 54(8):2013–2018

Postma S, Lawford M (2004) Computation of probabilistic supervisory controllers for model matching. In: Veeravalli V, Dullerud G (eds) Proceedings of allerton conference on communications, control, and computing, Monticello, Illinois

Thorsley D, Klavins E (2010) Approximating stochastic biochemical processes with wasserstein pseudometrics. IET Syst Biol 4(3):193–211

van Breugel F, Worrell J (2001) An algorithm for quantitative verification of probabilistic transition systems. In: Larsen KG, Nielsen M (eds) Proceedings of international conference on concurrency theory. Lecture Notes in Computer Science, vol 2154. Springer, pp 336–350

van Breugel F, Worrell J (2005) A behavioural pseudometric for probabilistic transition systems. Theor Comp Sci 331(1):115–142

van Breugel F, Worrell J (2006) Approximating and computing behavioural distances in probabilistic transition systems. Theor Comp Sci 360(1-3):373–385

van Breugel F, Hermida C, Makkai M, Worrell J (2005) An accessible approach to behavioural pseudometrics. In: Caires L, Italiano G, Monteiro L, Palamidessi C, Yung M (eds) Automata, languages and programming. Lecture Notes in Computer Science, vol 3580. Springer Berlin / Heidelberg, pp 1018–1030

van Breugel F, Hermida C, Makkai M, Worrell J (2007) Recursively defined metric spaces without contraction. Theor Comp Sci 380(1–2):143–163

Wasserstein L (1969) Markov processes over denumerable products of spaces describing large systems of automata. Probl Inf Transm 5(3):47–52

Vera Pantelic received her B.Eng. in Electrical Engineering from University of Belgrade, Serbia, in 2001, and both her M.A.Sc, and Ph.D. in Software Engineering from McMaster University, Hamilton, in 2005, and 2011, respectively. She is currently working as a postdoctoral fellow with the McMaster Centre for Software Certification (McSCert). Her main research interests include supervisory control of discrete event systems, and verification and certification of safety-critical software-intensive systems.

Mark Lawford has a B.Sc. ('89) in Engineering Mathematics from Queen's University, Kingston, where he received the University Medal in Engineering Mathematics. His M.A.Sc. ('92) and Ph.D. ('97) are from the Systems Control Group in the Department of Electrical and Computer Engineering at the University of Toronto. His research interests include software certification, application of formal methods to safety critical real-time systems, and supervisory control of discrete event systems.

He worked at Ontario Hydro as a real-time software verification consultant on the Darlington Nuclear Generating Station Shutdown Systems Redesign project, receiving the Ontario Hydro New Technology Award for Automation of Systematic Design Verification of Safety Critical Software in 1999. He joined McMaster University's Department of Computing and Software in 1998 where he helped to develop the Software Engineering programs and Mechatronics Engineering programs. In 2003 he was a guest co-editor of joint special issues on Software Inspection of IEEE Software and IEEE Transactions on Software Engineering. He is a licenced Professional Engineer in the province of Ontario and a Senior Member of the IEEE.