Optimal Supervisory Control of Probabilistic Discrete Event Systems

Vera Pantelic, and Mark Lawford, *Senior Member, IEEE*

Abstract

Probabilistic discrete event systems (PDESs) are modeled as generators of probabilistic languages and the supervisors employed are a probabilistic generalization of deterministic supervisors used in standard supervisory control theory. In the case when there exists no probabilistic supervisor such that the behaviour of a plant under control exactly matches the probabilistic language given as the requirements specification, we want to find a probabilistic control such that the behaviour of the plant under control is "as close as possible" to the desired behaviour. First, as a measure of this proximity, a pseudometric on states of generators is defined. Two algorithms for the calculation of the distance between states in this pseudometric are described. Then, an algorithm to synthesize a probabilistic supervisor that minimizes the distance between generators representing the achievable and required behaviour of the plant is presented.

I. Introduction

The supervisory control theory of discrete event systems (DESs) was developed in the seminal work of Ramadge and Wonham [1]. A supervisor (controller) controls a plant by enabling/disabling controllable events based on the observation of the previous behaviour of the plant. DESs are most often modeled by finite automata whose transitions are labeled with events: therefore, the behaviour of a DES can be represented as a regular language. The supervisory control problem considered is to supervise the plant so it generates a given specification language.

Authors are with the SQRL, Department of Computing and Sofware, Faculty of Engineering, McMaster University, 1280 Main Street West, Hamilton ON, L8S 4K1. (Tel: 905 525-9140 ext. 24991; E-mail: pantelv at mcmaster dot ca (corresponding author's e-mail), lawford at mcmaster dot ca; Fax: 905 524-0340)

Manuscript received July 27, 2010.

Many models of stochastic behaviour of discrete event systems have been proposed (e.g., Markov chains [2], Rabin's probabilistic automata [3], stochastic Petri nets [4]). The work of [5]–[7] uses an algebraic approach. A stochastic discrete event system is represented as an automaton with transitions labeled with events and probabilities. We use this approach. The model is generative: the probabilities of all the events in a certain state add up to at most one. On the other hand, with reactive models, such as Rabin probabilistic automata [3], the sum of the transition probabilities of one event at a state is one. Also, unlike Markov chains [2], the emphasis of the approach of [5]–[7] is on event traces rather than state traces. However, as opposed to the nondeterministic model of [5]–[7], our probabilistic automaton (*probabilistic generator*) is deterministic in the following sense: for each state of the automaton, there is at most one next state to which the automaton can move on a given event.

The control of different models of stochastic discrete event systems has been investigated in [8], [9], etc. Rabin's probabilistic automata are used in [8] as the underlying model. The optimal control theory of Markov chains is widely investigated in the framework of controlled Markov chains, also known as Markov decision processes (e.g., see [9]). A deterministic supervisory control framework for stochastic discrete event systems was developed in [10] using the model of [5]–[7]. In [10], controllable events are disabled dynamically as first suggested in [11], so that the probabilities of their execution become zero, and the probabilities of the occurrence of other events proportionally increase. The control objective considered in [10] is to construct a supervisor such that the controlled plant does not execute specified illegal traces, and occurrences of the legal traces in the system are greater than or equal to specified values. The paper gives necessary and sufficient condition for the existence of a supervisor. Further, in [12], a technique to compute a maximally permissive supervisor on-line is given. In [13], [14], the deterministic version of probabilistic automata used in [5]–[7] (our model) is used. The requirements specification is given by weights assigned to states of a plant and the control goal is, roughly speaking, to reach the states with more weight (more desirable states) more often. A deterministic control is synthesized for a given requirements specification so that a measure based on the specification and probabilities of the plant is optimized. Optimal supervisory theory of probabilistic systems was considered in [15], where the system is allowed to violate the specification, but with a probability lower than a prespecified value.

Controller synthesis for probabilistic systems has also attracted attention in the formal methods

community. E.g., [16], [17] consider different control policies: deterministic or randomized (probabilistic) on one hand; memoryless (Markovian) or history-independent on the other. The systems considered are finite Markov decision processes, where the state space is divided into two disjoint sets: controllable states and uncontrollable states. In [16], the controller synthesis problem for a requirements specification given as a probabilistic computation tree logic (PCTL) formula is shown to be NP-hard, and a synthesis algorithm for automata specifications is presented. Controller synthesis was considered in [17] for a requirements specification given as a formula of PCTL extended with long-run average propositions. It is shown that the existence of such a controller is decidable, and an algorithm for the synthesis of a controller, when it exists, is presented. Further, controller robustness with respect to slight changes in the probabilities of the plant is discussed. The paper shows that the existence of robust controllers is decidable and the controller, if it exists, is effectively computable.

Deterministic control of PDES is easier to deal with than probabilistic control, both from the viewpoint of analysis, and practice. However, probabilistic control of PDES is much more powerful. It has been shown in [11], [18] that probabilistic supervisory control can generate a much larger class of probabilistic languages than deterministic control. In the sense of the supervisory control problem discussed in this paper, the use of deterministic control might be too restrictive for a designer. Hence, [11], [18], [19] investigate probabilistic supervisory control: conditions under which a probabilistic control can generate a prespecified probabilistic language, and, if the supervisor exists, the algorithm for its synthesis. PDESs are modeled as the probabilistic generators from [5]–[7]. Probabilistic supervisors are so named because they employ the control method of *random disablement*: after observing a string s, the probabilistic supervisor enables an event σ with a certain probability.

Our work focuses on optimal supervisory control theory inside this framework. 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 supervisor exists such that the plant under control generates the prespecified language. In this case, when the exact solution is not achievable, a designer tries to find a supervisor such that the plant under control generates a *closest approximation* of the desired behaviour. The nonprobabilistic behaviour of the requirements specification is considered to be a safety constraint in the standard supervisory control sense similar to [10]. Therefore, the supremal controllable sublanguage of the intersection of the plant

and the specification is generated as the maximal achievable legal nonprobabilistic behaviour of the plant under control. Then, the closest approximation is calculated by minimizing the distance between the achievable probabilistic behaviour of the plant under control and the probabilistic behaviour of the specification (whose nonprobabilistic behaviour is reduced to the mentioned supremal controllable sublanguage). The distances used are defined by a pseudometric on states of probabilistic transition systems. This paper also offers two algorithms for the calculation of distances in this pseudometric.

In Section II, PDESs are presented as generators of probabilistic languages, and the probabilistic control of PDES is introduced. The proposed pseudometric is presented in Section III. The rationale for choosing the pseudometric is given, and its characterization as the greatest fixed point of a monotone function (as taken from [20]) is presented. Section IV uses this characterization for the derivation and proof of correctness of two algorithms for the calculation of the distances between the states of a PDES in this pseudometric. Section V presents the algorithm for finding the closest approximation to within a prespecified accuracy. Section VI concludes with avenues for future work.

This paper represents the journal version of results of [21]. Also, the paper offers the full literature review, informal reasoning and formal proofs lacking in [21]. Still, due to space restrictions, not all the proofs can be presented here. For review purposes, they are included in the Appendix.

II. Preliminaries

In this section, PDESs modeled as generators of probabilistic languages are presented. Then, the probabilistic control of PDESs, the probabilistic supervisory problem, and the main results of [11], [18], [19] are introduced.

A. Modeling PDES

Following [11], [18], a probabilistic DES is modeled as a probabilistic generator $G = (Q, \Sigma, \delta, q_0, p)$, 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 $q \in Q$, $\sum_{\sigma \in \Sigma} p(q, \sigma) \leq 1$. The probability that the event $\sigma \in \Sigma$ is going to occur at the state $q \in Q$ is

$p(q,\sigma)$. For generator G to be well-defined, (i) $p(q,\sigma)=0$ should hold if and only if $\delta(q,\sigma)$ is undefined and (ii) $\forall q$ $\sum_{\sigma\in\Sigma} p(q,\sigma)\leq 1$. The probabilistic generator G is nonterminating if, for every reachable state $q\in Q$, $\sum_{\sigma\in\Sigma} p(q,\sigma)=1$. Conversely, G is terminating if there is at least one reachable state $q\in Q$ such that $\sum_{\sigma\in\Sigma} p(q,\sigma)<1$. The probability that the system terminates at state q is $1-\sum_{\sigma\in\Sigma} p(q,\sigma)$. Throughout the sequel, unless stated otherwise, we assume nonterminating generators. If a PDES is terminating, it can easily be transformed into a nonterminating generator using the technique described in [11].

The state transition function is traditionally extended by induction on the length of strings to $\delta: Q\times\Sigma^*\rightarrow Q$ in a natural way. For a state q, and a string s, the expression $\delta(q,s)!$ will denote that δ is defined for the string s in the state q. 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 a probabilistic DES generator $G=(Q,\Sigma,\delta,q_0,p)$ is $L(G)=\{s\in\Sigma^*\mid\delta(q_0,s)!\}$. The probabilistic language generated by G is defined as:

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

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

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

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

Next, the synchronous product of (nonprobabilistic) discrete event systems (DESs) that underlie PDESs is defined in a standard manner. For a probabilistic generator $G=(Q,\Sigma,\delta,q_0,p)$, the (nonprobabilistic) discrete event system (DES) that underlies G will be denoted G^{np} (i.e., $G^{np}=(Q,\Sigma,\delta,q_0)$) throughout the sequel. Let G_1^{np} and G_2^{np} be the nonprobabilistic generators (DESs) 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 1: 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 $\delta_2(q_2, \sigma)!$.

B. Probabilistic Supervisors: Existence and Synthesis

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*. The control technique used is called *random disablement*. Instead of deterministically enabling or disabling controllable events, probabilistic supervisors enable them with certain probabilities. This means that, upon reaching a certain state q, the control pattern is chosen according to supervisor's probability distributions of controllable events. Consequently, the controller does not always enable the same events when in the state q.

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, the supervisor enables event σ with probability $V_p(s)(\sigma)$. After a set of controllable events to be enabled has been decided upon (uncontrollable events are always enabled), the system acts as if supervised by a deterministic supervisor. Given sets A, B, we will denote 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:

$$P(\alpha \text{ in } V_p/G|s) = \sum_{\Theta \in \mathcal{P}(Pos(q) \cap \Sigma_c)} P(\alpha | V_p \text{ enables } \Theta \text{ after } s) \cdot P(V_p \text{ enables } \Theta | s) \qquad (1)$$

where

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

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

The goal is to match the behaviour of the controlled plant with a given probabilistic specification language. We call this problem the *Probabilistic Supervisory Control Problem (PSCP)*. More formally:

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

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.

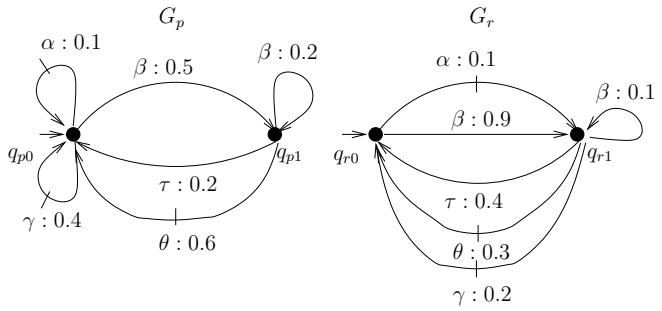

Fig. 1. Plant G_p, and requirements specification G_r

Next, we present the conditions for the existence of a probabilistic supervisor for the PSCP from [11], [18].

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 nonterminating PDESs 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 $\sigma \in Pos(q) \cap \Sigma_u$,

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

(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.

The results hold for nonterminating generators. Terminating PDESs can be transformed into nonterminating as shown in [11].

Also, it should be stressed that a special case was implicitly considered in the previous theorem. This special case arises when all the possible events in a state of the plant are controllable. Then, disabling all the events can cause termination. In order to avoid this (as nonterminating generators are considered), there is always at least one event enabled (for details, an interested reader is referred to [18]).

When the conditions are satisfied, a solution to the PSCP exists. The probabilistic supervisor can then be computed by the fixpoint iteration algorithm as presented in [18], [19].

III. The metric for the closest approximation

In this section, the problem of optimal supervisory control in our probabilistic setting is described, related research done on pseudometrics on states of probabilistic systems is reviewed, and the chosen metric is presented.

A. Problem Formulation

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. We define the problem as follows.

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 the 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: a *pseudometric* on 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 behaviour is reduced to the maximal permissible legal nonprobabilistic behaviour of the plant under control. In other words, the (nonprobabilistic) language of the modified specification is the supremal controllable sublanguage of $L(G_r)$ with respect to G_p. Consequently, the probabilities of the specification are appropriately normalized as it makes sense to revise the specification so that probabilities of the events inadmissible for not satisfying the first criterion are redistributed over the admissible ones.

B. Probabilistic pseudometrics

Probabilistic bisimulation is commonly used to define an equivalence relation between probabilistic systems. However, probabilistic bisimulation is not a robust relation: roughly speaking, two states of probabilistic systems are bisimilar if and only if they have the same transitions with exactly the same probabilities to states in the same equivalence classes.

As a more robust way to compare probabilistic systems, a notion of *pseudometric* is introduced. A *pseudometric on a set of states* Q is a function $d : Q \times Q \to \mathbb{R}$ that defines a distance between two elements of Q, and satisfies the following conditions: $d(x, y) \geq 0$, $d(x, x) = 0$, $d(x, y) = d(y, x)$, and $d(x, z) \leq d(x, y) + d(y, z)$, for any $x, y, z \in Q$. If all distances are in $[0, 1]$, the pseudometric is *1-bounded*. In the sequel, we will use the terms metric and pseudometric interchangeably.

Little work on metrics has focused on generative models. The first paper that discussed the use of a metric as a way to measure the distance between two probabilistic processes is [22]. This early work considers deterministic generative probabilistic systems. The distance between processes is a number between 0 and 1, and represents a measure of a behavioural proximity between the processes: the smaller the number, the smaller the distance. The work of [5] suggests a metric based on probabilities of occurrence of strings in languages generated by two automata. More precisely, the distance between two automata in the metric is defined as a maximal difference in occurrence probabilities of strings in the corresponding languages. Probabilistic generators are used to model probabilistic systems in [23]. In a symbolic pattern recognition application, a metric is introduced to measure the distance between the original model and the transformed one, where the transformed model has the same long term distribution over the states as the original one. The work of [20] 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 [24] (also known as the Hutchinson metric). The metric is characterized as the greatest fixed point of a function. Two states are at distance 0 in this metric if and only if they are probabilistic bisimilar. This is the metric we will use in the solution of our problem. The metric intuitively matches our notion of the distance between PDESs and accounts for all differences between corresponding transition probabilities, as opposed to e.g., that of [22] that, roughly speaking, considers only the maximum of the differences between the corresponding probabilities. Furthermore, as the metric is suggested for a large class of systems, it allows for an extension of our work to e.g., nondeterministic systems. The metric discounts the future. The concept of discount has been widely applied in game theory, economics and optimal control. From an engineering point of view, one cares more about an error in the near future than the one in the distant future [25]. Also, there is a simple algorithm to compute distances in this metric for our generative, deterministic model (see Section IV). Further, as presented in [26], the metric has both logical and trace characterization. The logical characterization measures the distance between two systems by a $[0, 1]$-valued formula that distinguishes between the systems the most, while the trace characterization describes the similarity between the probabilistic traces of similar systems. More precisely, the trace characterization shows that the metric measures not only the difference in (appropriately discounted) occurrence probabilities of strings in two systems, but

also differences in (appropriately discounted) occurrence probabilities of certain sets of strings as well as complicated properties of strings.

For reactive systems, the work of [20] is closely related to [27]–[30]. Compared to the classical reactive models of [27], [29], [30], a generative model is more general: for every state, it contains not only the information about relative probabilities of transitions on the same event, but also information on relative probabilities of transitions on different events [31], [32]. In the probabilistic model checking literature, the model of [30] is extended with a state labeling function that assigns to each state a set of atomic propositions valid in that state, i.e. *probabilistic transition systems* of [30] are extended to *(labeled) discrete time Markov chains* [33], [34]. Our generative model can be transformed to (labeled) discrete time Markov chain [33], [34], but with a state space expansion by a factor of $O(|\Sigma|)$. Also, with the same state expansion factor, our generators can be converted to the *labeled concurrent Markov chains* of [28] and *partial labeled Markov chains* of [27], [29] (standard Markov decision processes). However, as the current mathematical apparatus allows for direct reasoning about distance between our generators, no benefits in regards to the optimal supervisory control of PDESs would have been gained by a transformation to one of the aforementioned models. For a more detailed discussion of probabilistic metrics, see [35].

C. The metric

Let $G = (Q, \Sigma, \delta, q_0, p)$ be a PDES, where $Q = \{q_0, q_1, \dots q_{N-1}\}$. This is the system that will be used throughout the sequel. Our pseudometric is based on the metric suggested in [20] for a large class of automata which includes our generator: the two metrics are the same up to a constant.

First, [20] introduces the class $\mathcal{M}$ of 1-bounded pseudometrics on states with the (partial) ordering

$$d_1 \preceq d_2 \text{ if } \forall q_q, q_r \in Q \quad d_1(q_q, q_r) \geq d_2(q_q, q_r) \tag{2}$$

as was initially suggested in [28]. It is proved that $(\mathcal{M}, \preceq)$ is a complete lattice.

Then, 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 transitions further in the future is discounted: the smaller the value of e, the greater the discount on future transitions. Next, we introduce some

useful notation. Let $q_q, q_r \in Q$ and let ρ_{q_q} and ρ_{q_r} be the distributions on $\Sigma \times Q$ induced by the states q_q and q_r, respectively. Assume $0 \leq i, j \leq N-1$. For notational convenience, we will write $\rho_{\sigma,i}$ instead of $\rho_{q_q}(\sigma, q_i)$, and, similarly, $\rho'_{\sigma,j}$ instead of $\rho_{q_r}(\sigma, q_j)$. Without loss of generality, we assume that the total mass of ρ_{q_q} is greater than or equal to the total mass of ρ_{q_r}:

$$\sum_{\substack{\sigma\in\Sigma \\ 0\leq i\leq N-1}} \rho_{\sigma,i} \geq \sum_{\substack{\sigma\in\Sigma \\ 0\leq i\leq N-1}} \rho'_{\sigma,i}.$$

This assumption is not needed for nonterminating automata. Then, the distance between the distributions ρ_{q_q} and ρ_{q_r}, $d(\rho_{q_q}, \rho_{q_r})$ (note the slight abuse of notation) for our generators is given as:

$$\text{Maximize} \left(\sum_{\substack{\sigma\in\Sigma \\ 0\leq i\leq N-1}} a_{\sigma,i}\rho_{\sigma,i} \right) - \left(\sum_{\substack{\sigma\in\Sigma \\ 0\leq i\leq N-1}} a_{\sigma,i}\rho'_{\sigma,i} \right) \tag{3}$$

$$\text{subject to} \quad 0 \leq a_{\sigma,i} \leq 1, \qquad \sigma \in \Sigma, \;\; 0 \leq i \leq N-1$$

$$a_{\sigma,i} - a_{\alpha,j} \leq c_{ij}^{\sigma\alpha}, \quad \sigma, \alpha \in \Sigma, \;\; 0 \leq i, j \leq N-1$$

where

$$c_{ij}^{\sigma\alpha} = \begin{cases} e \cdot d(q_i, q_j) & \text{if } \sigma = \alpha \\ 1 & \text{otherwise} \end{cases}$$

If the total mass of ρ_{q_q} is strictly less than the total mass of ρ_{q_r}, $d(\rho_{q_q}, \rho_{q_r})$ is defined to be $d(\rho_{q_r}, \rho_{q_q})$.

The pseudometric on states, d_{fp}, is then given as the greatest fixed-point of the function $\mathcal{D}$ on $\mathcal{M}$ (here, for probabilistic generators, we give a simplified version of that in [20]):

$$\mathcal{D}(d)(q_q, q_r) = d(\rho_{q_q}, \rho_{q_r}), \quad d \in \mathcal{M}, q_q, q_r \in Q \tag{4}$$

The definition of the metric on distributions is a modified version of that of [20]: the metric is changed such that the distances between the states in d_{fp} are larger by a factor of $1/e$ than the distances in the metric defined in [20]. This is done so that the distances in our metric are in $[0, 1]$ interval instead of $[0, e]$. A number of existing results can be reused in reasoning about our metric. The distance between distributions (3) is a 1-bounded pseudometric, and is consistent with the ordering (2) (see [28], [30]). The proofs that the function defined by (4) is monotone on $\mathcal{M}$, and that it does have a greatest fixed point originate from [28]. Also, according to Tarski's

fixed point theorem, the greatest fixed point of function $\mathcal{D}$ 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 ([20], [28]) (equivalently, the closure ordinal of $\mathcal{D}$ is ω).

The work of [20] does not offer any algorithms for calculation of the distances in their metric. In the following section, two algorithms for calculating distances in our metric are proposed.

IV. Calculating the metric

As a prelude to the solution of the optimal approximation problem, two algorithms that calculate/approximate distances in the metric d_{fp} are suggested. First, the function $\mathcal{D}$ is simplified, and then the algorithms are described, and their correctness proven.

A. Simplifying function $\mathcal{D}$ for deterministic generators

The function that represents the pseudometric on distributions is defined as the linear programming problem (3). We now show that, for deterministic generators, this function, and consequently, function $\mathcal{D}$ as defined by (4), can be simplified by explicitly solving the linear programming problem (3).

First, recall that our generators are deterministic: for an event σ and a state q, there is at most one state q' such that $q' = \delta(q, \sigma)$. For the purposes of the following analysis of our deterministic generators, we rewrite the objective function of the optimization problem of (3) as:

$$\sum_{\sigma \in \Sigma} (a_{\sigma,i(q_q,\sigma)} \rho_{\sigma,i(q_q,\sigma)} - a_{\sigma,j(q_r,\sigma)} \rho'_{\sigma,j(q_r,\sigma)}) \tag{5}$$

where $i(q_q, \sigma) = i$ such that $q_i = \delta(q_q, \sigma)$ if $\delta(q_q, \sigma)!$, and $i(q_q, \sigma) = 0$, otherwise. We arbitrarily choose $i(q_q, \sigma)$ to be 0 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 $i(q_q, \sigma) \in \{1, \ldots, N-1\}$. 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 $j(q_r, \sigma)$.

We are now ready to state our first result.

Lemma 1: Let $G = (Q, \Sigma, \delta, q_0, p)$ be a PDES. Then, the function $\mathcal{D}$ simplifies to:

$$\mathcal{D}(d)(q_q, q_r) = \sum_{\sigma \in \Sigma} max(\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}, c_{ij}\rho_{\sigma,i}) \tag{6}$$

where, again, $c_{ij} = e \cdot d(q_i, q_j)$ as before, and i and j denote $i(q_q, \sigma)$ and $j(q_r, \sigma)$, respectively, as defined in (5).

Proof: The objective function (5) can be maximized by maximizing each of its summands separately. In order to explain this observation, we consider a summand $a_{\sigma,i}\rho_{\sigma,i} - a_{\sigma,j}\rho'_{\sigma,j}$. Due to the generator's determinism, there is no other nonzero summand containing $a_{\sigma,k}$, $0 \leq k \leq N-1$, $k \neq i$, $k \neq j$. Therefore, the last constraint of (3) for any two coefficients $a_{\sigma,i}$ and $a_{\alpha,j}$ ($0 \leq i, j \leq N-1$) from different summands becomes $a_{\sigma,i} - a_{\alpha,j} \leq 1$. This constraint is already implied by the first constraint, so we can independently pick the coefficients a in different summands, and, consequently, independently maximize the summands in order to maximize the sum.

In order to maximize a summand of the objective function (5), we solve the following linear programming problem for $\sigma \in \Sigma$:

$$\text{Maximize } (a_{\sigma,i}\rho_{\sigma,i} - a_{\sigma,j}\rho'_{\sigma,j})$$

$$\text{subject to } 0 \leq a_{\sigma,i}, a_{\sigma,j} \leq 1,$$

$$a_{\sigma,i} - a_{\sigma,j} \leq c_{ij}$$

where i and j are defined as in (5), and $c_{ij} = ed(q_i, q_j)$ as before. Also, note that the set of constraints does not contain the inequality $a_{\sigma,j} - a_{\sigma,i} \leq c_{ji}$. In order to maximize the given function, the coefficient $a_{\sigma,i}$ is to be chosen to be greater than $a_{\sigma,j}$ since the given constraints allow it. In that case, since $c_{ij} = c_{ji}$, if $a_{\sigma,i} - a_{\sigma,j} \leq c_{ij}$, then $a_{\sigma,j} - a_{\sigma,i} \leq c_{ji}$ follows, so the latter constraint is redundant. Further, it is not hard to see that the solution of the given linear programming problem for $\rho_{\sigma,i} \geq \rho'_{\sigma,j}$ is equal to $\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}$. We can solve this problem using graphical method, simplex method or using the following line of reasoning. In order to maximize the given function, we can either choose $a_{\sigma,i}$ to be 1 and then pick $a_{\sigma,j}$ so that it has the minimal value for the given constraints, or we choose $a_{\sigma,j}$ to be 0, and then pick $a_{\sigma,i}$ so that it has the maximal value under the given constraints. In the first case, we pick $a_{\sigma,i}$ to be 1, $a_{\sigma,j}$ to be $1 - c_{ij}$, and value of the objective function is $\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}$. In the second case, since $a_{\sigma,j}$ is 0, then $a_{\sigma,i}$ is equal to c_{ij}, and the objective function becomes $c_{ij}\rho_{\sigma,i}$. The latter is our solution, since $c_{ij}\rho_{\sigma,i} = c_{ij}\left(\rho_{\sigma,i} - \rho'_{\sigma,j} + \rho'_{\sigma,j}\right) \leq \rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}$ (for $\rho_{\sigma,i} \geq \rho'_{\sigma,j}$ and $c_{ij} \in [0, 1]$). Using the same reasoning, for $\rho_{\sigma,i} < \rho'_{\sigma,j}$, the maximum is reached at $(a_i, a_j) = (c_{ij}, 0)$ and its value is $c_{ij}\rho_{\sigma,i}$.

Now, we put together the presented solution of the linear programming problem (3). The distance between the distributions ρ_{q_q} and ρ_{q_r} is then:

$$d(\rho_{q_q}, \rho_{q_r}) = \sum_{\sigma \in \Sigma} f(d, q_q, q_r, \sigma), \text{ where} \tag{7}$$

$$f(d, q_q, q_r, \sigma) = \begin{cases} \rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j} & \text{if } \rho_{\sigma,i} \geq \rho'_{\sigma,j} \\ c_{ij}\rho_{\sigma,i} & \text{otherwise} \end{cases}$$

$$or, equivalently,$$

$$f(d, q_q, q_r, \sigma) = max(\rho_{\sigma,i} - \rho'_{\sigma,j} + c_{ij}\rho'_{\sigma,j}, c_{ij}\rho_{\sigma,i}),$$

where $c_{ij} = e \cdot d(q_i, q_j)$ as before, and i and j denote $i(q_q, \sigma)$ and $j(q_r, \sigma)$, respectively, as defined as in (5).

To summarize, the function $\mathcal{D}(d)$ for our model is given as:

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

where, again, $c_{ij} = e \cdot d(q_i, q_j)$ as before, and i and j denote $i(q_q, \sigma)$ and $j(q_r, \sigma)$, respectively, as defined in (5). ■

As stated in Section III-C, the pseudometric d_{fp} is now characterized as the greatest fixed point of function $\mathcal{D}$.

B. Calculating the Pseudometric

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

First, some useful definitions and results from linear algebra are introduced.

A real $n \times n$ matrix $A = (a_{ij})$ defines a linear mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$, and we will write $A \in L(\mathbb{R}^n)$ to denote either the matrix or linear function, as no distinction between the two will be made. Also, the absolute value of column vector $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ will be denoted by $|x|$, and defined as $|x| = (|x_1|, \ldots, |x_n|)^T$. A partial ordering on $\mathbb{R}^n$ is defined in a natural way:

$$\forall x, y \in \mathbb{R}^n \;\; x \leq y \Leftrightarrow (\forall i = 1, \ldots, n \;\; x_i \leq y_i)$$

Definition 2: For any complex $n \times n$ matrix A, the spectral radius of A is defined as the maximum of $|\lambda_1|, \cdots, |\lambda_n|$, where $\lambda_1, \cdots, \lambda_n$ are the eigenvalues of A.

The spectral radius of A, denoted $\varphi(A)$, satisfies $\varphi(A) \leq ||A||$, where $||A||$ is an arbitrary norm on $\mathbb{R}^n$. During the course of the following proof, we will make use of infinity norm $||A||_\infty = \underset{1 \leq i \leq n}{max} \sum_{j=1}^{n} |a_{ij}|$.

Definition 3 ([36]): An operator $G : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ is called a *P-contraction* on a set $D_0 \subseteq D$ if there exists a linear operator $P \in L(\mathbb{R}^n)$ such that $P \geq 0$, $\varphi(P) < 1$ and

$$|G(x) - G(y)| \leq P\,|x - y| \text{ for all } x, y \in D_0. \tag{8}$$

Now, let $d \in \mathcal{M}$. Next, we define the function $\mathcal{V} : \mathcal{M} \rightarrow [0,1]^{N^2}$:

$$\mathcal{V}(d) = (d(q_0, q_0), d(q_0, q_1), \ldots, d(q_{N-1}, q_{N-2}), d(q_{N-1}, q_{N-1}))^T.$$

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

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

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

$$\mathcal{D}_k(\mathcal{V}(d)) = d(\rho_{q_i}, \rho_{q_j}),\ \ i = k\ div\ (N+1), j = (k-1)\ mod\ N. \tag{9}$$

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

Lemma 2: The function $\mathcal{D}$ is a P-contraction on D_0.

Proof: Let $d', d'' \in \mathcal{M}$, and $\mathfrak{d}' = \mathcal{V}(d')$, and $\mathfrak{d}'' = \mathcal{V}(d'')$. Let $k \in \{1, \ldots, N^2\}$, and let i and j ($0 \leq i, j \leq N-1$) be given as in (9). Also, let $t(i,\sigma) = t$ such that $\delta(q_i, \sigma) = q_t$ if $\delta(q_i, \sigma)!$, and $t(i,\sigma) = 0$, otherwise. Similarly, $l(j,\sigma) = l$ such that $\delta(q_j, \sigma) = q_l$ if $\delta(q_j, \sigma)!$, and $l(j,\sigma) = 0$, otherwise. Again, for notational convenience, we will write t instead of $t(i,\sigma)$, and l instead of $l(j,\sigma)$. Also, we will write $\rho_{\sigma,t}$ instead of $\rho_{q_i}(\sigma, q_t)$, and, similarly, $\rho'_{\sigma,l}$ instead of

$\rho_{q_j}(\sigma, q_l)$ for $q_t, q_l \in Q$. Then:

$$|\mathcal{D}_k(\mathfrak{d}') - \mathcal{D}_k(\mathfrak{d}'')| = \left|d'(\rho_{q_i}, \rho_{q_j}) - d''(\rho_{q_i}, \rho_{q_j})\right|$$
$$= |\sum_{\sigma\in\Sigma} max(\rho_{\sigma,t} - \rho'_{\sigma,l} + ed'(q_t, q_l)\rho'_{\sigma,l}, ed'(q_t, q_l)\rho_{\sigma,t})$$
$$- \sum_{\sigma\in\Sigma} max(\rho_{\sigma,t} - \rho'_{\sigma,l} + ed''(q_t, q_l)\rho'_{\sigma,l}, ed''(q_t, q_l)\rho_{\sigma,t})|$$

$$= |\sum_{\sigma\in\Sigma} (max(\rho_{\sigma,t} - \rho'_{\sigma,l} + ed'(q_t, q_l)\rho'_{\sigma,l}, ed'(q_t, q_l)\rho_{\sigma,t})$$
$$- max(\rho_{\sigma,t} - \rho'_{\sigma,l} + ed''(q_t, q_l)\rho'_{\sigma,l}, ed''(q_t, q_l)\rho_{\sigma,t}))|$$
$$= \left|\sum_{\sigma\in\Sigma} e(d'(q_t, q_l) - d''(q_t, q_l))min(\rho_{\sigma,t}, \rho'_{\sigma,l})\right|$$

$$\left(\begin{array}{l} \text{since, for any } d \in \mathcal{M}, \text{it holds that:} \\ \\ max(\rho_{\sigma,t} - \rho'_{\sigma,l} + ed(q_t, q_l)\rho'_{\sigma,l}, ed(q_t, q_l)\rho_{\sigma,t}) = \left\{ \begin{array}{cc} \rho_{\sigma,t} - \rho'_{\sigma,l} + ed(q_t, q_l)\rho'_{\sigma,l}, & \text{if } \rho_{\sigma,t} \geq \rho'_{\sigma,l} \\ ed(q_t, q_l)\rho_{\sigma,t}, & \text{otherwise} \end{array} \right. \end{array} \right)$$

$$\leq e\sum_{\sigma\in\Sigma} min(\rho_{\sigma,t}, \rho'_{\sigma,l})\, |d'(q_t, q_l) - d''(q_t, q_l)| \tag{10}$$

$$\leq e \sum_{\substack{\sigma\in\Sigma \\ m=tN+l+1}} min(\rho_{\sigma,t}, \rho'_{\sigma,l})\, |\mathfrak{d}'_m - \mathfrak{d}''_m|\,. \tag{11}$$

Note that $t = t(i, \sigma)$ and $l = l(j, \sigma)$ are also functions of k (since i and j are functions of k). Now, without the explicit construction of matrix P, we can see from (11) that there exists P such that $|\mathcal{D}(\mathfrak{d}') - \mathcal{D}(\mathfrak{d}'')| \leq P\,|\mathfrak{d}' - \mathfrak{d}''|$ where

$$||P||_\infty = \max_k \{e\sum_{\sigma\in\Sigma} min(\rho_{\sigma,t}, \rho'_{\sigma,l})\}$$
$$\leq e$$
$$\text{(since } \sum_{\substack{\sigma\in\Sigma \\ t\in\{0,\ldots,N-1\}}} \rho_{\sigma,t} = 1, \quad \sum_{\substack{\sigma\in\Sigma \\ l\in\{0,\ldots,N-1\}}} \rho'_{\sigma,l} = 1)$$
$$< 1 \text{ (since } e \in (0, 1))$$

Therefore, $\varphi(P) < 1$ and, since, obviously, $P \geq 0$, then $\mathcal{D}$ is P-contraction. ■

Lemma 3: Let $d', d'' \in \mathcal{M}$, and $\mathfrak{d}' = \mathcal{V}(d')$, and $\mathfrak{d}'' = \mathcal{V}(d'')$. For any $k \in \{1, \ldots, N^2\}$, there exists $m \in \{1, \ldots, N^2\}$ such that:

$$|\mathcal{D}_k(\mathfrak{d}') - \mathcal{D}_k(\mathfrak{d}'')| \leq e\,|\mathfrak{d}'_m - \mathfrak{d}''_m|$$

Proof: See the Appendix. ■

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

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

converges to the unique fixed point of $\mathcal{D}$ in D_0, $\mathfrak{d}^*$, and the error of convergence is given componentwise ($k \in \{1, \ldots, N^2\}$) as:

$$|\mathfrak{d}_k^n - \mathfrak{d}_k^*| \leq (1-e)^{-1} e^n, \;\; n = 1, 2, \ldots \tag{12}$$

Proof: Note that this is a variant of the contraction-mapping theorem extended to P-contractions ([36], Theorem 13.1.2.). A similar proof technique is employed.

Let $n, m \geq 1$. Then:

$$\begin{aligned}
&|\mathfrak{d}_k^{n+m} - \mathfrak{d}_k^n| \qquad (13)\\
&\leq \sum_{t=1}^{m} |\mathfrak{d}_k^{n+t} - \mathfrak{d}_k^{n+t-1}| \\
&\leq \sum_{t=1}^{m} e^t |\mathfrak{d}_{i(t)}^n - \mathfrak{d}_{i(t)}^{n-1}| \\
&\text{(applying Lemma 3 } t \text{ times, where } i(t) \in \{1, \ldots, N^2\}) \\
&\leq \sum_{t=1}^{m} e^t \underset{i(t)}{max} \{|\mathfrak{d}_{i(t)}^n - \mathfrak{d}_{i(t)}^{n-1}|\} \\
&\leq \left(\sum_{t=1}^{m} e^t \right) |\mathfrak{d}_j^n - \mathfrak{d}_j^{n-1}| \text{ (for some } j \in \{1, \ldots, N^2\}) \\
&\leq (1-e)^{-1} e |\mathfrak{d}_j^n - \mathfrak{d}_j^{n-1}| \text{ (since } \sum_{t=0}^{m} e^t \leq (1-e)^{-1} \text{ for } m \geq 0) \\
&\leq (1-e)^{-1} e^n |\mathfrak{d}_l^1 - \mathfrak{d}_l^0| \quad \text{(for some } l \in \{1, \ldots, N^2\}, \text{ using Lemma 3 } (n-1) \text{ times)} \\
&\leq (1-e)^{-1} e^n \qquad (14)
\end{aligned}$$

Therefore, the sequence $\{\mathfrak{d}_k^n\}_{n \geq 0}$ is a Cauchy sequence and hence converges to some $\mathfrak{d}_k^*$, and, consequently, the sequence $\{\mathfrak{d}^n\}_{n \geq 0}$ converges to some $\mathfrak{d}^* \in D_0$. Also, we have:

$$|\mathfrak{d}^* - \mathcal{D}(\mathfrak{d}^*)| \leq |\mathfrak{d}^* - \mathfrak{d}^{n+1}| + |\mathcal{D}(\mathfrak{d}^n) - \mathcal{D}(\mathfrak{d}^*)| \leq |\mathfrak{d}^* - \mathfrak{d}^{n+1}| + P|\mathfrak{d}^n - \mathfrak{d}^*|$$

When we let $n \to \infty$, we see that $\mathfrak{d}^* = \mathcal{D}(\mathfrak{d}^*)$. Also, the componentwise error estimate of (12) follows from (14) when $m \to \infty$ in (13).

Finally, it should be proven that $\mathfrak{d}^*$ is the only fixed point in D_0. Assume that there is another fixed point of $\mathcal{D}$ in the same set D_0, $\mathfrak{d}^+$. Then,

$$|\mathfrak{d}^* - \mathfrak{d}^+| = |\mathcal{D}(\mathfrak{d}^*) - \mathcal{D}(\mathfrak{d}^+)| \leq P|\mathfrak{d}^* - \mathfrak{d}^+|$$

Hence, $(I - P)|\mathfrak{d}^* - \mathfrak{d}^+| \leq 0$. However, since $\varphi(P) < 1$, $(I - P)^{-1} = \sum_{i=0}^{\infty} P^i \geq 0$ (see [36], 2.4.5.), then $|\mathfrak{d}^* - \mathfrak{d}^+| \leq 0$. Therefore, $\mathfrak{d}^* = \mathfrak{d}^+$. ■

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

Algorithm 1 Theorem 2 proves that the system of equations

$$\mathfrak{d} = \mathcal{D}(\mathfrak{d}) \tag{15}$$

has a unique solution. The system (15) is a system of linear equations. Therefore, the system (15) can be rewritten into the standard form $A\mathfrak{d} = b$, where A is a $N^2 \times N^2$ matrix and b is a column vector of dimension N^2. Therefore, the distances in the metric d_{fp} can be calculated by solving this system of linear equations. The distances found are exact solutions (if the round-off error is disregarded).

Algorithm 2 Theorem 2 also suggests an iterative algorithm to calculate distances in the metric d_{fp} between the states of a probabilistic generator. The algorithm turns out to be a straightforward modification of that of [30] that calculates distances in a pseudometric suggested for a different kind of probabilistic system, derived by using terminal coalgebras. Let $d^0(q_q, q_r) = 0$ for any two states $q_q, q_r \in Q$. As before, let ρ_{q_q} and ρ_{q_r} be the distributions induced by the states q_q and q_r, respectively. The n-th iteration of the algorithm calculates the distance d^n between each two states $q_q, q_r \in Q$:

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

where $c_{ij} = e \cdot d^{n-1}(q_i, q_j)$, and $i = i(q_q, \sigma)$ and $j = j(q_r, \sigma)$ are defined as in (5). The accuracy of the solution found at the n-th iteration is $(1 - e)^{-1}e^n$.

The iterative method can be useful for systems with large N^2, where the direct method can be rather expensive. Furthermore, the mathematical apparatus used to reach the iterative method will be reused in the solution of the closest approximation problem in Section V.

An important feature of d_{fp} is to be noted: metric d_{fp} is defined on any two states of a single PDES, not on two states that belong to different PDESs. In order to define the distance between

two PDESs (with disjoint sets of states) as the distance between their initial states, a new PDES is created that represents the union of the two PDESs (the union is defined in a natural way as will be presented formally in Section V-B).

Also, it should be stressed that the presented algorithm works for $e \in (0, 1)$. However, [37] presented an algorithm for calculating distances in the pseudometric of [29] for a variant of Markov chains for the case when $e = 1$. The key element in the algorithm is Tarski's decision procedure for the first order theory of real closed fields. We believe that this algorithm can be modified to calculate d_{fp} between the states of probabilistic generators. However, as this algorithm is impractical, an efficient calculation of distances in the metric for $e = 1$ is still an open problem.

V. Closest approximation: algorithm

In this section the algorithm that solves the closest approximation problem is presented. All the results in the sequel are applicable for $e \in (0, 1)$.

First, the formulation of the closest approximation problem is repeated. Assume that the plant is given as PDES $G_p = (Q_p, \Sigma, \delta_p, q_{p_0}, p_p)$, and the requirements specification is given as $G_r = (Q_r, \Sigma, \delta_r, q_{r_0}, p_r)$. If there is no probabilistic supervisor V_p such that $L_p(V_p/G_p) = L_p(G_r)$, the optimal solution is sought. The solution is optimal in the following sense. It is assumed that the nonprobabilistic language of the requirement is a safety requirement: no other strings are allowed in the plant. Then, it is required that maximal permissible behaviour (in the nonprobabilistic sense) is achieved. In this case, the probabilistic behaviour of the controlled plant should be as close as possible to the requirements specification that is now normalized so that it is constrained to the supremal controllable nonprobabilistic language. The proposed algorithm uses this separation of probabilistic and nonprobabilistic aspects of optimality so that it deals with each aspect separately: the first part handles the "nonprobabilistic optimality", and the second part handles the "probabilistic optimality".

A. Algorithm: Part I

Before we start looking for the closest approximation in the sense of probability matching, we resort to the classical supervisory theory of supremal controllable languages. First, the classical controllability condition (i) of Theorem 1 is checked while constructing $L(G_p) \cap L(G_r)$. Then,

if the condition is 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 as only nonterminating PDESs are considered. 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 $G_{1a} = (T, \Sigma, \zeta, t_0)$ represent language K. We define a PDES $G_1 = (T, \Sigma, \zeta, t_0, p_1)$, where the distribution $p_1 : T \times \Sigma \to [0, 1]$, for any $q \in T, \sigma \in \Sigma$, is defined as:

$$p_1(q, \sigma) = \frac{p_p(q_p, \sigma)}{\sum\limits_{\sigma \in \{\sigma \in \Sigma | \zeta(q,\sigma)!\}} p_p(q_p, \sigma)}$$

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

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

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

where $q_r = \delta_r(q_{r_0}, s)$ 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 automaton representing language K is performed.

B. Algorithm: Part II

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 our chosen metric. 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' 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 might not be met. As our metric is defined on the states of a single system, in order to define distances between the states of different systems, namely G_2 and G_2', the union PDES $G_u = (Q \cup Q', \Sigma, \delta_u, q_0, p_u)$ is considered, where for $\sigma \in \Sigma$ and $q \in Q \cup Q'$:

$$\delta_u(q,\sigma) = \begin{cases} \delta(q,\sigma), & \text{if } q \in Q \\ \delta'(q,\sigma), & \text{otherwise} \end{cases} \qquad \text{and} \qquad p(q,\sigma) = \begin{cases} p(q,\sigma), & \text{if } q \in Q \\ p'(q,\sigma), & \text{otherwise.} \end{cases}$$

Generator G_u is merely a PDES consisting of the union of G_2 and G_2' with the initial state arbitrarily chosen (between q_0 and q_0') to be q_0.

Then, $\mathcal{M}$ is the set of 1-bounded pseudometrics on the states of this union system with the same ordering as in (2).

First, note that, considering the isomorphism between G_2^{np} and $G_2'^{np}$, only the distances between (probability measures on) states $q \in Q$ of G_2 and $q' = f(q) \in Q'$ of G_2' are of interest, where f is the isomorphism between G_2^{np} and $G_2'^{np}$. Also, let h be an isomorphism between G_2^{np} and G_1^{np}. It is assumed that PDES G_2 is in state q after the occurrence of string $s \in L(G_2)$ ($\delta(q_0, s) = q$). Then, the closest approximation G_2' is in state q', respectively, where $q' = f(q)$. 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'.

Next, a class $\mathcal{A}$ of partial functions $a : Q \times Q' \to [0,1]$ is defined, such that $\forall q \in Q, q' = f(q) \in Q'$ $a(q,q') = d(q,q')$, where $d \in \mathcal{M}$. Therefore, the class $\mathcal{A}$ is the class of all 1-bounded pseudometrics with domain reduced to $Q \times Q'$, and only distances between $q \in Q$ and $q' = f(q) \in Q'$ defined since the algorithm is independent of the distance between the other states. Next, we define a family Δ as a set of probability distributions on $\Sigma \times Q'$. Now, for each $\rho' \in Q' \to \Delta$, we define function $\mathcal{D}^{\rho'} : \mathcal{A} \to \mathcal{A}$ as ($q \in Q, q' = f(q) \in Q', d \in \mathcal{A}$):

$$\mathcal{D}^{\rho'}(d)(q,q') = d(\rho_q, \rho'_{q'}) \text{ and } \rho'(q') = \rho'_{q'},$$

where, as before, d is lifted to the metric on distributions, and $d(\rho_q, \rho'_{q'})$ is defined as in (7). Also, the reversed ordering on $\mathcal{A}$ is introduced to match the one in (2):

$$d_1 \preceq' d_2 \text{ if } \forall q \in Q,\ q' = f(q),\ d_1(q,q') \geq d_2(q,q').$$

The fact that $(\mathcal{A}, \preceq')$ is a complete lattice follows from the fact that $(\mathcal{M}, \preceq)$ is a complete lattice. Further, for each $\rho' \in Q' \to \Delta$, we define function $d_{fp}^{\rho'}$ as the greatest fixed point of function $\mathcal{D}^{\rho'}$. The problem of finding the optimal approximation reduces now to finding $\rho'_m \in Q' \to \Delta$ such that $d_{fp}^{\rho'_m}(q_0, q'_0) = \min_{\rho'}\{d_{fp}^{\rho'}(q_0, q'_0) | \rho' \in Q' \to \Delta\}$ and the conditions for the existence of a probabilistic supervisor of Theorem 1 are satisfied. It follows straight from the definitions of $\mathcal{D}^{\rho'}$ and $d_{fp}^{\rho'}$ that, for any $\rho' \in Q' \to \Delta$, $q \in Q, q' = f(q) \in Q'$, the distances $d_{fp}^{\rho'}(q, q')$ are distances in our pseudometric.

We assume that $T = \{t_0, t_1, \ldots t_{N-1}\}$, $Q = \{q_0, q_1, \ldots q_{N-1}\}$, and $Q' = \{q'_0, q'_1, \ldots q'_{N-1}\}$, where $q'_i = f(q_i)$, $t_i = h(q_i)$, $i = 0, \ldots, N-1$. Note that, for probability distributions, a different notation will be used than the one used in the previous section. Let $d \in \mathcal{A}$, $0 \leq i \leq N-1$, $\Psi(q_i) = Pos(q_i)$, $\Psi_u(q_i) = Pos(q_i) \cap \Sigma_u$, and $\Psi_c(q_i) = Pos(q_i) \cap \Sigma_c$. 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 $\rho'_{q'_i}(\sigma, q'_k)$, $k = 0, 1, \ldots N-1$. Now, the function $\mathcal{P} : \mathcal{A} \to \mathcal{A}$ is defined as:

$$\mathcal{P}(d)(q_i, q'_i) = \underset{\rho'_{q_i,\sigma}}{\text{Minimize}} \sum_{\sigma \in \Psi(q_i)} max(\rho_{q_i,\sigma} - \rho'_{q'_i,\sigma} + c_j \rho'_{q'_i,\sigma}, c_j \rho_{q_i,\sigma}), \tag{16}$$

where $c_j = e \cdot d(q_j, q'_j)$ s.t. $q_j = \delta(q_i, \sigma)$

subject to

$$\frac{p_1(t_i, \sigma)}{\sum_{\alpha \in \Psi_u(q_i)} p_1(t_i, \alpha)} = \frac{\rho'_{q'_i,\sigma}}{\sum_{\alpha \in \Psi_u(q_i)} \rho'_{q'_i,\alpha}}, \qquad \sigma \in \Psi_u(q_i), \tag{17}$$

$$\frac{\sum_{\alpha \in \Psi_u(q_i)} p_1(t_i, \alpha)}{p_1(t_i, \sigma)} \rho'_{q'_i,\sigma} + \sum_{\alpha \in \Psi_c(q_i)} \rho'_{q_i,\alpha} \leq 1, \quad \sigma \in \Psi_c(q_i), \tag{18}$$

$$\sum_{\alpha \in \Psi(q_i)} \rho'_{q'_i,\alpha} = 1, \tag{19}$$

$$\rho'_{q'_i,\sigma} \geq 0, \qquad \sigma \in \Psi(q_i). \tag{20}$$

The constraints (17) and (18) represent the conditions for the existence of probabilistic supervisor given by Theorem 1. The function $\mathcal{P}$ is well-defined since, if $\rho'_{q'_i,\sigma} = p_1(t_i, \sigma)$ for all $\sigma \in \Sigma$, the constraints (17), (18), (19), (20) are satisfied. Therefore, the optimization problem has a feasible origin. Since $\mathcal{A}$ is a complete lattice, and the function $\mathcal{P}$ can be easily shown to be monotone, it has a greatest fixed point. Next, a useful lemma is stated.

Lemma 4: Let $(\mathcal{L}, \preceq)$ be a complete lattice, and let $f, g : \mathcal{L} \to \mathcal{L}$ be two monotone functions

such that $\forall x \in \mathcal{L} : g(x) \preceq f(x)$. Let $\mathit{gfp}(f)$ and $\mathit{gfp}(g)$ denote the greatest fixed point of functions f and g, respectively. Then, $\mathit{gfp}(g) \preceq \mathit{gfp}(f)$.

Proof: See the Appendix. ■

Obviously, because of the definition of function $\mathcal{P}$, for any function $\mathcal{D}^{\rho'}$, where $\rho' \in Q' \to \Delta$, it holds that $\forall d \in \mathcal{A}\ \mathcal{D}^{\rho'}(d) \preceq' \mathcal{P}(d)$. Using Lemma 4, we conclude that the greatest fixed point of $\mathcal{P}$ is greater than or equal to any $d_{fp}^{\rho'}$, $\rho' \in Q \to \Delta$. This greatest fixed point corresponds to the minimal distance between q_0 and q_0' because of the reversed ordering on $\mathcal{A}$. Therefore, the greatest fixed point of function $\mathcal{P}$ corresponds to the distances in our pseudometric where the distance between q_0 and q_0' is minimized under the conditions of Theorem 1 for the existence of a probabilistic supervisor. Consequently, the values of decision variables $\rho'_{q'}$ for $q' \in Q'$ when the greatest fixed point of $\mathcal{P}$ is reached correspond to the statewise probability distributions of the optimal approximation.

We suggest an iterative algorithm to calculate the minimum achievable distance (i.e. the only fixed point of the function $\mathcal{P}$) up to a desired accuracy and provide the probability distribution of the system's achievable behaviour when this distance is reached. The proof pattern used for the algorithm from Section IV-B is followed. However, as mentioned before, the only relevant distances are the ones between $q \in Q$ and $q' = f(q) \in Q'$.

Let $d \in \mathcal{A}$. Again, we assume that $Q = \{q_0, q_1, \ldots q_{N-1}\}$, and $Q' = \{q_0', q_1', \ldots q_{N-1}'\}$, where $q_i' = f(q_i)$, $i = 0, \ldots, N-1$. Further, let us define function $\hat{\mathcal{V}} : \mathcal{M} \to [0,1]^N$ as:

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

Therefore, $\hat{\mathcal{V}}(d) = (\hat{\mathcal{V}}_1(d), \ldots, \hat{\mathcal{V}}_N(d))^T$, where, for $k = 1, \ldots, N$:

$$\hat{\mathcal{V}}_k(d) = d(q_{k-1}, q_{k-1}'). \tag{21}$$

The function $\mathcal{P}$ is redefined in a natural way as $\mathcal{P}(\hat{\mathcal{V}}(d)) = (\mathcal{P}_1(\hat{\mathcal{V}}(d)), \ldots, \mathcal{P}_N(\hat{\mathcal{V}}(d)))^T$, where for any $k \in \{1, \ldots, N\}$:

$$\mathcal{P}_k(\hat{\mathcal{V}}(d)) = \mathcal{P}(d)(q_{k-1}, q_{k-1}'),$$

where $q_{k-1}' = f(q_{k-1})$. Also, let $P_0 = \{\hat{\mathcal{V}}(d) | d \in \mathcal{A}\}$.

Theorem 3: Function $\mathcal{P}$ is P-contractive on P_0.

Proof: Let $d', d'' \in \mathcal{A}$, and $\hat{\mathfrak{d}}' = \hat{\mathcal{V}}(d')$, and $\hat{\mathfrak{d}}'' = \hat{\mathcal{V}}(d'')$. Next, for $q \in Q$, $q' = f(q) \in Q'$, we define set $\Phi(q)$ to be the set of all distributions $\rho'_{q'}$ that satisfy conditions given by (17), (18), (19),

and (20). Let $k \in \{1, \ldots, N\}$. Then, $\mathcal{P}_k(\hat{\mathfrak{d}}') = \mathcal{P}(d')(q_{k-1}, q'_{k-1})$, and $\mathcal{P}_k(\hat{\mathfrak{d}}'') = \mathcal{P}(d'')(q_{k-1}, q'_{k-1})$. Assume that the minumum of the objective function in (16) in function $\mathcal{P}(d')(q_{k-1}, q'_{k-1})$ is reached for $\rho'_{q'} = \mu$ for $\mu \in \Phi(q)$. Further, assume that the minumum of the objective function in (16) in function $\mathcal{P}(d'')(q_{k-1}, q'_{k-1})$ is reached for $\rho'_{q'} = \nu$ for $\nu \in \Phi(q)$. Let $\Psi = \Psi(q_{k-1})$. Also, let $j(k, \sigma) = j$ such that $q_j = \delta(q_{k-1}, \sigma)$, and $f(q_j) = q'_j$. Assume that $\mathcal{P}_k(\hat{\mathfrak{d}}') \geq \mathcal{P}_k(\hat{\mathfrak{d}}'')$. Then:

$$
\begin{aligned}
&\left|\mathcal{P}_k(\hat{\mathfrak{d}}') - \mathcal{P}_k(\hat{\mathfrak{d}}'')\right| \\
&= |\sum_{\sigma\in\Psi} max(\rho_{q_{k-1},\sigma} - \mu_{q'_{k-1},\sigma} + ed'(q_j, q'_j)\mu_{q'_{k-1},\sigma}, ed'(q_j, q'_j)\rho_{q_{k-1},\sigma}) \\
&- \sum_{\sigma\in\Psi} max(\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed''(q_j, q'_j)\nu_{q'_{k-1},\sigma}, ed''(q_j, q'_j)\rho_{q_{k-1},\sigma})| \\
&\leq |\sum_{\sigma\in\Psi} max(\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed'(q_j, q'_j)\nu_{q'_{k-1},\sigma}, ed'(q_j, q'_j)\rho_{q_{k-1},\sigma}) \\
&- \sum_{\sigma\in\Psi} max(\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed''(q_j, q'_j)\nu_{q'_{k-1},\sigma}, ed''(q_j, q'_j)\rho_{q_{k-1},\sigma})| \qquad (22) \\
&\text{(for } \rho'_{q'_{k-1},\sigma} = \mu_{q'_{k-1},\sigma} \text{ the minimum in } \mathcal{P}_k(\hat{\mathfrak{d}}') \text{ is reached)} \\
&\leq |\sum_{\sigma\in\Psi} (max(\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed'(q_j, q'_j)\nu_{q'_{k-1},\sigma}, ed'(q_j, q'_j)\rho_{q_{k-1},\sigma}) \\
&- max(\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed''(q_j, q'_j)\nu_{q'_{k-1},\sigma}, ed''(q_j, q'_j)\rho_{q_{k-1},\sigma}))| \\
&\leq \sum_{\sigma\in\Psi} |max(\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed'(q_j, q'_j)\nu_{q'_{k-1},\sigma}, ed'(q_j, q'_j)\rho_{q_{k-1},\sigma}) \\
&- max(\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed''(q_j, q'_j)\nu_{q'_{k-1},\sigma}, ed''(q_j, q'_j)\rho_{q_{k-1},\sigma})| \qquad (23)
\end{aligned}
$$

(Similarly, when $\mathcal{P}_k(\hat{\mathfrak{d}}') \leq \mathcal{P}_k(\hat{\mathfrak{d}}'')$, we get (23), with $\mu_{q'_{k-1},\sigma}$ instead of $\nu_{q'_{k-1},\sigma}$.) Every summand in (23) has one of the following forms:

$$
\left|\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed'(q_j, q'_j)\nu_{q'_{k-1},\sigma} - (\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed''(q_j, q'_j)\nu_{q'_{k-1},\sigma})\right| \text{ or}
$$

$$
\left|ed'(q_j, q'_j)\rho_{q_{k-1},\sigma} - ed''(q_j, q'_j)\rho_{q_{k-1},\sigma})\right|, \text{ where}
$$

$$
\begin{aligned}
&\left|\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed'(q_j, q'_j)\nu_{q'_{k-1},\sigma} - (\rho_{q_{k-1},\sigma} - \nu_{q'_{k-1},\sigma} + ed''(q_j, q'_j)\nu_{q'_{k-1},\sigma})\right| \\
&= e\nu_{q'_{k-1},\sigma}\left|d'(q_j, q'_j) - d''(q_j, q'_j)\right| \leq e\rho_{q_{k-1},\sigma}\left|d'(q_j, q'_j) - d''(q_j, q'_j)\right|
\end{aligned}
$$

and

$$
\left|ed'(q_j, q'_j)\rho_{q_{k-1},\sigma} - ed''(q_j, q'_j)\rho_{q_{k-1},\sigma})\right| = e\rho_{q_{k-1},\sigma}\left|d'(q_j, q'_j) - d''(q_j, q'_j)\right|
$$

Hence,

$$\left|\mathcal{P}_k(\hat{\mathfrak{d}}') - \mathcal{P}_k(\hat{\mathfrak{d}}'')\right| \leq \sum_{\sigma\in\Psi} e\rho_{q_{k-1},\sigma}\left|d'(q_j, q'_j) - d''(q_j, q'_j)\right|$$

Further, using the same reasoning as in the proof of Lemma 2, it is straightforward to show that $\mathcal{P}$ is P-contractive. ■

Lemma 5: Let $d', d'' \in \mathcal{A}$, and $\hat{\mathfrak{d}}' = \hat{\mathcal{V}}(d')$, and $\hat{\mathfrak{d}}'' = \hat{\mathcal{V}}(d'')$. For any $k \in \{1, \ldots, N\}$, there exists $m \in \{1, \ldots, N\}$ such that:

$$\left|\mathcal{P}_k(\hat{\mathfrak{d}}') - \mathcal{P}_k(\hat{\mathfrak{d}}'')\right| \leq e\left|\hat{\mathfrak{d}}'_m - \hat{\mathfrak{d}}''_m\right|$$

Proof: See the Appendix. ■

Theorem 4: For any $\hat{\mathfrak{d}}^0 \in P_0$, the sequence

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

converges to the only fixed point of $\mathcal{P}$ in P_0, $\hat{\mathfrak{d}}^*$, and the error of convergence is given componentwise ($k \in \{1, \ldots, N\}$) as:

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

Proof: See the Appendix. ■

The problem of (16 – 20) is not a linear programming problem, but it is transformable into one by using additional variables $y_{q_i,\sigma}$, and by transforming (17) into (24):

$$\begin{aligned}
&\text{Minimize} \sum_{\sigma\in\Psi(q_i)} y_{q_i,\sigma} \\
&\text{subject to} \\
&\rho_{q_i,\sigma} - \rho'_{q'_i,\sigma} + c_j\rho'_{q'_i,\sigma} \leq y_{q_i,\sigma}, \qquad \sigma \in \Psi(q_i), \\
&c_j\rho_{q_i,\sigma} \leq y_{q_i,\sigma}, \qquad \sigma \in \Psi(q_i), \\
&\text{where } c_j = e \cdot d(q_i, q'_i) \text{ s.t. } q_j = \delta(q_i, \sigma), \\
&p_1(t_i,\sigma)\sum_{\alpha\in\Psi_u(q_i)} \rho'_{q_i,\alpha}, = \rho'_{q'_i,\sigma}\sum_{\alpha\in\Psi_u(q_i)} p_1(t_i,\alpha), \quad \sigma \in \Psi_u(q_i), \\
&\frac{\sum_{\alpha\in\Psi_u} p_1(t_i,\alpha)}{p_1(t_i,\sigma)}\rho'_{q'_i,\sigma} + \sum_{\alpha\in\Psi_c(q_i)} \rho'_{q'_i,\alpha} \leq 1, \quad \sigma \in \Psi_c(q_i), \\
&\sum_{\alpha\in\Psi(q_i)} \rho'_{q'_i,\alpha} = 1, \\
&\rho'_{q'_i,\sigma} \geq 0, \qquad \sigma \in \Psi(q_i).
\end{aligned} \tag{24}$$

It might look as if (24) is weaker than (17) as it allows the possibility of $\rho'_{q'_i,\sigma} = 0$ for all $\sigma \in \Psi_u(q_i)$, which (17) forbids. However, this is not the case. Let $\rho'_{q'_i,\sigma} = 0$ for every $\sigma \in \Psi_u(q_i)$. From (18) it follows that:

$$\frac{\sum_{\alpha\in\Psi_u} p_1(t_i,\alpha)}{p_1(t_i,\sigma)}\rho'_{q'_i,\sigma} \leq \sum_{\alpha\in\Psi_u(q_i)} \rho'_{q'_i,\alpha} = 0$$

which would mean that $\rho'_{q'_i,\sigma} = 0$ for every $\sigma \in \Psi_c(q_i)$ which contradicts the condition (19).

We now present the iterative algorithm for finding the fixed point of function $\mathcal{P}$.

Let $d^0(q_i, q'_i) = 0$, $i = 0, 1, ...N-1$. The distance $d^n(q_i, q'_i)$ in the n-th iteration ($n > 0$) is given as:

$$\text{Minimize} \sum_{\sigma\in\Psi(q_i)} y_{q_i,\sigma} \tag{25}$$

subject to

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

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

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

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

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

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

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

After the n-th iteration, the value 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 $(1-e)^{-1}e^n$ of the minimal achievable distance between the two systems (in our pseudometric). Note that the aforementioned results hold for $e \in (0, 1)$.

Also, it should be noted that the presented algorithm can be modified for the case when the requirements specification is not revised (see [26]). More precisely, a modification of the presented algorithm can be used to solve the control problem presented in Section III-A with requirement 2) changed so that the distance between the controlled plant and the unmodified requirement is minimized.

C. Summarizing the algorithm

We now summarize the presented algorithm and give a brief complexity analysis.

1) First, the classical algorithm for finding the supremal controllable sublanguage is modified. The automaton G_s, the synchronous product of the nonprobabilistic automata underlying G_p and G_r is constructed. While constructing the product, the classical controllability conditions are checked for each state. If the conditions are satisfied for each state of the product, then $G = G_s$, and go to 2). If there is at least one state of the product for which the classical conditions do not hold, the rest of the algorithm for finding the automaton representing the supremal controllable sublanguage is then applied. The algorithm can easily be modified to exclude deadlock states: these states are considered uncontrollable. Let (reachable and deadlock-free) DES $G = (Q, \Sigma, \delta, q_0)$ represent this supremal controllable language.

2) Let G_1, G_2, and G_2' be defined as previously in this section. Check the equalities and inequalities of Theorem 1 for each state: if they are satisfied, a supervisor exists, and G_2 is the optimal approximation. If not, then let $d^0(q_i, q_i') = 0$ for all $0 \leq i \leq N-1$. The distance $d^n(q_i, q_i')$ in the n-th iteration ($n > 0$) is given by (25).

For each of the states of G_2 (typically, the number of states of G_2 is much smaller than $|Q_p| \cdot |Q_r|$), either the simplex method or an interior point method can be used to solve the linear programming problem (25). Depending on what method is used, the running time of the algorithm is either exponential (simplex method) or polynomial (interior point methods) in the maximal number of events possible from a state of the supremal controllable sublanguage of the specification (with respect to the plant). Even the worst-case exponential complexity of the simplex method is not problematic for two reasons: first, the method is very efficient in practice, and second, the number of possible events from a state is small in practical applications. Furthermore, the number of iterations needed to reach the accuracy of ϵ is $\lceil (\log_e \epsilon + \log_e (1-e)) \rceil$. This term is obtained from the fact that the number of iterations n for which an accuracy ϵ is achieved should be the smallest natural number for which $\epsilon \geq (1-e)^{-1} e^n$ is satisfied.

D. Example

For plant G_p depicted in Fig. 1, there does not exist a probabilistic supervisor V_p such that $G(V_p/G_p) = G_r$. Fig. 2 shows the modified plant G_1 and modified specification G_2, defined as suggested in Section V-A. For PDES G_2, let ρ_q be the probability distribution induced by the

state $q \in Q$ and, for PDES G'_2, let $\rho'_{q'}$ be the probability distribution induced by the state $q' \in Q'$. As before, we will write $\rho_{q,\sigma}$ instead of $\rho_q(\sigma, q_i)$, and $\rho'_{q',\sigma}$ instead of $\rho'_{q'}(\sigma, q'_j)$, $i, j = 0, 1, 2$.

At the n-th iteration, distances $d^n(q_0, q'_0), d^n(q_1, q'_1)$, and $d^n(q_2, q'_2)$ are calculated as follows:

$d^n(q_0, q'_0) =$ Minimize $(y_{q_0,\alpha} + y_{q_0,\beta})$

subject to

$$\rho_{q_0,\alpha} - \rho'_{q'_0,\alpha} + e \cdot d^{n-1}(q_1, q'_1)\rho'_{q'_0,\alpha} \leq y_{q_0,\alpha}, \quad e \cdot d^{n-1}(q_1, q'_1)\rho_{q_0,\alpha} \leq y_{q_0,\alpha},$$

$$\rho_{q_0,\beta} - \rho'_{q'_0,\beta} + e \cdot d^{n-1}(q_2, q'_2)\rho'_{q'_0,\beta} \leq y_{q_0,\beta}, \quad e \cdot d^{n-1}(q_2, q'_2)\rho_{q_0,\beta} \leq y_{q_0,\beta},$$

$$\frac{p(q_0,\beta)}{p(q_0,\alpha)}\rho'_{q_0,\alpha} + \rho'_{q_0,\alpha} \leq 1, \quad \rho'_{q'_0,\alpha} + \rho'_{q'_0,\beta} = 1, \quad \rho'_{q'_0,\alpha} \geq 0, \quad \rho'_{q'_0,\beta} \geq 0.$$

$d^n(q_1, q'_1) =$ Minimize $(y_{q_1,\beta} + y_{q_1,\gamma})$

subject to

$$\rho_{q_1,\beta} - \rho'_{q'_1,\beta} + e \cdot d^{n-1}(q_2, q'_2)\rho'_{q'_1,\beta} \leq y_{q_1,\beta}, \quad e \cdot d^{n-1}(q_2, q'_2)\rho_{q_1,\beta} \leq y_{q_1,\beta},$$

$$\rho_{q_1,\gamma} - \rho'_{q'_1,\gamma} + e \cdot d^{n-1}(q_0, q'_0)\rho'_{q'_1,\gamma} \leq y_{q_1,\gamma}, \quad e \cdot d^{n-1}(q_0, q'_0)\rho_{q_1,\gamma} \leq y_{q_1,\gamma},$$

$$\frac{p(q_1,\beta)}{p(q_1,\gamma)}\rho'_{q_0,\gamma} + \rho'_{q_1,\gamma} \leq 1, \quad \rho'_{q'_1,\beta} + \rho'_{q'_1,\gamma} = 1, \quad \rho'_{q'_1,\beta} \geq 0, \quad \rho'_{q'_1,\gamma} \geq 0.$$

$d^n(q_2, q'_2) =$ Minimize $(y_{q_2,\beta} + y_{q_2,\theta} + y_{q_2,\tau})$

subject to

$$\rho_{q_2,\beta} - \rho'_{q'_2,\beta} + e \cdot d^{n-1}(q_2, q'_2)\rho'_{q'_2,\beta} \leq y_{q_2,\beta}, \quad e \cdot d^{n-1}(q_2, q'_2)\rho_{q_2,\beta} \leq y_{q_2,\beta},$$

$$\rho_{q_2,\theta} - \rho'_{q'_2,\theta} + e \cdot d^{n-1}(q_0, q'_0)\rho'_{q'_2,\theta} \leq y_{q_2,\theta}, \quad e \cdot d^{n-1}(q_0, q'_0)\rho_{q_2,\theta} \leq y_{q_2,\theta},$$

$$\rho_{q_2,\tau} - \rho'_{q'_2,\tau} + e \cdot d^{n-1}(q_0, q'_0)\rho'_{q'_2,\tau} \leq y_{q_2,\tau}, \quad e \cdot d^{n-1}(q_0, q'_0)\rho_{q_2,\tau} \leq y_{q_2,\tau},$$

$$p(q_2,\tau)\left(\rho'_{q'_2,\tau} + \rho'_{q'_2,\beta}\right) = \rho'_{q'_2,\tau}\left(p(q_2,\tau) + p(q_2,\beta)\right), \quad \frac{p(q_2,\beta) + p(q_2,\tau)}{p(q_2,\theta)}\rho'_{q_2,\theta} + \rho'_{q_2,\theta} \leq 1,$$

$$\rho'_{q'_2,\beta} + \rho'_{q'_2,\theta} + \rho'_{q'_2,\tau} = 1, \quad \rho'_{q'_2,\beta} \geq 0, \quad \rho'_{q'_2,\theta} \geq 0, \quad \rho'_{q'_2,\tau} \geq 0.$$

Note that, for each $d^n(q_0, q'_0)$, and $d^n(q_1, q'_1)$, the equation that corresponds to the controllability condition for the sole uncontrollable event is missing, as it is trivially satisfied. Also, for $d^n(q_2, q'_2)$, the controllability equation was generated only for one of two uncontrollable events, as the two equations can be derived from each other.

For the accuracy $\epsilon = 0.001$, and $e = 0.5$, 11 iterations of the algorithm are needed. It is

found that the closest behaviour achievable with probabilistic control is as given in Fig. 2. It took 0.3 seconds on a 2.6GHz dual core Opteron processor with 8GB of RAM running Red Hat Enterprise Linux Server 5.5. In order to find the corresponding probabilistic supervisor, the algorithm of [18], [19] can be used. For the state q_0' and event α, the control input is 0.6, for q_1' and event γ, the input is 1, and for q_2' and event θ, the input is 0.625.

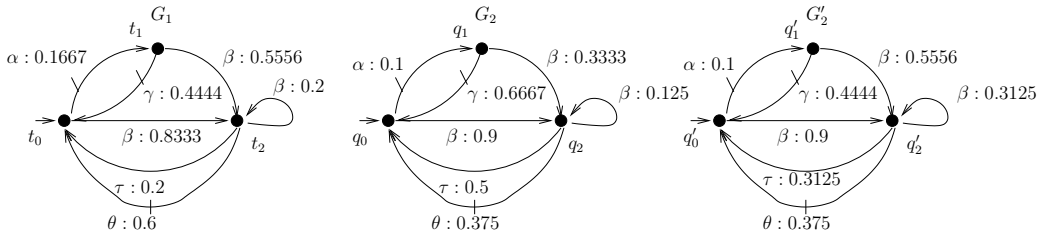

Fig. 2. Generators G_1, G_2, and optimal approximation G_2'

VI. CONCLUSIONS

This paper solves the classical problem of finding the closest approximation in the framework of probabilistic control of PDESs. Two algorithms for the calculation of distances (in the chosen pseudometric) between the states of a probabilistic generator used to model a PDES are suggested. Then, a modification of the iterative algorithm is proposed to minimize the distance (in this pseudometric) between the desired behaviour of the system and its achievable behaviour.

Although the rate of convergence of the algorithm to the minimal distance is known, we would like to investigate how unknown, desired probabilities of the closest approximation change as the distance converges. Experimental results indicate that the probabilities converge very quickly. Also, operators on probabilistic generators should be defined and their desired property of non-expansiveness with the respect to the metric should be checked. Non-expansiveness means that the distance between two systems does not increase when they are put in the same environment. This would allow for compositional reasoning. Further, the question of uniqueness of the closest approximation remains open as well as probabilistic control with marking.

REFERENCES

[1] P. Ramadge and W. Wonham, "On the supremal controllable sublanguage of a given language," *SIAM Journal on Control and Optimization*, vol. 25, no. 3, 1987.

[2] C. G. Cassandras, *Discrete Event Systems: Modeling and Performance Analysis*. Homewood, IL, USA: Richard D. Irwin, Inc., and Aksen Associates, Inc., 1993.

[3] M. O. Rabin, "Probabilistic automata," *Information and Control*, vol. 6, no. 3, pp. 230–245, September 1963.

[4] M. Molloy, "Performance analysis using stochastic Petri Nets," *IEEE Transactions on Computers*, vol. C-39, no. 9, pp. 913–917, 1982.

[5] V. Garg, R. Kumar, and S. Marcus, "Probabilistic language formalism for stochastic discrete event systems," *IEEE Transactions on Automatic Control*, vol. 44, no. 2, pp. 280–293, 1999.

[6] V. Garg, "An algebraic approach to modeling probabilistic discrete event systems," in *Proceedings of 31st IEEE Conference on Decision and Control*, Tucson, AZ, USA, Dec. 1992, pp. 2348–2353.

[7] ——, "Probabilistic languages for modeling of DEDS," in *Proceedings of 26th Conference on Information Sciences and Systems*, vol. 1, Princeton, NJ, Mar. 1992, pp. 198–203.

[8] H. Mortazavian, "Controlled stochastic languages," in *Proceedings of 31st Annual Allerton Conference on Communications, Control, and Computing*, Urbana, Illinois, 1993, pp. 938–947.

[9] V. S. Borkar, *Topics in controlled Markov chains*. New York: Wiley, 1991.

[10] R. Kumar and V. Garg, "Control of stochastic discrete event systems: Existence," in *Proceedings of 1998 International Workshop on Discrete Event Systems*, Cagliari, Italy, august 1998, pp. 24–29.

[11] M. Lawford and W. Wonham, "Supervisory control of probabilistic discrete event systems," in *Proceedings of the 36th IEEE Midwest Symposium on Circuits and Systems*, vol. 1. IEEE, Aug. 1993, pp. 327–331.

[12] R. Kumar and V. Garg, "Control of stochastic discrete event systems modeled by probabilistic languages," *IEEE Transactions on Automatic Control*, vol. 46, no. 4, pp. 593–606, Apr. 2001.

[13] I. Chattopadhyay and A. Ray, "Language-measure-theoretic optimal control of probabilistic finite state systems," in *Proceedings of 46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, Dec. 2007, pp. 5930–5935.

[14] ——, "Language-measure-theoretic optimal control of probabilistic finite-state systems," *International Journal of Control*, vol. 80, no. 8, pp. 1271–1290, 2007.

[15] Y. Li, F. Lin, and Z. H. Lin, "Supervisory control of probabilistic discrete event systems with recovery," *IEEE Trans. Autom. Control*, vol. 44, pp. 1971–1975, 1998.

[16] C. Baier, M. Größer, M. Leucker, B. Bollig, and F. Ciesinski, "Controller synthesis for probabilistic systems," in *Proceedings of the IFIP International Conference on Theoretical Computer Science*, J.-J. Lévy, E. W. Mayr, and J. C. Mitchell, Eds. Kluwer, 2004, pp. 493–506.

[17] A. Kučera and O. Stražovský, "On the controller synthesis for finite-state Markov decision processes," *Fundamenta Informaticae*, vol. 82, no. 1-2, pp. 141–153, 2008.

[18] V. Pantelic, S. Postma, and M. Lawford, "Probabilistic supervisory control of probabilistic discrete event systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 2013–2018, Aug. 2009.

[19] S. Postma and M. Lawford, "Computation of probabilistic supervisory controllers for model matching," in *Proceedings of Allerton Conference on Communications, Control, and Computing*, 2004.

[20] Y. Deng, T. Chothia, C. Palamidessi, and J. Pang, "Metrics for action-labelled quantitative transition systems," *Electronic Notes in Theoretical Computer Science*, vol. 153, no. 2, pp. 79–96, 2006, also appeared in *Proceedings of the 3rd Workshop on Quantitative Aspects of Programming Languages*.

[21] V. Pantelic and M. Lawford, "Towards optimal supervisory control of probabilistic discrete event systems," in *DCDS 2009: Proceedings of 2nd IFAC Workshop on Dependable Control of Discrete Systems*, Bari, Italy, June 2009, pp. 85–90.

[22] A. Giacalone, C. Jou, and S. Smolka, "Algebraic reasoning for probabilistic concurrent systems," in *M. Broy and C.B. Jones, eds, Proceedings of the Working Conference on Programming Concepts and Methods*. Sea of Gallilee, Israel: North-Holland, 1990, pp. 443–458.

[23] I. Chattopadhyay and A. Ray, "Structural transformations of probabilistic finite state machines," *International Journal of Control*, vol. 81, no. 5, pp. 820–835, 2008.

[24] L. Kantorovich, "A new method for solving some classes of extremal problems," *Comptes Rendus (Doklady) Acad. Sci. USSR*, vol. 28, pp. 211–214, 1940.

[25] L. de Alfaro, T. A. Henzinger, and R. Majumdar, "Discounting the future in systems theory," in *ICALP*, ser. Lecture Notes in Computer Science, J. C. M. Baeten, J. K. Lenstra, J. Parrow, and G. J. Woeginger, Eds., vol. 2719. Springer, 2003, pp. 1022–1037.

[26] V. Pantelic and M. Lawford, "Use of a metric in supervisory control of probabilistic discrete event systems," in *WODES 2010: The 10th International Workshop on Discrete Event Systems*, Berlin, Germany, August 2010, to appear.

[27] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, "Metrics for labeled Markov systems," in *CONCUR*, ser. Lecture Notes in Computer Science, J. C. M. Baeten and S. Mauw, Eds., vol. 1664. Springer, 1999, pp. 258–273.

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

[29] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden, "Metrics for labelled Markov processes," *Theoretical Computer Science*, vol. 318, no. 3, pp. 323–354, 2004.

[30] F. van Breugel and J. Worrell, "An algorithm for quantitative verification of probabilistic transition systems," in *CONCUR*, ser. Lecture Notes in Computer Science, K. G. Larsen and M. Nielsen, Eds., vol. 2154. Springer, 2001, pp. 336–350.

[31] L. Schröder and P. Mateus, "Universal aspects of probabilistic automata," *Mathematical. Structures in Comp. Sci.*, vol. 12, no. 4, pp. 481–512, 2002.

[32] R. J. V. Glabbeek, S. A. Smolka, and B. Steffen, "Reactive, generative and stratified models of probabilistic processes," *Information and Computation*, vol. 121, pp. 130–141, 1990.

[33] J. Rutten, M. Kwiatkowska, G. Norman, and D. Parker, *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems*, P. Panangaden and F. van Breugel (eds.), ser. CRM Monograph Series. American Mathematical Society, 2004, vol. 23.

[34] M. Kwiatkowska, G. Norman, and D. Parker, "Stochastic model checking," in *Formal Methods for the Design of Computer, Communication and Software Systems: Performance Evaluation (SFM'07)*, ser. LNCS (Tutorial Volume), M. Bernardo and J. Hillston, Eds., vol. 4486. Springer, 2007, pp. 220–270.

[35] V. Pantelic, "Probabilistic supervisory control of probabilistic discrete event systems," Ph.D. dissertation, 2009, in preparation.

[36] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*. New York, New York, USA: Academic Press, Inc., 1970.

[37] F. van Breugel, B. Sharma, and J. Worrell, "Approximating a behavioural pseudometric without discount for probabilistic systems," *Logical Methods in Computer Science*, vol. 4, no. 2, 2008.

APPENDIX

1) Proof of Lemma 3

We use the notation from Lemma 2.

$$
\begin{aligned}
|\mathcal{D}_k(\mathfrak{d}') - \mathcal{D}_k(\mathfrak{d}'')| &\leq e \sum_{\sigma\in\Sigma} min(\rho_{\sigma,t}, \rho'_{\sigma,l})\, |d'(q_t, q_l) - d''(q_t, q_l)| \;\; \text{(from } (10)\text{)} \\
&\leq e \sum_{\sigma\in\Sigma} min(\rho_{\sigma,t}, \rho'_{\sigma,l}) \underset{\substack{(t,l)\in \\ \{(t(\sigma,i),l(\sigma,j))|\sigma\in\Sigma\}}}{max} \{|d'(q_t, q_l) - d''(q_t, q_l)|\} \\
&\leq e \sum_{\sigma\in\Sigma} min(\rho_{\sigma,t}, \rho'_{\sigma,l})\, |d'(q_r, q_s) - d''(q_r, q_s)| \\
&\qquad\qquad \text{for some } r, s \in \{0, \ldots, N-1\} \\
&\leq e|d'(q_r, q_s) - d''(q_r, q_s)| \\
&\qquad\qquad \text{(since } \sum_{\substack{\sigma\in\Sigma \\ t\in\{0,\ldots,N-1\}}} \rho_{\sigma,t} = \sum_{\substack{\sigma\in\Sigma \\ l\in\{0,\ldots,N-1\}}} \rho'_{\sigma,l} = 1\text{)} \\
&\leq e\, |\mathfrak{d}'_m - \mathfrak{d}''_m| \;\; \text{for some } m \in \{1, \ldots, N^2\}
\end{aligned}
$$

2) Proof of Lemma 4

According to Knaster-Tarski theorem, the functions f and g have the greatest fixed points $\mathit{gfp}(f)$ and $\mathit{gfp}(g)$, respectively, where $\mathit{gfp}(f) = sup(\{x|x \preceq f(x)\})$, and $\mathit{gfp}(g) = sup(\{x|x \preceq g(x)\})$. Since $\forall x : g(x) \preceq f(x)$, then $\{x|x \preceq g(x)\} \subseteq \{x|x \preceq f(x)\}$; hence $\mathit{gfp}(g) \preceq \mathit{gfp}(f)$.

3) Proof of Lemma 5

First, use the proof of Theorem 3 up to (22), and, then, analogous to the proof of Lemma 3.

4) Proof of Theorem 4

Analogous to the proof of Theorem 2 (with using Lemma 5 instead of Lemma 3).