

COMPUTING PROBABILISTIC BISIMILARITY DISTANCES FOR PROBABILISTIC AUTOMATA

GIORGIO BACCI^a, GIOVANNI BACCI^a, KIM G. LARSEN^a,
RADU MARDARE^b, QIYI TANG^c, AND FRANCK VAN BREUGEL^d

^a Dept. of Computer Science, Aalborg University, Denmark
e-mail address: {grbacci,giovbacci,kgl}@cs.aau.dk

^b Dept. of Computer and Information Sciences, University of Strathclyde, Glasgow, UK
e-mail address: r.mardare@strath.ac.uk

^c Dept. of Computer Science, Oxford University, UK
e-mail address: qiyi.tang@cs.ox.ac.uk

^d DisCoVeri Group, Dept. of Electrical Engineering and Computer Science, York University, Canada
e-mail address: franck@eecs.yorku.ca

ABSTRACT. The probabilistic bisimilarity distance of Deng et al. has been proposed as a robust quantitative generalization of Segala and Lynch’s probabilistic bisimilarity for probabilistic automata. In this paper, we present a characterization of the bisimilarity distance as the solution of a simple stochastic game. The characterization gives us an algorithm to compute the distances by applying Condon’s simple policy iteration on these games. The correctness of Condon’s approach, however, relies on the assumption that the games are *stopping*. Our games may be non-stopping in general, yet we are able to prove termination for this extended class of games. Already other algorithms have been proposed in the literature to compute these distances, with complexity in **UP** \cap **coUP** and **PPAD**. Despite the theoretical relevance, these algorithms are inefficient in practice. To the best of our knowledge, our algorithm is the first practical solution.

The characterization of the probabilistic bisimilarity distance mentioned above crucially uses a dual presentation of the Hausdorff distance due to Mémoli. As an additional contribution, in this paper we show that Mémoli’s result can be used also to prove that the bisimilarity distance bounds the difference in the maximal (or minimal) probability of two states to satisfying arbitrary ω -regular properties, expressed, eg., as LTL formulas.

Key words and phrases: probabilistic automata, behavioural pseudometrics, stochastic games.

* This paper is an extended version of an earlier conference paper [BBL⁺19] presented at CONCUR 2019. Giovanni Bacci and Kim G. Larsen are supported by the ERC-Project LASSO. Franck van Breugel is supported by Natural Sciences and Engineering Research Council of Canada.

1. INTRODUCTION

In [GJS90], Giacalone et al. observed that for reasoning about the behaviour of probabilistic systems, rather than equivalences, a notion of distance is more reasonable in practice since it permits to capture the degree of difference between two states. This observation motivated the study of behavioural pseudometrics, that generalize behavioural equivalences in the sense that, when the distance is zero then the two states are behaviourally equivalent.

The systems we consider in this paper are labelled *probabilistic automata*. This model was introduced by Segala [Seg95] to capture both nondeterminism (hence, concurrency) and probabilistic behaviours. The labels on states are used to express that certain properties of interest hold in particular states.

In Figure 1 we consider an example of a probabilistic automaton describing two gamblers, f and b , deciding on which team to bet in a football match. Typically the two gamblers know on which team to bet, but occasionally they prefer to toss a coin to make a decision. This is represented by the three probabilistic transitions in the state f . The first two take f to state h (head) or t (tail) with probability one, the last takes f to states h and t with probability $\frac{1}{2}$ each. The difference between f and b is that the former uses a fair coin while the latter uses a biased coin landing on heads with slightly higher probability. Once the decision is taken, it is not changed anymore. This is seen on states h and t which have a single probabilistic transition taking the state to itself with probability one. The states h and t have distinct labels, here represented by colours.

A behavioural pseudometric for probabilistic automata capturing this difference is the *probabilistic bisimilarity distance* by Deng et al. [DCPP06], introduced as a robust generalization of Segala and Lynch's probabilistic bisimilarity [SL94]. The key ingredients of this pseudometric are the Hausdorff metric [Hau14] and the Kantorovich metric [Kan42], respectively used to capture nondeterministic and probabilistic behaviour. In the example above, the behaviours of the states h and t are very different since their labels are different. As a result, their probabilistic bisimilarity distance is one. On the other hand, the behaviours of the states f and b are very similar, which is reflected by the fact that their probabilistic bisimilarity distance is $\frac{1}{100}$.

The first attempt to compute the above distance is due to Chen et al. [CHL07], who proposed a doubly exponential-time procedure to approximate the distances up to any degree of accuracy. The complexity was later improved to **PSPACE** by Chattarjee et al. [CdAMR08, CdAMR10]. Their solutions exploit the decision procedure for the existential

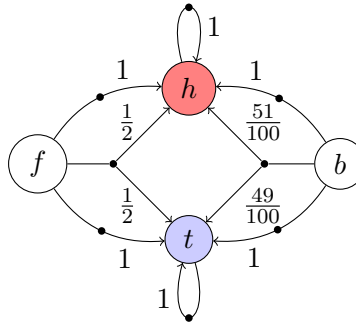


Figure 1: A probabilistic automaton describing two gablers.

fragment of the first-order theory of the reals. It is worth noting that [CHL07, CdAMR08] consider the pseudometric that does not discount the future (a.k.a. *undiscounted* distance) which entails additional algorithmic challenges. Later, Fu [Fu12] showed that the distances have rational values and that computing the discounted distance can be done in polynomial time by using a value-iteration procedure in combination with the continued fraction algorithm [Sch99, Section 6]. As for the undiscounted distance, he showed that the threshold problem, i.e., deciding whether the distance is smaller than a given rational, is in $\mathbf{NP} \cap \mathbf{coNP}$. The same proof can be adapted to show that the decision problem is in $\mathbf{UP} \cap \mathbf{coUP}$ [Fu14], where \mathbf{UP} is the subclass of \mathbf{NP} -problems with a unique accepting computation. Van Breugel and Worrell [vBW14] have later shown that the problem is in \mathbf{PPAD} , which is short for polynomial parity argument on directed graphs. Notably, their proof exploits a characterization of the distance as a simple stochastic game. The above algorithms were presented with the purpose of understanding the complexity of computing bisimilarity distances and, to the best of our knowledge, they have never been implemented. Their implementation would involve either an enumeration of possibly exponentially many fixed-points [Fu14], or the use of SMT solvers over the existential fragment of the first order theory of the reals [CdAMR08, CdAMR10]. An earlier attempt of approximating the bisimilarity distance for the more specific case of labelled Markov chains by expressing the problem in the existential fragment of the first order theory of the reals was proposed in [vBSW08]. Its latest implementation using CVC4 [BCD⁺11] is able to handle chains with 82 states in approximately 66 hours¹. In this paper, we propose an alternative approach that is inspired by the successful implementations of similar pseudometrics on labelled Markov chains [BBLM13, TvB16, TvB18a].

Our solution is based on a novel characterization of the probabilistic bisimilarity distance as the solution of a simple stochastic game. Stochastic games were introduced by Shapley [Sha53]. A simplified version of these games, called *simple stochastic games*, were studied by Condon [Con92]. Several algorithms have been proposed to compute the value function of a simple stochastic game, many using policy iteration. Condon [Con90] proposed an algorithm, known as *simple policy iteration*, that switches only one non-optimal choice per iteration. The correctness of Condon’s algorithm, however, relies on the assumption that the game is *stopping*.

It turns out that the simple stochastic games characterizing the probabilistic bisimilarity distances are stopping only when the distances discount the future. In the case the distance is non-discounting, the corresponding games may not be stopping. To recover correctness of the policy iteration procedure we adapt Condon’s simple policy iteration algorithm by adding a non-local update of the strategy of the min player and an extra termination condition based on a notion of “self-closed” relation due to Fu [Fu12]. The practical efficiency of our algorithm has been evaluated on a significant set of randomly generated probabilistic automata. The results show that our algorithm performs better than the corresponding iterative algorithms proposed for the discounted distances in [Fu12], even though the theoretical complexity of our proposal is exponential in the worst case (*cf.* [TvB16]) whereas Fu’s is polynomial. The implementation of the algorithms exploits a coupling structure characterization of the distance that allows us to skip the construction of the simple stochastic game which may result in an exponential blow up of the memory required for storing the game.

¹The code is available at bitbucket.org/discoveri/first-order.

The two characterizations of probabilistic bisimilarity distances proposed in this paper (either via simple stochastic games or coupling structures) crucially use a dual presentation of the Hausdorff distance due to Mémoli [Mém11]. Still using Mémoli's result, as an additional contribution to this paper we show that the (undiscounted) bisimilarity distance can be used to bound the difference of the maximal (or minimal) probability of two states satisfying arbitrary ω -regular specifications, expressed, *e.g.*, as LTL formulas. Notably, this result allows us to relate the probabilistic bisimilarity pseudometric of Deng et al. to probabilistic model checking of probabilistic automata against linear-time specifications.

Synopsis. Section 2 introduces the notation and some preliminary results used in the paper. In Section 3 we recall the definition of probabilistic bisimilarity distances of Deng et al. for probabilistic automata; then, in Section 4 we propose a characterisation of the probabilistic bisimilarity distances as the values of a simple stochastic game constructed from the automaton, here called probabilistic bisimilarity game. Towards an algorithmic solution for computing bisimilarity distances, in Section 5 we provide an alternative characterisation of the distances in terms of *coupling structures*. Section 6 describes a procedure for computing the bisimilarity distances based on Condon's simple policy iteration algorithm. In Section 7 we discuss the relation between the notion of bisimilarity distance and probabilistic model checking of ω -regular linear-time specifications against probabilistic automata. Finally, Section 8 concludes with some remarks and future work directions.

2. PRELIMINARIES AND NOTATION

The set of functions f from X to Y is denoted by Y^X . We denote by $f[x/y] \in Y^X$ the *update* of f at $x \in X$ with $y \in Y$, defined by $f[x/y](x') = y$ if $x' = x$, otherwise $f[x/y](x') = f(x')$.

A (1-bounded) *pseudometric* on a set X is a function $d: X \times X \rightarrow [0, 1]$ such that, $d(x, x) = 0$, $d(x, y) = d(y, x)$, and $d(x, y) \leq d(x, z) + d(z, y)$ for all $x, y, z \in X$.

Kantorovich lifting. A (discrete) probability distribution on X is a function $\mu: X \rightarrow [0, 1]$ such that $\sum_{x \in X} \mu(x) = 1$, and its support is $\text{supp}(\mu) = \{x \in X \mid \mu(x) > 0\}$. We denote by $\mathcal{D}(X)$ the set of probability distributions on X . A pseudometric d on X can be lifted to a pseudometric on probability distributions in $\mathcal{D}(X)$ by means of the Kantorovich lifting [Vil08].

The *Kantorovich lifting* of $d \in [0, 1]^{X \times X}$ on distributions $\mu, \nu \in \mathcal{D}(X)$ is defined by

$$\mathcal{K}(d)(\mu, \nu) = \sup \left\{ \sum_{x \in X} f(x) \cdot (\mu(x) - \nu(x)) \mid f \in L_d \right\}, \quad (\text{KANTOROVICH LIFTING})$$

where L_d denotes the set of non-expansive $[0, 1]$ -valued functions over X , i.e., functions $f: X \rightarrow [0, 1]$ such that, for all $x, y \in X$, $|f(x) - f(y)| \leq d(x, y)$.

The Kantorovich distance has the following well known dual formulation

$$\mathcal{K}(d)(\mu, \nu) = \min \left\{ \sum_{x, y \in X} d(x, y) \cdot \omega(x, y) \mid \omega \in \Omega(\mu, \nu) \right\},$$

where $\Omega(\mu, \nu)$ denotes the set of *measure-couplings* for the pair (μ, ν) , i.e., distributions $\omega \in \mathcal{D}(X \times X)$ such that, for all $x \in X$, $\sum_{y \in X} \omega(x, y) = \mu(x)$ and $\sum_{x \in X} \omega(x, y) = \nu(y)$. It is a well known fact that this dual characterisation can be equivalently stated by ranging

ω over the set of vertices $V(\Omega(\mu, \nu))$ of the polytope $\Omega(\mu, \nu)$, as a concave function on a convex polytope attains its minimum at a vertex of the polytope (see [KW67, page 260]). Furthermore, if the set X is finite, the set $V(\Omega(\mu, \nu))$ is finite too [KW67, page 259].

Hausdorff lifting. A pseudometric d on X can be lifted to nonempty subsets of X by means of the Hausdorff lifting. The *Hausdorff lifting* of $d \in [0, 1]^{X \times X}$ on nonempty subsets $A, B \subseteq X$ is defined by

$$\mathcal{H}(d)(A, B) = \max \left\{ \sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b) \right\}. \quad (\text{HAUSDORFF LIFTING})$$

Following Mémoli [Mém11, Lemma 3.1], the Hausdorff lifting has a dual characterization in terms of *set-couplings*². Given $A, B \subseteq X$, a set-coupling for (A, B) is a relation $R \subseteq X \times X$ with left and right projections respectively equal to A and B , i.e., $\{a \mid \exists b \in X. a R b\} = A$ and $\{b \mid \exists a \in X. a R b\} = B$. We write $\mathcal{R}(A, B)$ for the set of the set-couplings for (A, B) .

Theorem 2.1 [Mém11]. $\mathcal{H}(d)(A, B) = \inf \{ \sup_{(a,b) \in R} d(a, b) \mid R \in \mathcal{R}(A, B) \}$.

Clearly, for finite A, B , inf and sup in Theorem 2.1 can be replaced by min and max, respectively.

3. PROBABILISTIC AUTOMATA AND PROBABILISTIC BISIMILARITY DISTANCE

In this section we recall some definitions and results from the literature. In particular, we introduce the models of interest, *probabilistic automata*, its best known behavioural equivalence, Segala and Lynch's *probabilistic bisimilarity* [SL94], and its quantitative generalization due to Deng et al. [DCPP06].

A probabilistic automaton is a model of computation that combines nondeterministic and probabilistic behaviours. Similarly to a standard nondeterministic automaton, states are labelled to express that certain properties of interest hold in that state. A probabilistic automaton in a current state $s \in S$ can nondeterministically proceed to a next probabilistic state $\mu \in \mathcal{D}(S)$, representing the probability distribution with which the automaton will move to the next state. This can be formalised as follows:

Definition 3.1. A *probabilistic automaton* (PA) is a tuple $\mathcal{A} = (S, L, \rightarrow, \ell)$ consisting of a nonempty finite set S of states, a finite set of labels L , a finite total transition relation $\rightarrow \subseteq S \times \mathcal{D}(S)$, and a labelling function $\ell: S \rightarrow L$.

For simplicity we assume the transition relation \rightarrow to be total, that is, for all $s \in S$, there exists a $\mu \in \mathcal{D}(S)$ such that $(s, \mu) \in \rightarrow$. For the remainder of this paper we fix a probabilistic automaton $\mathcal{A} = (S, L, \rightarrow, \ell)$. We write $s \rightarrow \mu$ to denote $(s, \mu) \in \rightarrow$ and use $\delta(s)$ to denote the set $\{\mu \mid s \rightarrow \mu\}$ of successor distributions of s .

Next we recall the notion of probabilistic bisimilarity due to Segala and Lynch [SL94] for probabilistic automata. Their definition exploits the notion of lifting of a relation $R \subseteq S \times S$ on states to a relation $\tilde{R} \subseteq \mathcal{D}(S) \times \mathcal{D}(S)$ on probability distributions on states, originally introduced by Jonsson and Larsen [JL91], and defined by $\mu \tilde{R} \nu$ if there exists a measure-coupling $\omega \in \Omega(\mu, \nu)$ such that $\text{supp}(\omega) \subseteq R$.

Definition 3.2. A relation $R \subseteq S \times S$ is a *probabilistic bisimulation* if whenever $s R t$,

²Mémoli uses the terminology “correspondence.” To avoid confusion, we adopted the same terminology used in [PC19, Section 10.6].

- $\ell(s) = \ell(t)$,
- if $s \rightarrow \mu$ then there exists $t \rightarrow \nu$ such that $\mu \tilde{R} \nu$, and
- if $t \rightarrow \nu$ then there exists $s \rightarrow \mu$ such that $\mu \tilde{R} \nu$.

Two states $s, t \in S$ are *probabilistic bisimilar*, written $s \sim t$, if they are related by some probabilistic bisimulation.

Intuitively, two states are probabilistic bisimilar if they have the same label and each transition of the one state to a distribution μ can be matched by a transition of the other state to a distribution ν assigning the same probability to states that behave the same, and vice versa. Probabilistic bisimilarity is an equivalence relation and the largest probabilistic bisimulation.

Deng et al. [DCPP06] proposed a family of 1-bounded pseudometrics \mathbf{d}_λ , parametric on a *discount factor* $\lambda \in (0, 1]$, called *probabilistic bisimilarity pseudometrics*. The pseudometrics \mathbf{d}_λ are defined as the least fixed-point of the functions $\Delta_\lambda: [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$

$$\Delta_\lambda(d)(s, t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \lambda \cdot \mathcal{H}(\mathcal{K}(d))(\delta(s), \delta(t)) & \text{otherwise.} \end{cases}$$

The well-definition of \mathbf{d}_λ follows by Knaster-Tarski's fixed point theorem, given the fact that Δ_λ is a monotone function on the complete partial order of $[0, 1]$ -valued functions on $S \times S$ ordered point-wise by $d \sqsubseteq d'$ iff for all $s, t \in S$, $d(s, t) \leq d'(s, t)$.

The fact that probabilistic bisimilarity distances provide a quantitative generalization of bisimilarity is captured by the following theorem due to Deng et al. [DCPP06, Corollary 2.14].

Theorem 3.3. *For all $\lambda \in (0, 1]$, $\mathbf{d}_\lambda(s, t) = 0$ if and only if $s \sim t$.*

4. PROBABILISTIC BISIMILARITY DISTANCE AS A SIMPLE STOCHASTIC GAME

A *simple stochastic game* (SSG) consists of a finite directed graph whose vertices are partitioned into sets of *0-sinks*, *1-sinks*, *max vertices*, *min vertices*, and *random vertices*. The game is played by two players, the *max player* and the *min player*, with a single token. At each step of the game, the token is moved from a vertex to one of its successors. At a min vertex the min player chooses the successor, at a max vertex the max player chooses the successor, and at a random vertex the successor is chosen randomly according to a prescribed probability distribution. The max player wins a play of the game if the token reaches a 1-sink and the min player wins if the play reaches a 0-sink or continues forever without reaching a sink. Since the game is stochastic, the max player tries to maximize the probability of reaching a 1-sink whereas the min player tries to minimize that probability.

Definition 4.1. A *simple stochastic game* is a tuple (V, E, P) consisting of

- a finite directed graph (V, E) such that
 - V is partitioned into the sets: V_0 of *0-sinks*, V_1 of *1-sinks*, V_{\max} of *max vertices*, V_{\min} of *min vertices*, and V_{rnd} of *random vertices*;
 - the vertices in V_0 and V_1 have outdegree zero and all other vertices have outdegree at least one, and
- a function $P: V_{\text{rnd}} \rightarrow \mathcal{D}(V)$ such that for all $v \in V_{\text{rnd}}$ and $w \in V$, $P(v)(w) > 0$ iff $(v, w) \in E$.

The above definition is slightly more general than the one given by Condon in [Con92, Section 2]. Note that the outdegree of min, max and random vertices is at least one (instead of exactly two), there may be multiple 0-sinks and 1-sinks (rather than exactly one). However, a simple stochastic game as defined above can be transformed in polynomial-time into a simple stochastic game as defined in [Con92], as shown by Zwick and Paterson [ZP96].

A *strategy*, also known as *policy*, for the min player is a function $\sigma_{\min}: V_{\min} \rightarrow V$ that assigns the target of an outgoing edge to each min vertex, that is, for all $v \in V_{\min}$, $(v, \sigma_{\min}(v)) \in E$. Likewise, a strategy for the max player is a function $\sigma_{\max}: V_{\max} \rightarrow V$ that assigns the target of an outgoing edge to each max vertex. These strategies are known as *pure stationary* strategies. We can restrict ourselves to these strategies since the optimal among all strategies for both players are of this type (see, for example, [LL69]).

Such strategies determine a sub-game in which each max vertex and each min vertex has outdegree one (see [Con92, Section 2] for details). Such a game can naturally be viewed as a Markov chain. We write $\phi_{\sigma_{\min}, \sigma_{\max}}: V \rightarrow [0, 1]$ for the function that gives the probability of a vertex in this Markov chain to reach a 1-sink.

The *value function* $\phi: V \rightarrow [0, 1]$ of a SSG is defined as $\min_{\sigma_{\min}} \max_{\sigma_{\max}} \phi_{\sigma_{\min}, \sigma_{\max}}$. It is folklore that the value function of a simple stochastic game can be characterised as the least fixed point of the following function (see, for example, [Jub05, Section 2.2 and 2.3]).

Definition 4.2. The function $\Phi: [0, 1]^V \rightarrow [0, 1]^V$ is defined by

$$\Phi(f)(v) = \begin{cases} 0 & \text{if } v \in V_0 \\ 1 & \text{if } v \in V_1 \\ \max_{(v,w) \in E} f(w) & \text{if } v \in V_{\max} \\ \min_{(v,w) \in E} f(w) & \text{if } v \in V_{\min} \\ \sum_{(v,w) \in E} P(v)(w) f(w) & \text{if } v \in V_{\text{rnd}} \end{cases}$$

The $[0, 1]$ -valued functions on V can be ordered point-wise by $f \sqsubseteq g$ iff for all $v \in V$, $f(v) \leq g(v)$. This partial order is complete in $[0, 1]^V$, with meet and join respectively given by the point-wise infimum and supremum.

Then, the existence the least fixed point of Φ is ensured by Knaster-Tarski's fixed point theorem and the following result.

Proposition 4.3. *The function Φ is monotone.*

Proof. Let $f, g \in [0, 1]^V$ and $f \sqsubseteq g$. Let $v \in V$. It suffices to show that $\Phi(f)(v) \leq \Phi(g)(v)$. We distinguish the following cases.

- If $v \in V_0$ then $\Phi(f)(v) = 0 = \Phi(g)(v)$.
- If $v \in V_1$ then $\Phi(f)(v) = 1 = \Phi(g)(v)$.
- If $v \in V_{\max}$ then $\Phi(f)(v) = \max_{(v,w) \in E} f(w) \leq \max_{(v,w) \in E} g(w) = \Phi(g)(v)$.
- If $v \in V_{\min}$ then $\Phi(f)(v) = \min_{(v,w) \in E} f(w) \leq \min_{(v,w) \in E} g(w) = \Phi(g)(v)$.
- If $v \in V_{\text{rnd}}$ then

$$\Phi(f)(v) = \sum_{(v,w) \in E} P(v)(w) f(w) \leq \sum_{(v,w) \in E} P(v)(w) g(w) = \Phi(g)(v). \quad \square$$

The set $[0, 1]^V$ can be turned into a Banach space by means of the supremum norm $\|f\| = \max_{v \in V} f(v)$. Recall that a function $F: [0, 1]^V \rightarrow [0, 1]^V$ is non-expansive if for all $f, g \in [0, 1]^V$, $\|f - g\| \geq \|F(f) - F(g)\|$.

Proposition 4.4. *The function Φ is nonexpansive.*

Proof. Let $f, g \in [0, 1]^V$. Let $v \in V$. It suffices to show that $|\Phi(f)(v) - \Phi(g)(v)| \leq \|f - g\|$. We distinguish the following cases.

- If $v \in V_0$ then $|\Phi(f)(v) - \Phi(g)(v)| = |0 - 0| = 0 \leq \|f - g\|$.
- If $v \in V_1$ then $|\Phi(f)(v) - \Phi(g)(v)| = |1 - 1| = 0 \leq \|f - g\|$.
- Let $v \in V_{\max}$. Without loss of generality, assume that $\max_{(v,w) \in E} f(w) \geq \max_{(v,w) \in E} g(w)$. Then

$$\begin{aligned} |\Phi(f)(v) - \Phi(g)(v)| &= \left| \max_{(v,w) \in E} f(w) - \max_{(v,w) \in E} g(w) \right| \\ &= \max_{(v,w) \in E} f(w) - \max_{(v,w) \in E} g(w) \\ &= f(x) - \max_{(v,w) \in E} g(w) \\ &\leq f(x) - g(x) \\ &\leq \|f - g\|, \end{aligned}$$

where $x \in V$ realises the maximum of $\{f(w) \mid (v, w) \in E\}$.

- Let $v \in V_{\min}$. Without loss of generality, assume that $\min_{(v,w) \in E} f(w) \geq \min_{(v,w) \in E} g(w)$. Then

$$\begin{aligned} |\Phi(f)(v) - \Phi(g)(v)| &= \left| \min_{(v,w) \in E} f(w) - \min_{(v,w) \in E} g(w) \right| \\ &= \min_{(v,w) \in E} f(w) - \min_{(v,w) \in E} g(w) \\ &= \min_{(v,w) \in E} f(w) - g(x) \\ &\leq f(x) - g(x) \\ &\leq \|f - g\|, \end{aligned}$$

where $x \in V$ realises the minimum of $\{g(w) \mid (v, w) \in E\}$.

- If $v \in V_{\text{rnd}}$ then

$$\begin{aligned} |\Phi(f)(v) - \Phi(g)(v)| &= \left| \sum_{(v,w) \in E} P(v)(w) f(w) - \sum_{(v,w) \in E} P(v)(w) g(w) \right| \\ &= \left| \sum_{(v,w) \in E} P(v)(w) (f(w) - g(w)) \right| \\ &\leq \sum_{(v,w) \in E} P(v)(w) \|f - g\| \\ &\leq \|f - g\|. \end{aligned}$$

□

4.1. A Probabilistic Bisimilarity Game. Fix a probabilistic automaton \mathcal{A} and $\lambda \in (0, 1]$. We will characterise the probabilistic bisimilarity distances as values of a simple stochastic game, which we call the *probabilistic bisimilarity game*, where the min player tries to show that two states are probabilistic bisimilar, while the max player tries to prove the opposite.

In our probabilistic bisimilarity game, there is a vertex (s, t) for each pair states s and t in \mathcal{A} . If $\ell(s) \neq \ell(t)$ then the vertex (s, t) is a 1-sink. Otherwise, (s, t) is a min vertex. In this vertex, the min player selects a set $R \in \mathcal{R}(\delta(s), \delta(t))$ of pairs of transitions. This set R captures potential matchings of transitions from state s and state t . Subsequently, the max player chooses a pair of transitions from the set R . Once the max player has chosen a pair (μ, ν) from the set R corresponding to the transitions $s \rightarrow \mu$ and $t \rightarrow \nu$, the min player can choose a measure-coupling $\omega \in \Omega(\mu, \nu)$. To ensure that the game graph is finite, we restrict our attention to the vertices $V(\Omega(\mu, \nu))$ of the polytope $\Omega(\mu, \nu)$. Such a measure-coupling ω captures a matching of the probability distributions μ and ν . Recall that a measure-coupling is a probability distribution on $S \times S$. From a random vertex ω , the game proceeds to vertex (u, v) with probability $\lambda \cdot \omega(u, v)$ and to the 0-sink vertex \perp with probability $1 - \lambda$. Intuitively, the choices of $R \in \mathcal{R}(\delta(s), \delta(t))$ and then $(\mu, \nu) \in R$, performed respectively by the min and the max player, correspond to the min and max of Theorem 2.1; analogously, the selection of $\omega \in V(\Omega(\mu, \nu))$ by the min player models the min in the definition of the Kantorovich lifting.

Formally, our probabilistic bisimilarity game for the automaton \mathcal{A} is defined as follows.

Definition 4.5. Let $\lambda \in (0, 1]$. The *probabilistic bisimilarity game* (V, E, P) is defined by

- $V_0 = \{\perp\}$,
- $V_1 = \{(s, t) \in S \times S \mid \ell(s) \neq \ell(t)\}$,
- $V_{\max} = \bigcup \{\mathcal{R}(\delta(s), \delta(t)) \mid (s, t) \in V_{\min}\}$,
- $V_{\min} = \{(s, t) \in S \times S \mid \ell(s) = \ell(t)\} \cup \bigcup \{R \mid R \in V_{\max}\}$,
- $V_{\text{rnd}} = \bigcup \{V(\Omega(\mu, \nu)) \mid (\mu, \nu) \in V_{\min}\}$,

$$\begin{aligned} E = & \{((s, t), R) \mid (s, t) \in V_{\min} \wedge R \in \mathcal{R}(\delta(s), \delta(t))\} \cup \\ & \{(R, (\mu, \nu)) \mid R \in V_{\max} \wedge (\mu, \nu) \in R\} \cup \\ & \{((\mu, \nu), \omega) \mid (\mu, \nu) \in V_{\min} \wedge \omega \in V(\Omega(\mu, \nu))\} \cup \\ & \{(\omega, (u, v)) \mid \omega \in V_{\text{rnd}} \wedge (u, v) \in \text{supp}(\omega)\} \cup \{(\omega, \perp) \mid \omega \in V_{\text{rnd}}\}, \end{aligned}$$

and, for all $\omega \in V_{\text{rnd}}$ and $(s, t) \in \text{supp}(\omega)$, $P(\omega)((s, t)) = \lambda \cdot \omega(s, t)$ and $P(\omega)(\perp) = 1 - \lambda$.

By construction of the probabilistic bisimilarity game, there is a direct correspondence between the function Φ from Definition 4.2 associated to the probabilistic bisimilarity game and the function Δ_λ from Section 3 associated to the probabilistic automaton. From this correspondence it is straightforward that the respective least fixed points of Φ and Δ_λ agree, that is, the probabilistic bisimilarity distances of a probabilistic automaton are the values of the corresponding vertices of the probabilistic bisimilarity game.

Theorem 4.6. For all $\lambda \in (0, 1]$ and $s, t \in S$, $\mathbf{d}_\lambda(s, t) = \phi(s, t)$.

Proof. The proof is similar to that of [vBW14, Theorem 14].

Let ϕ be the value function of the probabilistic bisimilarity game. Since Φ is monotone and non-expansive (Propositions 4.3 and 4.4), we conclude from [vB12, Corollary 1] that the closure ordinal of Φ is ω , that is, ϕ is the least upper bound of $\{\Phi^n(\mathbf{0}) \mid n \in \mathbb{N}\}$, where the function $\mathbf{0}$ maps every vertex to zero. Similarly, \mathbf{d}_λ is the least upper bound of $\{\Delta_\lambda^n(\mathbf{0}) \mid n \in \mathbb{N}\}$, where the function $\mathbf{0}$ maps every state pair to zero. Therefore, it suffices to show that for all $s, t \in S$ and $n \in \mathbb{N}$,

$$\Phi^{4n}(\mathbf{0})(s, t) = \Delta_\lambda^n(\mathbf{0})(s, t)$$

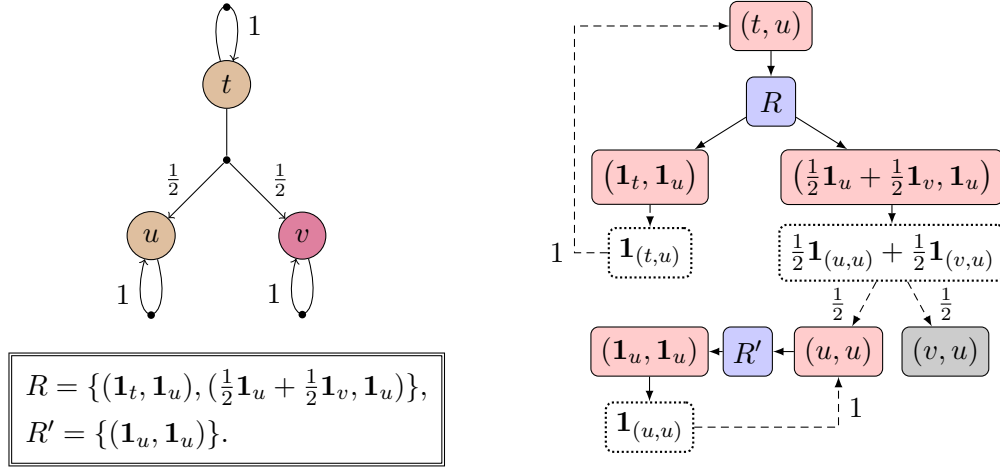


Figure 2: (Top left:) A probabilistic automaton and (Right:) the associated simple stochastic game constructed as in Definition 4.5 for $\lambda = 1$ (only the portion reachable from (t, u) is shown), where $\mathbf{1}_x$ denotes the Dirac distribution concentrated at x .

by induction on n . Obviously, the above holds if $n = 0$. Let $n > 0$. We distinguish the following cases.

- If $\ell(s) \neq \ell(t)$ then the vertex (s, t) is a 1-sink and, hence, $\Phi^{4n}(\mathbf{0})(s, t) = 1 = \Delta_\lambda^n(\mathbf{0})(s, t)$.
- If $\ell(s) = \ell(t)$ then

$$\begin{aligned}
\Phi^{4n}(\mathbf{0})(s, t) &= \\
&= \min_{R \in \mathcal{R}(\delta(s), \delta(t))} \Phi^{4n-1}(\mathbf{0})(R) \\
&= \min_{R \in \mathcal{R}(\delta(s), \delta(t))} \max_{(\mu, \nu) \in R} \Phi^{4n-2}(\mathbf{0})(\mu, \nu) \\
&= \min_{R \in \mathcal{R}(\delta(s), \delta(t))} \max_{(\mu, \nu) \in R} \min_{\omega \in V(\Omega(\mu, \nu))} \Phi^{4n-3}(\mathbf{0})(\omega) \\
&= \min_{R \in \mathcal{R}(\delta(s), \delta(t))} \max_{(\mu, \nu) \in R} \min_{\omega \in V(\Omega(\mu, \nu))} \sum_{(u, v) \in \text{supp}(\omega)} \lambda \omega(u, v) \Phi^{4n-4}(\mathbf{0})(u, v) + (1 - \lambda) \Phi^{4n-4}(\mathbf{0})(\perp) \\
&= \min_{R \in \mathcal{R}(\delta(s), \delta(t))} \max_{(\mu, \nu) \in R} \min_{\omega \in V(\Omega(\mu, \nu))} \lambda \sum_{u, v \in S} \omega(u, v) \Phi^{4n-4}(\mathbf{0})(u, v) \quad (\perp \text{ is a 0-sink}) \\
&= \min_{R \in \mathcal{R}(\delta(s), \delta(t))} \max_{(\mu, \nu) \in R} \min_{\omega \in V(\Omega(\mu, \nu))} \lambda \sum_{u, v \in S} \omega(u, v) \Delta_\lambda^{n-1}(\mathbf{0})(u, v) \quad (\text{by induction}) \\
&= \lambda \cdot \min_{R \in \mathcal{R}(\delta(s), \delta(t))} \max_{(\mu, \nu) \in R} \mathcal{K}(\Delta_\lambda^{n-1}(\mathbf{0}))(\mu, \nu) \\
&= \lambda \cdot \mathcal{H}(\mathcal{K}(\Delta_\lambda^{n-1}(\mathbf{0}))) (\delta(s), \delta(t)) \quad (\text{Theorem 2.1}) \\
&= \Delta_\lambda^n(\mathbf{0})(s, t). \quad \square
\end{aligned}$$

Consider a state pair (s, t) with $s \sim t$. By Theorem 3.3, $\mathbf{d}_\lambda(s, t) = 0$. Hence, from Theorem 4.6 we can conclude that $\phi(s, t) = 0$. Therefore, by pre-computing probabilistic bisimilarity, (s, t) can be represented as a 0-sink, rather than a min vertex. For example, in Figure 2 this amounts to turning (u, u) into a 0-sink and disconnecting it from its successors.

Games similar to the above introduced probabilistic bisimilarity game have been presented in [DLT08, vBW14, FKP17, KM18]. The game presented by van Breugel and Worrell in [vBW14] is most closely related to our game. They also consider probabilistic automata and map a probabilistic automaton to a simple stochastic game. The only difference is that they use the original definition of the Hausdorff distance, whereas we use Mémoli's alternative characterization. The games described in [DLT08, FKP17, KM18] are not stochastic. Desharnais, Laviolette and Tracol [DLT08] define an ϵ -probabilistic bisimulation game for probabilistic automata, where $\epsilon > 0$ captures the maximal amount of difference in behaviour that is allowed. Their measure of difference in behaviour is incomparable to our probabilistic bisimilarity distances (see [DLT08, Section 6]). König and Mika-Michalski [KM18] generalize the game of Desharnais et al. in a categorical setting so that it is applicable to a large class of systems including probabilistic automata. Fijalkow, Klin and Panangaden [FKP17] consider a more restricted class of systems, namely systems with probabilities but without nondeterminism. In the games in [DLT08, FKP17, KM18] players choose sets of states, a phenomenon that one does not encounter in our game.

5. A COUPLING CHARACTERISATION OF THE BISIMILARITY DISTANCE

In this section we provide an alternative characterisation for the probabilistic bisimilarity distance \mathbf{d}_λ based on the notion of coupling structure for a probabilistic automaton. This characterisation generalises the one by Chen et al. [CvBW12, Theorem 8] (see also [BBLM13, Theorem 8]) for the bisimilarity pseudometric of Desharnais et al. [DGJP04] for labelled Markov chains. Our construction exploits Mémoli's dual characterisation of the Hausdorff distance (Theorem 2.1).

Definition 5.1. A *coupling structure* for \mathcal{A} is a tuple $\mathcal{C} = (f, \rho)$ consisting of

- a map $f: \mathcal{D}(S) \times \mathcal{D}(S) \rightarrow \mathcal{D}(S \times S)$ such that, for all $\mu, \nu \in \mathcal{D}(S)$, $f(\mu, \nu) \in \Omega(\mu, \nu)$, and
- a map $\rho: S \times S \rightarrow 2^{\mathcal{D}(S) \times \mathcal{D}(S)}$, such that for all $s, t \in S$, $\rho(s, t) \in \mathcal{R}(\delta(s), \delta(t))$.

For convenience, the components f and ρ of a coupling structure will be called *measure-coupling map* and *set-coupling map*, respectively.

The definition of coupling structure is better understood in relation to the automaton it induces. The probabilistic automaton induced from $\mathcal{C} = (f, \rho)$, denoted

$$\mathcal{A}_{\mathcal{C}} = (S \times S, L \times L, \rightarrow_{\mathcal{C}}, \ell_{\mathcal{C}}),$$

has $S \times S$ as set of states, $L \times L$ as set of labels, transition relation $\rightarrow_{\mathcal{C}} \subseteq (S \times S) \times \mathcal{D}(S \times S)$, defined as $(s, t) \rightarrow_{\mathcal{C}} f(\mu, \nu)$ if $(\mu, \nu) \in \rho(s, t)$, and labeling function $\ell_{\mathcal{C}}: S \times S \rightarrow L \times L$ defined as $\ell_{\mathcal{C}}(s, t) = (\ell(s), \ell(t))$. Intuitively, $\mathcal{A}_{\mathcal{C}}$ describes the concurrent execution of two copies of the probabilistic automaton \mathcal{A} , synchronized by the coupling structure \mathcal{C} . Coupling structures are used to limiting the non-determinism only to set-couplings by fixing a particular choice $f(\mu, \nu)$ of measure-coupling between any pair of distributions μ, ν .

Let $\lambda \in (0, 1]$. For each \mathcal{C} we define the function $\Gamma_{\lambda}^{\mathcal{C}}: [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$ as

$$\Gamma_{\lambda}^{\mathcal{C}}(d)(s, t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \lambda \cdot \max\{\sum_{u, v \in S} d(u, v) \cdot \omega(u, v) \mid (s, t) \rightarrow_{\mathcal{C}} \omega\} & \text{otherwise.} \end{cases}$$

Lemma 5.2. *The function $\Gamma_{\lambda}^{\mathcal{C}}$ is well-defined and monotone.*

Proof. The well definition of Γ_λ^C follows by the fact that $\lambda \in (0, 1]$ and $\sum_{u,v \in S} d(u, v) \cdot \omega(u, v)$ is a convex combination of a sequence of $[0, 1]$ -valued numbers, namely $(d(u, v))_{u,v \in S}$.

As for monotonicity, let $d, d' \in [0, 1]^{S \times S}$ and $d \sqsubseteq d'$. Let $s, t \in S$, it suffices to show that $\Gamma_\lambda^C(d)(s, t) \leq \Gamma_\lambda^C(d')(s, t)$. We distinguish the following cases:

- If $\ell(s) \neq \ell(t)$, then $\Gamma_\lambda^C(d)(s, t) = 1 = \Gamma_\lambda^C(d')(s, t)$.
- If $\ell(s) = \ell(t)$, then we have

$$\begin{aligned}
 \Gamma_\lambda^C(d)(s, t) &= \lambda \cdot \max\{\sum_{u,v \in S} d(u, v) \cdot \omega(u, v) \mid (s, t) \rightarrow_C \omega\} \\
 &= \lambda \sum_{u,v \in S} d(u, v) \cdot \omega^*(u, v) && \text{(for some } (s, t) \rightarrow_C \omega^*) \\
 &\leq \lambda \sum_{u,v \in S} d'(u, v) \cdot \omega^*(u, v) && (d \sqsubseteq d' \text{ and } \omega^*(u, v) \geq 0 \text{ for all } u, v \in S) \\
 &\leq \lambda \cdot \max\{\sum_{u,v \in S} d'(u, v) \cdot \omega(u, v) \mid (s, t) \rightarrow_C \omega\} && ((s, t) \rightarrow_C \omega^*) \\
 &= \Gamma_\lambda^C(d')(s, t). \quad \square
 \end{aligned}$$

By Knaster-Tarski's fixed point theorem, Γ_λ^C has a least fixed point, denoted by γ_λ^C . As in [BBLM13], we call γ_λ^C the λ -discounted discrepancy w.r.t. \mathcal{C} or simply λ -discrepancy.

Remark 5.3. Note that, the 1-discrepancy $\gamma_1^C(s, t)$ is the maximal probability of reaching a pair of states (u, v) in the probabilistic automaton \mathcal{A}_C such that $\ell(u) \neq \ell(v)$ by starting from the state pair (s, t) . It is well known that the maximal reachability probability can be computed in polynomial-time as the optimal solution of a linear program (see [BK08, Theorem 10.100] or [Put94, Chapter 6]). The linear program can be trivially generalized to compute γ_λ^C , for any $\lambda \in (0, 1]$.

Lemma 5.4. For all $\lambda \in (0, 1]$ and coupling structure \mathcal{C} of \mathcal{A} , $\Delta_\lambda(\gamma_\lambda^C) \sqsubseteq \gamma_\lambda^C$.

Proof. Let $\mathcal{C} = (f, \rho)$. Let $s, t \in S$ and $R = \rho(s, t)$. We distinguish two cases.

- If $\ell(s) \neq \ell(t)$, then $\Delta_\lambda(\gamma_\lambda^C)(s, t) = 1 = \gamma_\lambda^C(s, t)$.
- If $\ell(s) = \ell(t)$,

$$\begin{aligned}
 \Delta_\lambda(\gamma_\lambda^C)(s, t) &= \lambda \cdot \mathcal{H}(\mathcal{K}(\gamma_\lambda^C))(\delta(s), \delta(t)) && \text{(def. } \Delta_\lambda) \\
 &= \lambda \cdot \min\{\max_{(\mu, \nu) \in R'} \mathcal{K}(\gamma_\lambda^C)(\mu, \nu) \mid R' \in \mathcal{R}(\delta(s), \delta(t))\} && \text{(Theorem 2.1)} \\
 &\leq \lambda \cdot \max_{(\mu, \nu) \in R} \mathcal{K}(\gamma_\lambda^C)(\mu, \nu) && (R \in \mathcal{R}(\delta(s), \delta(t))) \\
 &= \lambda \cdot \max_{(\mu, \nu) \in R} \min_{\omega \in \Omega(\mu, \nu)} \sum_{u,v \in S} \gamma_\lambda^C(u, v) \cdot \omega(u, v) && \text{(def. } \mathcal{K}(\gamma_\lambda^C)) \\
 &\leq \lambda \cdot \max_{(\mu, \nu) \in R} \sum_{u,v \in S} \gamma_\lambda^C(u, v) \cdot f(\mu, \nu)(u, v) && (f(\mu, \nu) \in \Omega(\mu, \nu)) \\
 &= \lambda \cdot \max\{\sum_{u,v \in S} \gamma_\lambda^C(u, v) \cdot \omega(u, v) \mid (s, t) \rightarrow_C \omega\} && \text{(def. } \rightarrow_C) \\
 &= \Gamma_\lambda^C(\gamma_\lambda^C)(s, t) && \text{(def. } \Gamma_\lambda^C) \\
 &= \gamma_\lambda^C(s, t). && (\gamma_\lambda^C \text{ fixed point of } \Gamma_\lambda^C)
 \end{aligned}$$

By the generality of the chosen s and t , we conclude that $\Delta_\lambda(\gamma_\lambda^C) \sqsubseteq \gamma_\lambda^C$. \square

Corollary 5.5. For all $\lambda \in (0, 1]$ and coupling structure \mathcal{C} for \mathcal{A} , $\mathbf{d}_\lambda \sqsubseteq \gamma_\lambda^C$.

Proof. By Knaster-Tarski's fixed point theorem, \mathbf{d}_λ is the least prefix point of Δ_λ , therefore by Lemma 5.4 we can conclude that $\mathbf{d}_\lambda \sqsubseteq \gamma_\lambda^C$. \square

The next lemma shows that the probabilistic bisimilarity distance can be characterised as the λ -discrepancy for a *vertex coupling structure*, that is, a coupling structure $\mathcal{C} = (f, \rho)$ such that $f(\mu, \nu) \in V(\Omega(\mu, \nu))$ for all $\mu, \nu \in \mathcal{D}(S)$.

Lemma 5.6. *For all $\lambda \in (0, 1]$ there exists a vertex coupling structure \mathcal{C} for \mathcal{A} such that $\mathbf{d}_\lambda = \gamma_\lambda^\mathcal{C}$.*

Proof. We construct a vertex coupling structure $\mathcal{C} = (f, \rho)$ as follows.

We define $f: \mathcal{D}(S) \times \mathcal{D}(S) \rightarrow \mathcal{D}(S \times S)$ by

$$f(\mu, \nu) \in \operatorname{argmin}_{\omega \in V(\Omega(\mu, \nu))} \sum_{u, v \in S} \mathbf{d}_\lambda(u, v) \cdot \omega(u, v).$$

Hence,

$$\mathcal{K}(\mathbf{d}_\lambda)(\mu, \nu) = \sum_{u, v \in S} \mathbf{d}_\lambda(u, v) \cdot f(\mu, \nu)(u, v). \quad (5.1)$$

We define $\rho: S \times S \rightarrow 2^{\mathcal{D}(S) \times \mathcal{D}(S)}$ by

$$\rho(s, t) = \left\{ (\mu, \operatorname{argmin}_{\nu \in \delta(t)} \mathcal{K}(\mathbf{d}_\lambda)(\mu, \nu)) \mid \mu \in \delta(s) \right\} \cup \left\{ (\operatorname{argmin}_{\mu \in \delta(s)} \mathcal{K}(\mathbf{d}_\lambda)(\mu, \nu), \nu) \mid \nu \in \delta(t) \right\}.$$

Hence, $\rho(s, t) \in \mathcal{R}(\delta(s), \delta(t))$ and

$$\mathcal{H}(\mathcal{K}(\mathbf{d}_\lambda))(\delta(s), \delta(t)) = \max\{\mathcal{K}(\mathbf{d}_\lambda)(\mu, \nu) \mid (\mu, \nu) \in \rho(s, t)\}. \quad (5.2)$$

Next, we show that $\Gamma_\lambda^\mathcal{C}(\mathbf{d}_\lambda) \sqsubseteq \mathbf{d}_\lambda$. Let $s, t \in S$. We distinguish two cases:

- If $\ell(s) \neq \ell(t)$, then $\mathbf{d}_\lambda(s, t) = \Delta_\lambda(\mathbf{d}_\lambda)(s, t) = 1 = \Gamma_\lambda^\mathcal{C}(\mathbf{d}_\lambda)(s, t)$.
- If $\ell(s) = \ell(t)$, we have

$$\begin{aligned} \Gamma_\lambda^\mathcal{C}(\mathbf{d}_\lambda)(s, t) &= \lambda \cdot \max\{\sum_{u, v \in S} \mathbf{d}_\lambda(u, v) \cdot \omega(u, v) \mid (s, t) \rightarrow_\mathcal{C} \omega\} && (\text{def. } \Gamma_\lambda^\mathcal{C}) \\ &= \lambda \cdot \max\{\sum_{u, v \in S} \mathbf{d}_\lambda(u, v) \cdot f(\mu, \nu)(u, v) \mid (\mu, \nu) \in \rho(s, t)\} && (\text{def. } \rightarrow_\mathcal{C}) \\ &= \lambda \cdot \max\{\mathcal{K}(\mathbf{d}_\lambda)(\mu, \nu) \mid (\mu, \nu) \in \rho(s, t)\} && (\text{eq. (5.1)}) \\ &= \lambda \cdot \mathcal{H}(\mathcal{K}(\mathbf{d}_\lambda))(\delta(s), \delta(t)) && (\text{eq. (5.2)}) \\ &= \mathbf{d}_\lambda(s, t). && (\mathbf{d}_\lambda \text{ fixed point of } \Delta_\lambda) \end{aligned}$$

Therefore $\Gamma_\lambda^\mathcal{C}(\mathbf{d}_\lambda) = \mathbf{d}_\lambda$. Since $\gamma_\lambda^\mathcal{C}$ is the least fixed point of $\Gamma_\lambda^\mathcal{C}$, by Knaster-Tarski's fixed point theorem $\gamma_\lambda^\mathcal{C} \sqsubseteq \mathbf{d}_\lambda$. Moreover, by Corollary 5.5, $\mathbf{d}_\lambda \sqsubseteq \gamma_\lambda^\mathcal{C}$. Thus $\mathbf{d}_\lambda = \gamma_\lambda^\mathcal{C}$. \square

Theorem 5.7. *Let $\lambda \in (0, 1]$. Then, the following hold:*

- (1) $\mathbf{d}_\lambda = \sqcap\{\gamma_\lambda^\mathcal{C} \mid \mathcal{C} \text{ coupling structure for } \mathcal{A}\}$;
- (2) $s \sim t$ iff $\gamma_\lambda^\mathcal{C}(s, t) = 0$ for some vertex coupling structure \mathcal{C} for \mathcal{A} .

Proof. (1) follows by Corollary 5.5 and Lemma 5.6; (2) by Theorem 3.3 and Lemma 5.6. \square

Note that together with Lemma 5.6, Theorem 5.7.1 states that \mathbf{d}_λ is the minimal λ -discrepancy obtained by ranging over the subset of vertex coupling structures.

Remark 5.8 (On the relation with probabilistic bisimilarity games). The coupling structure characterization of the distance is strongly related to the simple stochastic game characterization presented in Section 4. Indeed, the notion of vertex coupling structure captures essentially the strategies for the min player on a probabilistic bisimilarity game in the following sense: the measure-coupling map component describes the strategy on the vertices of the form $(\mu, \nu) \in R$ for some $R \in V_{\max}$, while the set-coupling map deals with the description of the strategy on the min vertices $(s, t) \in S \times S$. The discrepancy $\gamma_1^\mathcal{C}$ captures the value w.r.t an optimal strategy for the max player when the min player has fixed their strategy a priori.

6. COMPUTING THE BISIMILARITY DISTANCE

We describe a procedure for computing the bisimilarity distances based on Condon's simple policy iteration algorithm [Con90]. Our procedure extends a similar one proposed in [TvB16, BBLM13] for computing the bisimilarity distances of Desharnais et al. [DGJP04] for labelled Markov chains. The extension takes into account the additional presence of nondeterminism in the choice of the transitions.

Condon's simple policy iteration algorithm computes the values of a simple stochastic game provided that the game is *stopping*, i.e., for each pair of strategies for the min and max players the token reaches a 0-sink or 1-sink vertex with probability one.

As we have shown in Theorem 4.6, the probabilistic bisimilarity distances are the values of the corresponding vertices in the simple stochastic game given in Definition 4.5. Thus, if we prove that the game is stopping we can apply Condon's simple policy iteration algorithm to compute the probabilistic bisimilarity distances.

Proposition 6.1. *For $\lambda \in (0, 1)$, the simple stochastic game in Definition 4.5 is stopping.*

Proof. For each pair of strategies for the min and max players, each vertex in the induced Markov chain reaches the 0-sink vertex \perp with probability at least $1 - \lambda$. Since $\lambda < 1$, from any state, the probability of never reaching \perp is zero, i.e., the probability of eventually reaching the sink state \perp is one. \square

However, for $\lambda = 1$ the game in Definition 4.5 may not be stopping as shown below.

Example 6.2. Consider the probabilistic automaton in Figure 2 and its associated probabilistic bisimilarity game. By choosing a strategy σ_{\max} for the max player such that $\sigma_{\max}(R) = (\mathbf{1}_t, \mathbf{1}_u)$, the vertex (t, u) has probability zero to reach a sink. This can be seen in Figure 2, since there are no paths using the edge $(R, (\mathbf{1}_t, \mathbf{1}_u))$ leading to a sink.

In [TvB16], by imposing the bisimilar state pairs to be 0-sinks, for the case of labelled Markov chains the simple stochastic game was proven to be stopping. This method does not generalize to probabilistic automata. Indeed, Example 6.2 provides a counterexample even when bisimilar state pairs are 0-sinks.

In the remainder of the section, we provide a general algorithm to compute the bisimilarity distance for every $\lambda \in (0, 1]$, by adapting Condon's simple policy iteration algorithm. Our solution will exploit the coupling characterization of the distance discussed in Section 5. This allows us to skip the construction of the simple stochastic game which may have size exponential in the number of states of the automaton.

6.1. Simple Policy Iteration Strategy. Condon's algorithm iteratively updates the strategies of the min and max players in turn, on the basis of the current over-approximation of the value of the game. Next we show how Condon's policy updates can be performed directly on coupling structures.

For the update of the coupling structure, we use a measure-coupling map $k(d)(\mu, \nu) \in V(\Omega(\mu, \nu))$ and a set-coupling map $h(d)(s, t) \in \mathcal{R}(\delta(s), \delta(t))$ such that

$$k(d)(\mu, \nu) \in \operatorname{argmin} \left\{ \sum_{u,v \in S} \omega(u, v) \cdot d(u, v) \mid \omega \in V(\Omega(\mu, \nu)) \right\}, \text{ and} \quad (6.1)$$

$$h(d)(s, t) \in \operatorname{argmin} \left\{ \max_{(\mu, \nu) \in R} \mathcal{K}(d)(\mu, \nu) \mid R \in \mathcal{R}(\delta(s), \delta(t)) \right\}. \quad (6.2)$$

for $d: S \times S \rightarrow [0, 1]$, $\mu, \nu \in \mathcal{D}(S)$, and $s, t \in S$.

The following lemma explains how the above ingredients can be used by the min player to improve its strategy.

Lemma 6.3. *Let $\mathcal{C} = (f, \rho)$. If there exist $s, t \in S$ such that $\Delta_\lambda(\gamma_\lambda^\mathcal{C})(s, t) < \gamma_\lambda^\mathcal{C}(s, t)$ then, $\gamma_\lambda^\mathcal{D} \sqsubset \gamma_\lambda^\mathcal{C}$ for a coupling structure $\mathcal{D} = (k(\gamma_\lambda^\mathcal{C}), \rho[(s, t)/R])$, where $R = h(\gamma_\lambda^\mathcal{C})(s, t)$.*

Proof. Assume $\Delta_\lambda(\gamma_\lambda^\mathcal{C})(s, t) < \gamma_\lambda^\mathcal{C}(s, t)$. Next we show $\Gamma_\lambda^\mathcal{D}(\gamma_\lambda^\mathcal{C}) \sqsubset \gamma_\lambda^\mathcal{C}$. In particular we prove that $\Gamma_\lambda^\mathcal{D}(\gamma_\lambda^\mathcal{C})(s, t) < \gamma_\lambda^\mathcal{C}(s, t)$ and, for all $(u, v) \neq (s, t)$, $\Gamma_\lambda^\mathcal{D}(\gamma_\lambda^\mathcal{C})(u, v) \leq \gamma_\lambda^\mathcal{C}(u, v)$.

By $\Delta_\lambda(\gamma_\lambda^\mathcal{C})(s, t) < \gamma_\lambda^\mathcal{C}(s, t)$, we necessarily have $\ell(s) = \ell(t)$. Thus

$$\begin{aligned} \Delta_\lambda(\gamma_\lambda^\mathcal{C})(s, t) &= \lambda \cdot \mathcal{H}(\mathcal{K}(\gamma_\lambda^\mathcal{C}))(\delta(s), \delta(t)) && (\ell(s) = \ell(t) \text{ and def. } \Delta_\lambda) \\ &= \lambda \cdot \min\left\{ \max_{(\mu, \nu) \in R'} \mathcal{K}(\gamma_\lambda^\mathcal{C})(\mu, \nu) \mid R' \in \mathcal{R}(\delta(s), \delta(t)) \right\} && (\text{Theorem 2.1}) \\ &= \lambda \cdot \max_{(\mu, \nu) \in R} \mathcal{K}(\gamma_\lambda^\mathcal{C})(\mu, \nu) && (R = h(\gamma_\lambda^\mathcal{C})(s, t) \text{ and (6.2)}) \\ &= \lambda \cdot \max_{(\mu, \nu) \in R} \min_{\omega \in \Omega(\mu, \nu)} \sum_{u, v \in S} \gamma_\lambda^\mathcal{C}(u, v) \cdot \omega(u, v) && (\text{def. } \mathcal{K}(\gamma_\lambda^\mathcal{C})) \\ &= \lambda \cdot \max_{(\mu, \nu) \in R} \sum_{u, v \in S} \gamma_\lambda^\mathcal{C}(u, v) \cdot k(\gamma_\lambda^\mathcal{C})(\mu, \nu)(u, v) && (\text{by (6.1)}) \\ &= \Gamma_\lambda^\mathcal{D}(\gamma_\lambda^\mathcal{C})(s, t). && (\text{def. } \mathcal{D} \text{ and } \Gamma_\lambda^\mathcal{D}) \end{aligned}$$

Therefore, $\Gamma_\lambda^\mathcal{D}(\gamma_\lambda^\mathcal{C})(s, t) = \Delta_\lambda(\gamma_\lambda^\mathcal{C})(s, t) < \gamma_\lambda^\mathcal{C}(s, t)$.

Let $u, v \in S$ such that $(u, v) \neq (s, t)$. We distinguish two cases.

- If $\ell(u) \neq \ell(v)$, then $\Gamma_\lambda^\mathcal{D}(\gamma_\lambda^\mathcal{C})(u, v) = 1 = \Gamma_\lambda^\mathcal{C}(\gamma_\lambda^\mathcal{C})(u, v) = \gamma_\lambda^\mathcal{C}(u, v)$.
- If $\ell(u) = \ell(v)$, then

$$\begin{aligned} \Gamma_\lambda^\mathcal{D}(\gamma_\lambda^\mathcal{C})(u, v) &= \lambda \cdot \max_{(\mu, \nu) \in \rho(u, v)} \sum_{x, y \in S} k(\gamma_\lambda^\mathcal{C})(\mu, \nu)(x, y) \cdot \gamma_\lambda^\mathcal{C}(x, y) && (\text{def. } \Gamma_\lambda^\mathcal{D} \text{ and } \mathcal{D}) \\ &\leq \lambda \cdot \max_{(\mu, \nu) \in \rho(u, v)} \sum_{x, y \in S} f(\mu, \nu)(x, y) \cdot \gamma_\lambda^\mathcal{C}(x, y) && ((6.1), f(\mu, \nu) \in \Omega(\mu, \nu)) \\ &= \Gamma_\lambda^\mathcal{C}(\gamma_\lambda^\mathcal{C})(u, v) = \gamma_\lambda^\mathcal{C}(u, v). && (\text{def. } \Gamma_\lambda^\mathcal{C} \text{ and } \gamma_\lambda^\mathcal{C}) \end{aligned}$$

Thus $\Gamma_\lambda^\mathcal{D}(\gamma_\lambda^\mathcal{C}) \sqsubset \gamma_\lambda^\mathcal{C}$. By Knaster-Tarski's fixed point theorem, we conclude that $\gamma_\lambda^\mathcal{D} \sqsubset \gamma_\lambda^\mathcal{C}$. \square

Lemma 6.3 suggests that $\mathcal{C} = (f, \rho)$ can be improved by replacing the measure-coupling map f with $k(\gamma_\lambda^\mathcal{C})$ and updating the set-coupling map ρ at (s, t) with $R = h(\gamma_\lambda^\mathcal{C})(s, t)$.

Note that a measure-coupling $k(d)(\mu, \nu)$ satisfying (6.1) can be computed by solving a linear program and ensuring that the optimal solution is a vertex of the polytope $\Omega(\mu, \nu)$ [Orl85, KS95]. A set-coupling $h(d)(s, t)$ satisfying (6.2) is the following:

$$R = \{(\mu, \phi(\mu)) \mid \mu \in \delta(s)\} \cup \{(\psi(\nu), \nu) \mid \nu \in \delta(t)\} \in \mathcal{R}(\delta(s), \delta(t)), \quad (6.3)$$

where ϕ, ψ are such that $\phi(\mu) \in \operatorname{argmin}_{\nu \in \delta(t)} \mathcal{K}(d)(\mu, \nu)$ and $\psi(\nu) \in \operatorname{argmin}_{\mu \in \delta(s)} \mathcal{K}(d)(\mu, \nu)$. The following lemma justifies our choice of $h(d)(s, t)$.

Lemma 6.4. *Let R be as in (6.3). Then $\mathcal{H}(\mathcal{K}(d))(\delta(s), \delta(t)) = \max_{(\mu, \nu) \in R} \mathcal{K}(d)(\mu, \nu)$.*

Proof. By Theorem 2.1 and $R \in \mathcal{R}(\delta(s), \delta(t))$, we have

$$\mathcal{H}(\mathcal{K}(d))(\delta(s), \delta(t)) \leq \max_{(\mu, \nu) \in R} \mathcal{K}(d)(\mu, \nu).$$

Hence, it suffices to prove

- (i) $\mathcal{K}(d)(\mu, \phi(\mu)) \leq \mathcal{H}(\mathcal{K}(d))(\delta(s), \delta(t))$, for all $\mu \in \delta(s)$, and
- (ii) $\mathcal{K}(d)(\psi(\nu), \nu) \leq \mathcal{H}(\mathcal{K}(d))(\delta(s), \delta(t))$, for all $\nu \in \delta(t)$.

Algorithm 1: Simple policy iteration algorithm computing \mathbf{d}_λ for $\lambda \in (0, 1)$.

```

1 Initialise  $\mathcal{C} = (f, \rho)$  as an arbitrary vertex coupling structure for  $\mathcal{A}$ 
2 while  $\exists(s, t). \Delta_\lambda(\gamma_\lambda^{\mathcal{C}})(s, t) < \gamma_\lambda^{\mathcal{C}}(s, t)$  do
3   |  $R \leftarrow h(\gamma_\lambda^{\mathcal{C}})(s, t)$ 
4   |  $\mathcal{C} \leftarrow (k(\gamma_\lambda^{\mathcal{C}}), \rho[(s, t)/R])$  /* update coupling structure */
5 end
6 return  $\gamma_\lambda^{\mathcal{C}}$  /*  $\gamma_\lambda^{\mathcal{C}} = \mathbf{d}_\lambda$  */
```

We prove (i). Let $\mu \in \delta(s)$. Then

$$\begin{aligned}
\mathcal{H}(\mathcal{K}(d))(\delta(s), \delta(t)) &\geq \max_{\mu' \in \delta(s)} \min_{\nu \in \delta(t)} \mathcal{K}(d)(\mu', \nu) && (\text{def. } \mathcal{H}) \\
&\geq \min_{\nu \in \delta(t)} \mathcal{K}(d)(\mu, \nu) && (\mu \in \delta(s)) \\
&= \mathcal{K}(d)(\mu, \phi(\mu)) && (\phi(\mu) \in \operatorname{argmin}_{\nu \in \delta(t)} \mathcal{K}(d)(\mu, \nu))
\end{aligned}$$

The proof for (ii) follows similarly. \square

Remark 6.5. The update procedure entailed by Lemma 6.3 can be performed in polynomial-time in the size of the probabilistic automaton \mathcal{A} . Indeed, $k(d)(\mu, \nu)$ can be obtained by solving a transportation problem in polynomial time [Orl85, KS95]. As for $h(d)(s, t)$, one can obtain $\phi(\mu)$ (resp. $\psi(\nu)$) by computing $\mathcal{K}(d)(\mu, \nu)$ in polynomial time and selecting the ν (resp. μ) ranging over $\delta(t)$ (resp. $\delta(s)$) that achieves the minimum.

6.2. Discounted case. The simple policy iteration algorithm for computing \mathbf{d}_λ in the case $\lambda < 1$ is presented in Algorithm 1. The procedure starts by computing an initial vertex coupling structure \mathcal{C}_0 (line 1), *e.g.*, by using the North-West corner method in polynomial time (see, *e.g.*, [Str89, pg. 180]). Then it continues by iteratively generating a sequence $\mathcal{C}_0, \mathcal{C}_1, \dots, \mathcal{C}_n$ of vertex coupling structures where $\mathbf{d}_\lambda = \gamma_\lambda^{\mathcal{C}_n}$. At each iteration, the current coupling structure \mathcal{C}_i is tested for optimality (line 2) by checking whether the corresponding λ -discrepancy $\gamma_\lambda^{\mathcal{C}_i}$ is a fixed point for Δ_λ . If there exists $(s, t) \in S$ violating the equality $\gamma_\lambda^{\mathcal{C}_i} = \Delta_\lambda(\gamma_\lambda^{\mathcal{C}_i})$, it constructs \mathcal{C}_{i+1} by updating \mathcal{C}_i at (s, t) as prescribed by Lemma 6.3 (line 4). This guarantees that $\gamma_\lambda^{\mathcal{C}_i} \sqsupset \gamma_\lambda^{\mathcal{C}_{i+1}}$, i.e., a strict improvement of the λ -discrepancy towards the minimal one.

Termination follows by the fact that there are only finitely many vertex coupling structures for \mathcal{A} . Furthermore, the correctness of the output of the algorithm is due to the fact that, Δ_λ has a unique fixed point when $0 \leq \lambda < 1$.

Theorem 6.6. *Let $\lambda \in (0, 1)$. Algorithm 1 terminates and computes \mathbf{d}_λ .*

Proof. First we prove termination. Note that the set

$$\{\gamma_\lambda^{\mathcal{C}} \mid \mathcal{C} \text{ vertex coupling structure for } \mathcal{A}\} \quad (6.4)$$

is finite because for all $s, t \in S$ the set $\mathcal{R}(\delta(s), \delta(t))$ is finite, and for all $\mu \in \delta(s)$ and $\nu \in \delta(t)$ the polytope $\Omega(\mu, \nu)$ has finitely many vertices, i.e., $V(\Omega(\mu, \nu))$ is finite. Towards a contradiction, assume that Algorithm 1 does not terminate. Let $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \dots$ be the infinite sequence of coupling structures generated during a non-terminating execution of Algorithm 1. Since the set in (6.4) is finite, there must be $i < j$ such that $\gamma_\lambda^{\mathcal{C}_i} = \gamma_\lambda^{\mathcal{C}_j}$.

On the contrary, next we prove that the updates of the coupling structures in Algorithm 1 ensure that for all $n \in \mathbb{N}$, $\gamma_\lambda^{C_n} \sqsupseteq \gamma_\lambda^{C_{n+1}}$. Let $n \in \mathbb{N}$. Since we are considering a non-terminating execution we have that $\Delta_\lambda(\gamma_\lambda^{C_n})(s, t) < \gamma_\lambda^{C_n}(s, t)$, for some $s, t \in S$. C_{n+1} is obtained from the update performed in line 4, which is exactly the one prescribed by Lemma 6.3. Therefore we have that $\gamma_\lambda^{C_n} \sqsupseteq \gamma_\lambda^{C_{n+1}}$. Hence, Algorithm 1 terminates.

When the execution of Algorithm 1 has reached the return statement, we have that $\Delta_\lambda(\gamma_\lambda^{C_n})(s, t) \geq \gamma_\lambda^{C_n}(s, t)$ for all $s, t \in S$, i.e., $\gamma_\lambda^{C_n} \sqsubseteq \Delta_\lambda(\gamma_\lambda^{C_n})$. By Lemma 5.4, $\gamma_\lambda^{C_n} \sqsupseteq \Delta_\lambda(\gamma_\lambda^{C_n})$, therefore $\gamma_\lambda^{C_n} = \Delta_\lambda(\gamma_\lambda^{C_n})$. The operator Δ_λ is λ -Lipschitz continuous [Tan18, Proposition 10.3.2(b)] thus, by Banach's fixed-point theorem, Δ_λ has a unique fixed point. Hence, $\gamma_\lambda^C = \mathbf{d}_\lambda$. \square

6.3. Undiscounted case. For $\lambda = 1$, the termination condition of the simple policy-iteration algorithm of Section 6.2 is not sufficient to guarantee correctness, since Algorithm 1 may terminate prematurely by returning a fixed point of Δ_1 that is not the minimal one.

Towards a way to obtain a stronger termination condition, we introduce the notion of self-closed relations w.r.t. a fixed point for Δ_1 , originally due to [Fu12].

Definition 6.7. A relation $M \subseteq S \times S$ is *self-closed w.r.t. $d = \Delta_1(d)$* if, whenever $s M t$,

- (i) $\ell(s) = \ell(t)$ and $d(s, t) > 0$,
- (ii) if $s \rightarrow \mu$ and $d(s, t) = \min_{\nu' \in \delta(t)} \mathcal{K}(d)(\mu, \nu')$ then there exists $t \rightarrow \nu$ and $\omega \in \Omega(\mu, \nu)$ such that $d(s, t) = \sum_{u, v \in S} d(u, v) \cdot \omega(u, v)$ and $\text{supp}(\omega) \subseteq M$,
- (iii) if $t \rightarrow \nu$ and $d(s, t) = \min_{\mu' \in \delta(s)} \mathcal{K}(d)(\mu', \nu)$ then there exists $s \rightarrow \mu$ and $\omega \in \Omega(\mu, \nu)$ such that $d(s, t) = \sum_{u, v \in S} d(u, v) \cdot \omega(u, v)$ and $\text{supp}(\omega) \subseteq M$.

Two states are self-closed w.r.t. d , written $s \approx_d t$, if they are related by some self-closed relation w.r.t. d .

It can be easily shown that \approx_d is the largest self-closed relation w.r.t. d . Note that the concept of self-closedness above is defined only for fixed points of Δ_1 . As remarked in [Fu12], the largest self-closed relation \approx_d can be computed in polynomial time by using partition refinement techniques similar to those employed to compute the largest bisimilarity relation.

Example 6.8. The following is a parametric definition for a family of fixed points of Δ_1 over the probabilistic automaton in Figure 2, for $\frac{1}{2} \leq \alpha \leq 1$:

$$\begin{aligned} d_\alpha(t, u) &= d_\alpha(u, t) = \alpha, \\ d_\alpha(t, t) &= d_\alpha(u, u) = d_\alpha(v, v) = 0, \\ d_\alpha(t, v) &= d_\alpha(u, v) = d_\alpha(v, t) = d_\alpha(v, u) = 1. \end{aligned}$$

For $\alpha \neq \frac{1}{2}$, an example of a self-closed relation w.r.t. d_α is given by the relation $M = \{(t, u)\}$. It is easy to verify that condition (i) holds true since $\ell(t) = \ell(u)$ and $d_\alpha(t, u) = \alpha > 0$. The only case where (ii) is non-trivial is when $t \rightarrow \mathbf{1}_t$, as $\min_{\nu' \in \delta(t)} \mathcal{K}(d_\alpha)(\mathbf{1}_t, \nu') = \mathcal{K}(d_\alpha)(\mathbf{1}_t, \mathbf{1}_u) = d_\alpha(t, u)$. In this case, condition (ii) is satisfied since $u \rightarrow \mathbf{1}_u$ and $\omega = \mathbf{1}_{(t, u)} \in \Omega(\mathbf{1}_t, \mathbf{1}_u)$ is a measure-coupling such that $d_\alpha(s, t) = \sum_{u, v} d_\alpha(u, v) \cdot \omega(u, v)$ and $\text{supp}(\omega) \subseteq M$. As for the last condition (iii), it holds true because the premise of the implication is never satisfied. With similar arguments one can easily verify that $M' = \{(t, u), (u, t)\}$ is also a self-closed set w.r.t. d_α , and in particular it is the greatest one.

For $\alpha = \frac{1}{2}$, we have that d_α is the least fixed points. The only admissible self-closed relation for it is the empty set.

The next lemma (Lemma 6.10) characterizes self-closed relation M in terms of the existence of certain optimal strategies for the probabilistic bisimilarity game given in Definition 4.5. Intuitively, M is a set of nodes such that, if you play optimally with respect to the current value d you still remain within M .

Example 6.9. As shown in Example 6.8, for $\frac{1}{2} < \alpha \leq 1$, $M = \{(t, u)\}$ is a self-closed relation w.r.t. d_α for the probabilistic automaton depicted in Figure 2. Consider now d_α as the current value for the probabilistic bisimilarity game associated with the automaton (see Figure 2(right)). An optimal strategy relative to the current value d_α is the one where the min player chooses R from (t, u) , and $\mathbf{1}_{(t, u)}$ from $(\mathbf{1}_t, \mathbf{1}_u)$; while the max player chooses $(\mathbf{1}_t, \mathbf{1}_u)$ from R . This particular choice of the strategy makes the two players stay within M , without ever reaching a sink state.

Lemma 6.10. *Given $d = \Delta_1(d)$, a relation $M \subseteq S \times S$ is self-closed w.r.t. d , if and only if, whenever $s M t$ then*

- (a) $\ell(s) = \ell(t)$ and $d(s, t) > 0$,
- (b) *there exists $R \in \mathcal{R}(\delta(s), \delta(t))$ such that $d(s, t) = \max_{(\mu', \nu') \in R} \mathcal{K}(d)(\mu', \nu')$ and for all $(\mu, \nu) \in R$ such that $d(s, t) = \mathcal{K}(d)(\mu, \nu)$, there exists $\omega \in \Omega(\mu, \nu)$ such that $d(s, t) = \sum_{u, v \in S} d(u, v) \cdot \omega(u, v)$ and $\text{supp}(\omega) \subseteq M$.*

Proof. (\Leftarrow) Let $s M t$. Def. 6.7.(i) follows from (a). Next we show that (b) implies Def. 6.7.(ii). Let R be the set-coupling for $(\delta(s), \delta(t))$ satisfying (b) and $\mu \in \delta(s)$ be such that $d(s, t) = \min_{\nu' \in \delta(t)} \mathcal{K}(d)(\mu, \nu')$. Then by definition of set-coupling, there exists $\nu \in \delta(t)$ such that $(\mu, \nu) \in R$. Clearly $\mathcal{K}(d)(\mu, \nu) \geq \min_{\nu' \in \delta(t)} \mathcal{K}(d)(\mu, \nu')$. Moreover,

$$\begin{aligned} \mathcal{K}(d)(\mu, \nu) &\leq \max_{(\mu', \nu') \in R} \mathcal{K}(d)(\mu', \nu') && ((\mu, \nu) \in R) \\ &= d(s, t) && \text{(by (b))} \\ &= \min_{\nu' \in \delta(t)} \mathcal{K}(d)(\mu, \nu'). && \text{(hp. on } \mu) \end{aligned}$$

Therefore, $\mathcal{K}(d)(\mu, \nu) = d(s, t)$. By $(\mu, \nu) \in R$ and (b), there exists $\omega \in \Omega(\mu, \nu)$ such that $d(s, t) = \sum_{u, v \in S} d(u, v) \cdot \omega(u, v)$ and $\text{supp}(\omega) \subseteq M$. Hence, Def. 6.7.(ii) holds true. Condition Def. 6.7.(iii) follows similarly.

(\Rightarrow) Let $s M t$. Condition (a) follows by Def. 6.7.(i). Assume Def. 6.7.(ii) and (iii) hold true. Then, we can define $\phi: \delta(s) \rightarrow \delta(t)$ and $\psi: \delta(t) \rightarrow \delta(s)$ such that

- (1) $\phi(\mu) \in \text{argmin}_{\nu' \in \delta(t)} \mathcal{K}(d)(\mu, \nu')$ and if $\mathcal{K}(d)(\mu, \phi(\mu)) = d(s, t)$ then, there exists $\omega \in \Omega(\mu, \phi(\mu))$ such that $\sum_{u, v \in S} d(u, v) \cdot \omega(u, v) = d(s, t)$ and $\text{supp}(\omega) \subseteq M$.
- (2) $\psi(\nu) \in \text{argmin}_{\mu' \in \delta(s)} \mathcal{K}(d)(\mu', \nu)$ and if $\mathcal{K}(d)(\psi(\nu), \nu) = d(s, t)$ then there exists $\omega \in \Omega(\psi(\nu), \nu)$ such that $\sum_{u, v \in S} d(u, v) \cdot \omega(u, v) = d(s, t)$ and $\text{supp}(\omega) \subseteq M$.

Define $R = \{(\mu, \phi(\mu)) \mid \mu \in \delta(s)\} \cup \{(\psi(\nu), \nu) \mid \nu \in \delta(t)\}$. By construction $R \in \mathcal{R}(\delta(s), \delta(t))$. Then, the following hold

$$\begin{aligned} \max_{(\mu, \nu) \in R} \mathcal{K}(d)(\mu, \nu) &= \max\left\{ \max_{\mu \in \delta(s)} \mathcal{K}(d)(\mu, \phi(\mu)), \max_{\nu \in \delta(t)} \mathcal{K}(d)(\psi(\nu), \nu) \right\} && \text{(def. } R) \\ &= \mathcal{H}(\mathcal{K}(d))(\delta(s), \delta(t)) && \text{(by def. } \phi \text{ and } \psi) \\ &= d(s, t). && (d = \Delta_1(d)) \end{aligned}$$

Let $(\mu, \nu) \in R$ such that $d(s, t) = \mathcal{K}(d)(\mu, \nu)$. By definition, $\nu = \phi(\mu)$ or $\mu = \psi(\nu)$. If $\nu = \phi(\mu)$ then, $d(s, t) = \mathcal{K}(d)(\mu, \phi(\mu))$ and, by ϕ , there exists $\omega \in \Omega(\mu, \phi(\mu))$ such that $\sum_{u, v \in S} d(u, v) \cdot \omega(u, v) = d(s, t)$ and $\text{supp}(\omega) \subseteq M$. The case $\mu = \psi(\nu)$ is analogous. \square

The next lemma states that if for a fixed point $d = \Delta_1(d)$ the relation \approx_d is nonempty, then d is not the least fixed point of Δ_1 .

Lemma 6.11. *Let $d = \Delta_1(d)$. If there exists a nonempty self-closed relation M w.r.t. d , then there exists $d_M \sqsubset d$ such that $\Delta_1(d_M) \sqsubseteq d_M$. Moreover, d_M can be computed in polynomial time in the size of the probabilistic automaton \mathcal{A} .*

Proof. Let M be a nonempty self-closed relation w.r.t. d . For arbitrary $s, t \in S$, $\mu \in \delta(s)$, and $\nu \in \delta(t)$, define

$$\theta_s(\mu, t) = d(s, t) - \min_{\nu \in \delta(t)} \mathcal{K}(d)(\mu, \nu) \quad \text{and} \quad \theta_t(s, \nu) = d(s, t) - \min_{\mu \in \delta(s)} \mathcal{K}(d)(\mu, \nu).$$

Note that, $\theta_s(\mu, t)$ and $\theta_t(s, \nu)$ are non-negative since $d = \Delta_1(d)$. Let $\theta = \min\{\theta_1, \theta_2, \theta_3\}$ where

- $\theta_1 = \min\{\theta_s(\mu, t) \mid (s, t) \in M \wedge \mu \in \delta(s) \wedge \theta_s(\mu, t) > 0\}$;
- $\theta_2 = \min\{\theta_t(s, \nu) \mid (s, t) \in M \wedge \nu \in \delta(t) \wedge \theta_t(s, \nu) > 0\}$;
- $\theta_3 = \min\{d(s, t) \mid (s, t) \in M\}$;

where $\min \emptyset = 1$. Note that $\theta_3 > 0$, because M is a nonempty self-closed relation w.r.t. d . Therefore $\theta > 0$. We define the map $d_M: S \times S \rightarrow [0, 1]$ as

$$d_M(s, t) = \begin{cases} d(s, t) - \theta & \text{if } (s, t) \in M \\ d(s, t) & \text{if } (s, t) \notin M \end{cases}$$

It is clear that d_M is well-defined. Moreover $d_M \sqsubset d$ because M is nonempty and $\theta > 0$.

Next we prove that $\Delta_1(d_M) \sqsubseteq d_M$. Let $s, t \in S$. We consider two cases:

- Assume $(s, t) \notin M$. Then

$$\begin{aligned} \Delta_1(d_M)(s, t) &\leq \Delta_1(d)(s, t) && \text{(by } d_M \sqsubseteq d \text{ and } \Delta_1 \text{ monotone)} \\ &= d(s, t) && (d = \Delta_1(d)) \\ &= d_M(s, t) && ((s, t) \notin M) \end{aligned}$$

- Assume $(s, t) \in M$. Then $\ell(s) = \ell(t)$. Let $\mu \in \delta(s)$. We consider two subcases below:

- (1) If $\theta_s(\mu, t) > 0$ we have

$$\begin{aligned} d_M(s, t) &= d(s, t) - \theta && \text{(def. } d_M) \\ &\geq d(s, t) - \theta_s(\mu, t) && (0 < \theta \leq \theta_s(\mu, t)) \\ &= \min_{\nu \in \delta(t)} \mathcal{K}(d)(\mu, \nu) && \text{(def. } \theta_s(\mu, t)) \\ &\geq \min_{\nu \in \delta(t)} \mathcal{K}(d_M)(\mu, \nu) && (d_M \sqsubseteq d \text{ and } \mathcal{K} \text{ monotone)} \end{aligned}$$

- (2) If $\theta_s(\mu, t) = 0$, then $d(s, t) = \min_{\nu \in \delta(t)} \mathcal{K}(d)(\mu, \nu)$. Since M is self-closed w.r.t. d , there exists $\nu' \in \delta(t)$ such that $d(s, t) = \sum_{u, v \in S} d(u, v) \cdot \omega(u, v)$, for some $\omega \in \Omega(\mu, \nu')$ such

that $\text{supp}(\omega) \subseteq M$. Thus

$$\begin{aligned}
\min_{\nu \in \delta(t)} \mathcal{K}(d_M)(\mu, \nu) &\leq \mathcal{K}(d_M)(\mu, \nu') && (\nu' \in \delta(t)) \\
&= \min_{\omega' \in \Omega(\mu, \nu')} \sum_{u, v \in S} d_M(u, v) \cdot \omega'(u, v) && (\text{def. } \mathcal{K}(d_M)) \\
&\leq \sum_{u, v \in S} d_M(u, v) \cdot \omega(u, v) && (\omega \in \Omega(\mu, \nu')) \\
&= \sum_{u, v \in M} d_M(u, v) \cdot \omega(u, v) && (\text{supp}(\omega) \subseteq M) \\
&= \sum_{u, v \in M} (d(u, v) - \theta) \cdot \omega(u, v) && (\text{def. } d_M) \\
&= \left(\sum_{u, v \in M} d(u, v) \cdot \omega(u, v) \right) - \theta && (\sum_{u, v \in M} \omega(u, v) = 1) \\
&= d(s, t) - \theta && (d(s, t) = \sum_{u, v \in S} d(u, v) \cdot \omega(u, v)) \\
&= d_M(s, t) && (\text{def. } d_M)
\end{aligned}$$

So that, in both cases 1 and 2 we have $d_M(s, t) \geq \min_{\nu \in \delta(t)} \mathcal{K}(d_M)(\mu, \nu)$. Since this inequality holds for all $\mu \in \delta(s)$, we have $d_M(s, t) \geq \max_{\mu \in \delta(s)} \min_{\nu \in \delta(t)} \mathcal{K}(d_M)(\mu, \nu)$. Symmetrically, we can prove $d_M(s, t) \geq \max_{\nu \in \delta(t)} \min_{\mu \in \delta(s)} \mathcal{K}(d_M)(\mu, \nu)$. Thus, by definition of Hausdorff lifting, $d_M(s, t) \geq \mathcal{H}(\mathcal{K}(d_M))(\delta(s), \delta(t))$. From this we conclude

$$\begin{aligned}
d_M(s, t) &\geq \mathcal{H}(\mathcal{K}(d_M))(\delta(s), \delta(t)) \\
&= \Delta_1(d_M)(s, t) && (\ell(s) = \ell(t) \text{ and def. } \Delta_1)
\end{aligned}$$

Finally, we consider the complexity of computing d_M . For computing θ , we need to compute in turn θ_1 , θ_2 , and θ_3 . Since $M \subseteq S \times S$, computing θ_3 can be done in quadratic time in $|S|$. The computation of θ_1 requires at most $|M| \cdot \sum_{s \in S} |\delta(s)|$ solutions of a transportation problem. This can be done in polynomial-time in the size of \mathcal{A} . Similarly for θ_2 . \square

The proof of Lemma 6.11 is essentially that of [Fu12, Theorem 3]. Given a nonempty self-closed relation M w.r.t. d , the above result can be used to obtain a prefix point of Δ_1 , namely d_M , that improves d towards the search of the least fixed point. The prefix point d_M of Lemma 6.11 is obtained from d by subtracting a suitable value $\theta > 0$ from all the distances computed at pairs of states in M :

$$d_M(s, t) = \begin{cases} d(s, t) - \theta & \text{if } (s, t) \in M, \\ d(s, t) & \text{if } (s, t) \notin M. \end{cases}$$

The value of θ that gives us the smallest prefix point defined as above, is the maximal value satisfying the following inequalities

$$\begin{aligned}
\theta &\leq d(s, t) - \min_{\nu' \in \delta(t)} \mathcal{K}(d)(\mu, \nu') && \text{for all } (s, t) \in M \text{ and } \mu \in \delta(s), \\
\theta &\leq d(s, t) - \min_{\mu' \in \delta(s)} \mathcal{K}(d)(\mu', \nu) && \text{for all } (s, t) \in M \text{ and } \nu \in \delta(t), \\
\theta &\leq d(s, t) && \text{for all } (s, t) \in M.
\end{aligned}$$

The fact that d_M is a prefix point follows by the fact that M is a self-closed relation.

The following lemma provides us with a termination condition for the simple policy iteration algorithm to compute \mathbf{d}_1 . Indeed, according to it, if d is a fixed point of Δ_1 , we can assert that d is equal to bisimilarity distance \mathbf{d}_1 by simply checking that the maximal self-closed relation w.r.t. d is empty.

Lemma 6.12. *Let $d = \Delta_1(d)$. If $\approx_d = \emptyset$, then $d = \mathbf{d}_1$.*

Proof. Let $d = \Delta_1(d)$. We proceed by contraposition. Assume that $d \neq \mathbf{d}_1$. We define a non-empty self-closed relation M w.r.t. d as follows.

$$m = \max_{s,t \in S} d(s,t) - \mathbf{d}_1(s,t), \quad M = \{(s,t) \in S \times S \mid d(s,t) - \mathbf{d}_1(s,t) = m\}.$$

Clearly, $m > 0$ and $M \neq \emptyset$ because $d \neq \mathbf{d}_1$.

Let $(s,t) \in M$. We prove that the three conditions of Definition 6.7 hold true.

- (1) $d(s,t) > 0$ because $0 < m = d(s,t) - \mathbf{d}_1(s,t) \leq d(s,t)$. Now we prove that $\ell(s) = \ell(t)$. Towards a contradiction, assume $\ell(s) \neq \ell(t)$. Then, the following inequalities hold

$$0 < m = d(s,t) - \mathbf{d}_1(s,t) = \Delta_1(d)(s,t) - \Delta_1(\mathbf{d}_1)(s,t) = 1 - 1 = 0,$$

leading to the contradiction that $0 < 0$.

- (2) Let $\mu \in \delta(s)$ such that $d(s,t) = \min_{\nu \in \delta(t)} \mathcal{K}(d)(\mu, \nu)$. Then we have

$$\begin{aligned} \mathbf{d}_1(s,t) &= \Delta_1(\mathbf{d}_1)(s,t) && \text{(by def. } \mathbf{d}_1) \\ &= \mathcal{H}(\mathcal{K}(\mathbf{d}_1))(\delta(s), \delta(t)) && \text{(by } \ell(s) = \ell(t)) \\ &\geq \min_{\nu \in \delta(t)} \mathcal{K}(\mathbf{d}_1)(\mu, \nu) && \text{(by def. } \mathcal{H}) \end{aligned}$$

Let $\nu^* \in \delta(t)$, $\omega \in \Omega(\mu, \nu^*)$ such that

$$\min_{\nu \in \delta(t)} \mathcal{K}(\mathbf{d}_1)(\mu, \nu) = \mathcal{K}(\mathbf{d}_1)(\mu, \nu^*) = \sum_{u,v \in S} \mathbf{d}_1(u,v) \cdot \omega(u,v). \quad (6.5)$$

Then, the following inequalities hold

$$\begin{aligned} \mathcal{K}(\mathbf{d}_1)(\mu, \nu^*) &= \sum_{u,v \in S} \mathbf{d}_1(u,v) \cdot \omega(u,v) && \text{(by (6.5))} \\ &= \sum_{u,v \in S} (d(u,v) - (d(u,v) - \mathbf{d}_1(u,v))) \cdot \omega(u,v) \\ &\geq \sum_{u,v \in S} (d(u,v) - m) \cdot \omega(u,v) && \text{(def. } m) \\ &= \left(\sum_{u,v \in S} d(u,v) \cdot \omega(u,v) \right) - m && (\omega \in \mathcal{D}(S \times S)) \\ &\geq \mathcal{K}(d)(\mu, \nu^*) - m && \text{(def. } \mathcal{K} \text{ and } \omega \in \Omega(\mu, \nu^*)) \end{aligned}$$

Thus, we have

$$\begin{aligned} d(s,t) &\leq \mathcal{K}(d)(\mu, \nu^*) && (d(s,t) = \min_{\nu \in \delta(t)} \mathcal{K}(d)(\mu, \nu)) \\ &\leq \mathcal{K}(\mathbf{d}_1)(\mu, \nu^*) + m && (\mathcal{K}(\mathbf{d}_1)(\mu, \nu^*) \geq \mathcal{K}(d)(\mu, \nu^*) - m) \\ &\leq \mathbf{d}_1(s,t) + m && (\mathbf{d}_1(s,t) \geq \min_{\nu \in \delta(t)} \mathcal{K}(\mathbf{d}_1)(\mu, \nu) \text{ and def. } \nu^*) \\ &= \mathbf{d}_1(s,t) + (d(s,t) - \mathbf{d}_1(s,t)) && \text{(def. } m \text{ and } (s,t) \in M) \\ &= d(s,t) \end{aligned}$$

Therefore, all the above inequalities are, in fact, equalities. Hence,

$$d(s,t) = \mathcal{K}(d)(\mu, \nu^*) \quad \text{and} \quad \mathbf{d}_1(s,t) = \mathcal{K}(\mathbf{d}_1)(\mu, \nu^*). \quad (6.6)$$

We conclude by proving that ω satisfies the following

$$d(s,t) = \sum_{u,v \in S} d(u,v) \cdot \omega(u,v) \quad \text{and} \quad \text{supp}(\omega) \subseteq M.$$

Algorithm 2: Simple policy iteration algorithm computing \mathbf{d}_1 .

```

1 Initialise  $\mathcal{C} = (f, \rho)$  as an arbitrary vertex coupling structure for  $\mathcal{A}$ 
2 ISMIN  $\leftarrow false$ 
3 while  $\neg$ ISMIN do
4   while  $\exists(s, t). \Delta_1(\gamma_1^{\mathcal{C}})(s, t) < \gamma_1^{\mathcal{C}}(s, t)$  do
5      $R \leftarrow h(\gamma_1^{\mathcal{C}})(s, t)$ 
6      $\mathcal{C} \leftarrow (k(\gamma_1^{\mathcal{C}}), \rho[(s, t)/R])$            /* update coupling structure */
7   end
8   Let  $M \leftarrow \approx_{\gamma_1^{\mathcal{C}}}$            /* note that  $\gamma_1^{\mathcal{C}} = \Delta_1(\gamma_1^{\mathcal{C}})$  */
9   if  $M = \emptyset$  then
10    ISMIN  $\leftarrow true$            /*  $\gamma_1^{\mathcal{C}} = \mathbf{d}_1$  */
11  else
12    Compute  $d = (\gamma_1^{\mathcal{C}})_M$  as in Lemma 6.11
13    Re-initialise  $\mathcal{C}$  as a vertex coupling structure s.t.  $\Gamma_1^{\mathcal{C}}(d) = \Delta_1(d)$ 
14  end
15 end
16 return  $\gamma_1^{\mathcal{C}}$ 

```

This can be observed as follows

$$\begin{aligned}
& \sum_{u,v \in S} d(u, v) \cdot \omega(u, v) \geq d(s, t) && (\omega \in \Omega(\mu, \nu^*) \text{ and (6.6)}) \\
& = \mathbf{d}_1(s, t) + m && ((s, t) \in M \text{ and def. } m) \\
& = (\sum_{u,v \in S} \mathbf{d}_1(u, v) \cdot \omega(u, v)) + m && (\text{by (6.6) and (6.5)}) \\
& = \sum_{u,v \in S} (\mathbf{d}_1(u, v) + m) \cdot \omega(u, v) && (\omega \in \mathcal{D}(S \times S)) \\
& \geq \sum_{u,v \in S} (\mathbf{d}_1(u, v) + (d(u, v) - \mathbf{d}_1(u, v))) \cdot \omega(u, v) && (\text{def. } m) \\
& = \sum_{u,v \in S} d(u, v) \cdot \omega(u, v)
\end{aligned}$$

Hence, the above are in fact equalities and in particular $d(s, t) = \sum_{u,v \in S} d(u, v) \cdot \omega(u, v)$.

Consider now the following inequalities

$$\begin{aligned}
m &= d(s, t) - \mathbf{d}_1(s, t) && ((s, t) \in M) \\
&= d(s, t) - \sum_{u,v \in S} \mathbf{d}_1(u, v) \cdot \omega(u, v) && (\text{by (6.6) and (6.5)}) \\
&= \sum_{u,v \in S} (d(u, v) - \mathbf{d}_1(u, v)) \cdot \omega(u, v) && (d(s, t) = \sum_{u,v \in S} d(u, v) \cdot \omega(u, v))
\end{aligned}$$

Since $d(u, v) - \mathbf{d}_1(u, v) \leq m$ for all $u, v \in S$, the above equalities imply that whenever $\omega(u, v) > 0$ then $d(u, v) - \mathbf{d}_1(u, v) = m$. Therefore $\text{supp}(\omega) \subseteq M$.

(3) Can be argued symmetrically to the previous case.

Therefore, M is a nonempty self-closed relation with respect to d . \square

Algorithm 2 extends the procedure described in Section 6.2 by encapsulating the policy iteration update (lines 4–7) into an outer-loop (lines 3–15) that is responsible to check whether the fixed point $\gamma_1^{\mathcal{C}_i}$ returned is the minimal one. According to Lemma 5.4, $\Delta_1(\gamma_1^{\mathcal{C}_i}) \sqsubseteq \gamma_1^{\mathcal{C}_i}$. Hence, when we reach line 8, we have that $\Delta_1(\gamma_1^{\mathcal{C}_i}) = \gamma_1^{\mathcal{C}_i}$. Therefore, by Lemmas 6.11 and 6.12, $\gamma_1^{\mathcal{C}_i} = \mathbf{d}_1$ if and only if $M = \approx_{\gamma_1^{\mathcal{C}_i}}$ is empty. If M is empty, we set the variable ISMIN

to *true* (line 10) causing the outer-loop to terminate. Otherwise, we construct $d = (\gamma_1^{\mathcal{C}_i})_M$ as in Lemma 6.11 (line 12) and re-start the inner-loop from a vertex coupling structure \mathcal{C}_{i+1} such that $\Gamma_1^{\mathcal{C}_{i+1}}(d) = \Delta_1(d)$ (line 13) (e.g., by using $\mathcal{C}_{i+1} = (k(d), \rho)$ where $\rho(s, t) = h(d)(s, t)$ for all $s, t \in S$). As proven in Theorem 6.13, $\gamma_1^{\mathcal{C}_i} \sqsupset \gamma_1^{\mathcal{C}_{i+1}}$. This guarantees a strict improvement of the discrepancy towards the minimal one. Termination of Algorithm 2 is justified by similar arguments as for the discounted case.

Theorem 6.13. *Algorithm 2 terminates and computes \mathbf{d}_1 .*

Proof. First we prove termination. Recall that $\{\gamma_1^{\mathcal{C}} \mid \mathcal{C} \text{ vertex coupling structure for } \mathcal{A}\}$ is finite. Towards a contradiction, assume that Algorithm 2 does not terminate. Let $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_2, \dots$ be the infinite sequence of coupling structures updates generated during a non-terminating execution of Algorithm 2. Since the set above is finite, there must be $i < j$ such that $\gamma_1^{\mathcal{C}_i} = \gamma_1^{\mathcal{C}_j}$. On the contrary, next we prove that the updates of the coupling structures in Algorithm 2 ensure that for all $n \in \mathbb{N}$, $\gamma_1^{\mathcal{C}_n} \sqsupset \gamma_1^{\mathcal{C}_{n+1}}$. Let $n \in \mathbb{N}$. We consider two cases:

- Assume $\Delta_1(\gamma_1^{\mathcal{C}_n})(s, t) < \gamma_1^{\mathcal{C}_n}(s, t)$, for some $s, t \in S$. In this case \mathcal{C}_{n+1} is obtained from the update performed in line 6, which is exactly the one prescribed by Lemma 6.3. Therefore we have that $\gamma_1^{\mathcal{C}_n} \sqsupset \gamma_1^{\mathcal{C}_{n+1}}$.
- Assume $\Delta_1(\gamma_1^{\mathcal{C}_n})(s, t) \geq \gamma_1^{\mathcal{C}_n}(s, t)$ for all $s, t \in S$, i.e., $\gamma_1^{\mathcal{C}_n} \sqsubseteq \Delta_1(\gamma_1^{\mathcal{C}_n})$. By Lemma 5.4 $\gamma_1^{\mathcal{C}_n} \sqsupseteq \Delta_1(\gamma_1^{\mathcal{C}_n})$, therefore $\gamma_1^{\mathcal{C}_n} = \Delta_1(\gamma_1^{\mathcal{C}_n})$. In this case \mathcal{C}_{n+1} is constructed as a vertex coupling structure such that $\Gamma_1^{\mathcal{C}_{n+1}}(d) = \Delta_1(d)$ where $M = \approx_{\gamma_1^{\mathcal{C}_n}} \neq \emptyset$ and $d = (\gamma_1^{\mathcal{C}_n})_M$ (see line 13). Then the following inequalities hold

$$\begin{aligned}
 \gamma_1^{\mathcal{C}_n} \sqsupset d &\sqsupseteq \Delta_1(d) && \text{(Lemma 6.11)} \\
 &= \Gamma_1^{\mathcal{C}_{n+1}}(d) && \text{(by construction)} \\
 &\sqsupseteq \gamma_1^{\mathcal{C}_{n+1}} && \text{(by } \Gamma_1^{\mathcal{C}_{n+1}}(d) \sqsubseteq d \text{ and Knaster-Tarski fixed point theorem)}
 \end{aligned}$$

This concludes the proof that $\gamma_1^{\mathcal{C}_n} \sqsupset \gamma_1^{\mathcal{C}_{n+1}}$ for all $n \in \mathbb{N}$.

When the execution of Algorithm 2 has reached line 10 we have that $\gamma_1^{\mathcal{C}} = \Delta_1(\gamma_1^{\mathcal{C}})$. Moreover, we have $\approx_{\gamma_1^{\mathcal{C}}} = \emptyset$. Therefore, by Lemma 6.12 we have that $\gamma_1^{\mathcal{C}} = \mathbf{d}_1$. From here ISMIN is set to *true*. This prevents further executions of the body of the outer-loop. Therefore Algorithm 2 reached the return statement with $\gamma_1^{\mathcal{C}} = \mathbf{d}_1$. \square

6.4. Experimental Results. In this section, we evaluate the performance of the simple policy iteration algorithms on a collection of randomly generated probabilistic automata. All the algorithms have been implemented in Java and the source code is publicly available³.

The performance of Algorithm 1 has been compared with an implementation of the *value iteration algorithm* proposed by Fu [Fu12, Section 4]. This algorithm works as follows. Starting from the bottom element, it iteratively applies Δ_λ to the current distance function generating the increasing chain $\mathbf{0} \sqsubseteq \Delta_\lambda(\mathbf{0}) \sqsubseteq \Delta_\lambda^2(\mathbf{0}) \sqsubseteq \dots \sqsubseteq \Delta_\lambda^{k-1}(\mathbf{0}) \sqsubseteq \Delta_\lambda^k(\mathbf{0})$.

For each input instance, the comparison involves the following steps:

³<https://bitbucket.org/discoveri/probabilistic-bisimilarity-distances-probabilistic-automata>.

$n = S $	Simple Policy Iteration			Value Iteration			Error
	time (sec)	# TP	# \mathcal{C}	time (sec)	# TP	# Iter	
10	0.122	360.9	25.0	0.138	607.1	5.8	0.03018
11	0.167	457.2	30.5	0.189	779.5	6.0	0.03090
12	0.238	565.9	37.4	0.265	976.0	6.3	0.02814
13	0.309	679.3	44.2	0.351	1177.7	6.4	0.04198
14	0.412	813.4	52.2	0.463	1443.0	6.7	0.03673
15	0.569	963.3	61.1	0.634	1764.0	6.9	0.03790
20	2.694	1881.4	113.4	2.874	3325.5	7.6	0.03781
30	15.832	4629.1	263.6	16.642	10309.5	9.9	0.02615
40	99.433	17812.6	710.8	104.036	30870.3	8.8	0.01162
50	137.624	13985.4	753.0	144.098	35118.3	12.3	0.00975

Table 1: Comparison between Simple Policy and Value Iteration Algorithm. Average performance conducted on 100 randomly generated automata with number of states $n = 10..50$, nondeterministic out-degree $k = 1..3$, and probabilistic out-degree $p = 2..3$. Discount $\lambda = 0.8$; accuracy 0.000001.

- (1) We run Algorithm 1, storing execution time, the number of solved transportation problems, and the number of coupling structures generated during the execution (i.e., the number of times a λ -discrepancy has been computed);
- (2) Then, on the same instance, we execute the value iteration algorithm until the running time exceeds that of step 1. We report the execution time, the number of solved transportation problems, and the number of iterations.
- (3) Finally, we report the error $\max_{s,t \in S} |\mathbf{d}_\lambda(s,t) - d(s,t)|$ between the distance \mathbf{d}_λ computed in step 1 and the approximate result d obtained in step 2.

This has been done for a collection of automata varying from 10 to 50 states. For each $n = 10, \dots, 50$, we considered 100 randomly generated probabilistic automata, varying probabilistic out-degree and nondeterministic out-degree. Table 1 reports the average results of the comparison. Our algorithm is able to compute the solution before value iteration can under-approximate it with an error ranging from 0.004 to 0.06 which is a non negligible error considering that we fixed $\lambda = 0.8$ and the distance has values in $[0, 1]$.

Furthermore in Figure 3 we observe that the execution time of the simple policy iteration algorithm is particularly influenced by the degree of nondeterminism of the automaton. This may be explained by the fact that the current implementation uses a linear program for computing the λ -discrepancy (*cf.* Remark 5.3) which has $O(n^2k^2)$ variables and $O(n^2k^2)$ constraints where n and k are the number of states and the nondeterministic out-degree of the automaton, respectively.

Algorithm 2 extends the simple policy iteration algorithms proposed in [BBLM13, TvB16] for labelled Markov chains. As pointed out in [Tan18], implementations based on the decision procedure for the existential fragment of the first-order theory of the reals struggle to handle labelled Markov chains with a fifty states. For probabilistic automata, the algorithms in [CdAMR08, CdAMR10] suffer from the same problem. The performance of Algorithm 2 is comparable to that of Algorithm 1 (*cf.* Table 2). Despite the fact that the simple policy algorithm is not guaranteed to be sound when the discount factor equals one,

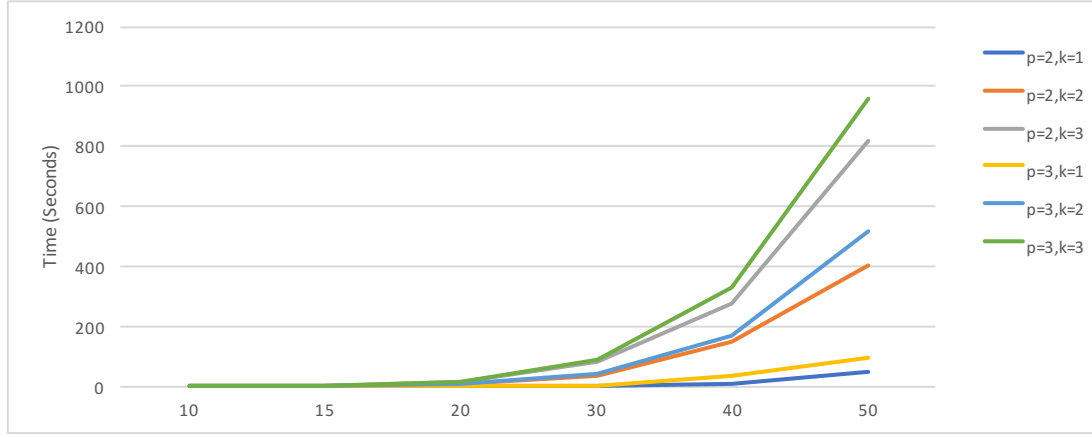


Figure 3: Average performance for the Simple Policy Iteration Algorithm conducted on 100 randomly generated automata varying number of states $n = 10..50$, nondeterministic out-degree $k = 1..3$, and probabilistic out-degree $p = 2..3$. Discount factor $\lambda = 0.8$; accuracy 0.000001.

$n = S $	Simple Policy Iteration				Value Iteration			Error
	time (sec)	# TP	# C	# Iter	time (sec)	# TP	# Iter	
10	0.129	394.8	25.9	1	0.144	647.4	6.2	0.07726
11	0.179	513.3	32.1	1	0.202	837.1	6.4	0.08236
12	0.263	655.1	39.1	1	0.293	1080.1	6.8	0.08988
13	0.352	815.2	46.6	1	0.394	1310.5	7.0	0.10222
14	0.465	966.6	53.9	1	0.514	1615.5	7.3	0.11156
15	0.703	1159.5	62.5	1	0.786	2042.7	7.7	0.12045
20	3.044	2291.7	111.8	1	3.316	3845.2	8.7	0.13823
30	15.905	5088.6	223.1	1	16.929	9956.3	10.2	0.16030
40	44.354	8597.3	364.0	1	47.580	16527.8	10.5	0.18830
50	113.586	13484.7	545.0	1	121.639	28694.5	11.7	0.20939

Table 2: Average performance of Algorithm 2 conducted on 100 randomly generated automata with number of states $n = 10..50$, nondeterministic out-degree $k = 1..3$, and probabilistic out-degree $p = 2..3$. Discount $\lambda = 1$; accuracy 0.000001.

our experiments show that in practice a single iteration of the outer-loop of Algorithm 2 is often sufficient to yield the correct solution, although it is still not clear to us how to precisely characterise the conditions under which this situation happens.

Remark 6.14 (Worst-case Time Complexity). Tang and van Breugel [TvB16, Tan18] showed that in the worst case, the simple policy iteration algorithm takes exponential time even with automata that are purely probabilistic (i.e., labelled Markov chains). The addition of nondeterminism also contributes in the exponential growth of the running time (*cf.* Figure 3).

7. RELATION WITH PROBABILISTIC MODEL CHECKING

In this section we show how the probabilistic bisimilarity distance of Deng et al. relates to the problem of model checking ω -regular specifications against probabilistic automata, where the nondeterministic choices are resolved by randomized schedulers.

Probabilistic automata are used for the verification of concurrent probabilistic systems, where the choice of how to interleave the executions of the parallel components is modelled by means of nondeterminism in the choice of the next transition to be taken. Technically, an execution of a probabilistic automaton $\mathcal{A} = (S, L, \rightarrow, \ell)$ is an infinite sequence $s_0 s_1 \dots \in S^\omega$ of labelled states obtained by taking a succession of probabilistic transitions $s_i \rightarrow \mu_i$ such that $\mu_i(s_{i+1}) > 0$, for each $i \in \mathbb{N}$. The choice of the transition to be taken at each state is resolved by means of a scheduler. In this paper we consider *randomized schedulers*, i.e., functions $\pi: S^+ \rightarrow \mathcal{D}(S)$ mapping a nonempty and finite sequence of states $s_0 \dots s_n \in S^+$ (the execution history) to a convex combination of distributions of the form $\sum_{s_n \rightarrow \mu} \alpha_\mu \cdot \mu \in \mathcal{D}(S)$, for some $\alpha_\mu \in [0, 1]$ such that $\sum_{s_n \rightarrow \mu} \alpha_\mu = 1$. Roughly, a randomized scheduler decides the probability with which the next transition is chosen given the history of visited states.

The combination of a probabilistic automata \mathcal{A} with a scheduler π induces a Markov chain on a random variable $X = (X_0, X_1, \dots) \in S^\omega$ on the measurable space of infinite sequences over S with distribution

$$\Pr_s^\pi(X_0 = s_0, \dots, X_n = s_n) = \mathbf{1}_s(s_0) \cdot \prod_{i=0}^{n-1} \pi(s_0 \dots s_i)(s_{i+1}),$$

where $\mathbf{1}_x$ denotes the Dirac distribution concentrated at x . The above describes the probability of executing the sequence of steps $s_0 \dots s_n$ by starting from the state s under the randomized scheduler π .

The measurable sets of S^ω are the elements of the infinite product σ -algebra $(2^S)^\omega$, i.e., the smallest σ -algebra containing the subsets of the form $s_0 \dots s_n S^\omega = \{s_0 \dots s_n w \mid w \in S^\omega\}$ (a.k.a., *discrete cylinders*), for arbitrary $n \in \mathbb{N}$, $s_i \in S$ and $0 \leq i \leq n$. Measurable sets are the subsets of sequences where the probability measure \Pr_s^π is well-defined. For a measurable set $H \subseteq S^\omega$, we denote by $\Pr_s^\pi(X \in H)$ the probability that an execution starting from s under the scheduler π belongs to H .

Rather than measuring the probability of concrete executions over S^ω , one is often more interested in the probability that certain execution traces satisfy abstract properties over the measurable space L^ω of infinite sequences of labels, representing the sequences of atomic properties satisfied by concrete executions of the automaton.

Formally, for a measurable set $E \subseteq L^\omega$, we denote by $\Pr_s^\pi(\ell(X) \in E)$ the probability that an execution generates a sequence of labels in E , where $\ell(X) = (\ell(X_0), \ell(X_1), \dots) \in L^\omega$ is the random variable induced from X by the labelling function ℓ . The σ -algebra of L^ω contains all the ω -regular languages expressible over the alphabet L [BK08, Chapter 10]. This means that the probability of the runs of \mathcal{A} of satisfying ω -regular properties, possibly expressed in the form of LTL formulas, can be formally measured by \Pr_s^π , hence allowing the quantitative analysis of probabilistic automata.

The quantitative analysis of a probabilistic automaton \mathcal{A} against ω -regular specifications, more commonly known as *probabilistic model-checking*, amounts to establishing the maximal and minimal probability of satisfying ω -regular properties $E \subseteq L^\omega$ over infinite sequences of

labels from a starting state s . Formally, this corresponds to computing

$$\text{Max}_s(E) = \sup_{\pi \in \Pi} \Pr_s^\pi(\ell(X) \in E) \quad \text{and} \quad \text{Min}_s(E) = \inf_{\pi \in \Pi} \Pr_s^\pi(\ell(X) \in E)$$

where the infimum and supremum are taken over the set Π of all randomized schedulers. Note that, considering minimal or maximal probabilities corresponds to a worst/best-case analysis (see [BK08, Chapter 10] for more details).

The following is the main result of the section. It states that the probabilistic bisimilarity distance bounds the difference between maximal and minimal probability of satisfying any measurable linear-time property (*e.g.*, ω -regular specifications) on two given initial states.

Theorem 7.1. *For all measurable subsets $E \subseteq L^\omega$,*

$$|\text{Max}_s(E) - \text{Max}_t(E)| \leq \mathbf{d}_1(s, t) \quad \text{and} \quad |\text{Min}_s(E) - \text{Min}_t(E)| \leq \mathbf{d}_1(s, t).$$

The above can be seen as a quantitative generalization of the folklore result that bisimilar states satisfy the same linear-time properties with the same probability.

Remark 7.2. The relevance of Theorem 7.1 is not just theoretical, but could possibly lead to significant practical applications. Imagine that the distance $\mathbf{d}_1(s, t)$ between some given states s and t is small (and known); then, computing $\text{Max}_s(E)$ (resp. $\text{Min}_s(E)$) in the state s may be enough for obtaining a good approximation for the actual value of $\text{Max}_t(E)$ (resp. $\text{Min}_t(E)$) without the need of computing it on the state t . This approach may lead to savings in the overall model-checking time of t , especially if the executions starting from s have a significant reduced degree of nondeterminism than those starting from t .

The proof of Theorem 7.1 is based on the coupling characterisation of the bisimilarity distance presented in Theorem 5.7 and the following technical lemma (Lemma 7.3) which establishes under which conditions the discrepancy associated with a coupling structure can be used to bound the variational distance between the probability induced by a probabilistic automaton \mathcal{A} under two different schedulers. Specifically, we establish how, from a coupling structure \mathcal{C} , one can retrieve a set-coupling $R_{\mathcal{C}} \in \mathcal{R}(\Pi, \Pi)$ of schedulers for \mathcal{A} such that for each pair of schedulers (π, π') for \mathcal{A} in $R_{\mathcal{C}}$, the difference $|\Pr_s^\pi(\ell(X) \in E) - \Pr_t^{\pi'}(\ell(X) \in E)|$ for any measurable $E \subseteq L^\omega$, is bounded by the discrepancy $\gamma_1^{\mathcal{C}}(s, t)$.

The definition of $R_{\mathcal{C}}$ can be intuitively understood by recalling that $\gamma_1^{\mathcal{C}}(s, t)$ corresponds to the maximal probability of reaching a pair of states with different labels from the state pair (s, t) in the automaton $\mathcal{A}_{\mathcal{C}}$ induced from \mathcal{C} . Roughly, $R_{\mathcal{C}}$ collects all the pairs of schedulers for \mathcal{A} obtained as the left and right projection of a scheduler for $\mathcal{A}_{\mathcal{C}}$. How the projections are defined is technical and the interested reader can find the formal definition in the proof.

Lemma 7.3. *For any coupling structure \mathcal{C} for \mathcal{A} and $s, t \in S$, exists $R_{\mathcal{C}} \in \mathcal{R}(\Pi, \Pi)$ such that, for all measurable $E \subseteq L^\omega$ and $(\pi, \pi') \in R_{\mathcal{C}}$, $|\Pr_s^\pi(\ell(X) \in E) - \Pr_t^{\pi'}(\ell(X) \in E)| \leq \gamma_1^{\mathcal{C}}(s, t)$.*

Proof. Fix $s, t \in S$ and $\mathcal{C} = (f, \rho)$ a coupling structure for \mathcal{A} . Let $\mathcal{A}_{\mathcal{C}}$ be the automaton associated with the coupling structure \mathcal{C} .

We split the proof in two parts. (Part 1) deals with the definition of the set-coupling $R_{\mathcal{C}} \in \mathcal{R}(\Pi, \Pi)$; (Part 2) with proving that $|\Pr_s^\pi(\ell(X) \in E) - \Pr_t^{\pi'}(\ell(X) \in E)| \leq \gamma_1^{\mathcal{C}}(s, t)$, for all pairs of randomized schedulers $(\pi, \pi') \in R_{\mathcal{C}}$.

Hereafter, for a nonempty finite sequence $\sigma \in S^+$ and a random variable X on S^ω , we use $X \prec \sigma$ to denote $X \in \sigma S^\omega$.

Part 1: Let $(X, Y) \in S^\omega \times S^\omega$ be the random variable describing the infinite sequence of state pairs along a run of \mathcal{A}_C . Then, for any two nonempty finite sequences of the same length $\sigma = \sigma_0 \dots \sigma_n$ and $\tau = \tau_0 \dots \tau_n$ over S ,

$$\Pr_{(s,t)}^\pi((X, Y) \prec \langle \sigma, \tau \rangle) = \Pr_{(s,t)}^\pi(X \prec \sigma, Y \prec \tau)$$

is the probability that, starting from (s, t) , a run of \mathcal{A}_C under the scheduler π has prefix $\langle \sigma, \tau \rangle = (\sigma_0, \tau_0) \dots (\sigma_n, \tau_n)$. The above can be alternatively formulated in terms of conditional probabilities in the following two ways:

$$\Pr_{(s,t)}^\pi(X \prec \sigma, Y \prec \tau) = \Pr_{(s,t)}^\pi(X \prec \sigma) \cdot \Pr_{(s,t)}^\pi(Y \prec \tau \mid X \prec \sigma), \quad (7.1)$$

$$\Pr_{(s,t)}^\pi(X \prec \sigma, Y \prec \tau) = \Pr_{(s,t)}^\pi(Y \prec \tau) \cdot \Pr_{(s,t)}^\pi(X \prec \sigma \mid Y \prec \tau). \quad (7.2)$$

Given a scheduler π for \mathcal{A}_C , we define the maps $\pi_S, \pi_T: S^+ \rightarrow \mathcal{D}(S)$ as follows⁴, for arbitrary nonempty sequences σ, τ over S

$$\pi_S(\sigma)(u) = \sum_{\tau \in S^{|\sigma|}} \left(\Pr_{(s,t)}^\pi(Y \prec \tau \mid X \prec \sigma) \sum_{v \in S} \pi(\langle \sigma, \tau \rangle)(u, v) \right), \quad (7.3)$$

$$\pi_T(\tau)(u) = \sum_{\sigma \in S^{|\tau|}} \left(\Pr_{(s,t)}^\pi(X \prec \sigma \mid Y \prec \tau) \sum_{v \in S} \pi(\langle \sigma, \tau \rangle)(u, v) \right). \quad (7.4)$$

We call π_S and π_T , the left and right projections of π .

Intuitively, $\pi_S(\sigma)(u)$ describes the probability that under the scheduler π a run of \mathcal{A}_C with initial state (s, t) has a prefix of the form $\langle \sigma u, \tau' \rangle$, for some $\tau' \in S^{|\sigma|+1}$; symmetrically, $\pi_T(\tau)(u)$ is the probability that the prefix is of the form $\langle \sigma', \tau u \rangle$, for some $\sigma' \in S^{|\tau|+1}$.

Next, we prove that π_S and π_T are well-defined schedulers for \mathcal{A} . We provide the proof only for π_S , as it is similar for π_T . We need to show that $\pi_S(\sigma)$ is a convex combination of the form $\sum_{\mu \in \delta(\sigma_n)} \alpha_\mu \cdot \mu$, for some $\alpha_\mu \in [0, 1]$ such that $\sum_{\mu \in \delta(\sigma_n)} \alpha_\mu = 1$. By hypothesis that π is a scheduler for \mathcal{A}_C , we have that $\pi(\langle \sigma, \tau \rangle) = \sum_{(\mu, \nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu, \nu} \cdot f(\mu, \nu)$, for some $\xi_{\mu, \nu} \in [0, 1]$, such that $\sum_{(\mu, \nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu, \nu} = 1$. Without loss of generality, we can assume that $\xi_{\mu, \nu} = 0$, whenever $(\mu, \nu) \notin \rho(\sigma_n, \tau_n)$.

⁴We assume that $\Pr_{(s,t)}^\pi(E \mid F) = 0$, whenever $\Pr_{(s,t)}^\pi(F) = 0$ for any two measurable events E, F .

Let $\kappa_{\sigma,\tau} = \Pr_{(s,t)}^\pi(Y \prec \tau \mid X \prec \sigma)$, then

$$\begin{aligned}
\pi_S(\sigma)(u) &= \sum_{\tau \in S^{|\sigma|}} \left(\kappa_{\sigma,\tau} \sum_{v \in S} \pi(\langle \sigma, \tau \rangle)(u, v) \right) && (\text{eq. (7.3)}) \\
&= \sum_{\tau \in S^{|\sigma|}} \left(\kappa_{\sigma,\tau} \sum_{v \in S} \sum_{(\mu,\nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu,\nu} \cdot f(\mu, \nu)(u, v) \right) && (\pi \text{ scheduler}) \\
&= \sum_{\tau \in S^{|\sigma|}} \left(\kappa_{\sigma,\tau} \sum_{(\mu,\nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu,\nu} \sum_{v \in S} f(\mu, \nu)(u, v) \right) \\
&= \sum_{\tau \in S^{|\sigma|}} \left(\kappa_{\sigma,\tau} \sum_{(\mu,\nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu,\nu} \cdot \mu(u) \right) && (f \text{ measure-coupling map}) \\
&= \sum_{\tau \in S^{|\sigma|}} \left(\kappa_{\sigma,\tau} \sum_{\mu \in \delta(\sigma_n)} \sum_{\nu \in \delta(\tau_n)} \xi_{\mu,\nu} \cdot \mu(u) \right) && (\rho \text{ set-coupling map}) \\
&= \sum_{\mu \in \delta(\sigma_n)} \left(\sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma,\tau} \sum_{\nu \in \delta(\tau_n)} \xi_{\mu,\nu} \right) \cdot \mu(u)
\end{aligned}$$

By letting $\alpha_\mu = \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma,\tau} \sum_{\nu \in \delta(\tau_n)} \xi_{\mu,\nu}$, we get $\pi_S(\sigma) = \sum_{\mu \in \delta(\sigma_n)} \alpha_\mu \cdot \mu$ in the desired form. Next we show that this is a convex combination, i.e., $\sum_{\mu \in \delta(\sigma_n)} \alpha_\mu = 1$.

$$\begin{aligned}
\sum_{\mu \in \delta(\sigma_n)} \alpha_\mu &= \sum_{\mu \in \delta(\sigma_n)} \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma,\tau} \sum_{\nu \in \delta(\tau_n)} \xi_{\mu,\nu} && (\text{def. } \alpha_\mu) \\
&= \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma,\tau} \sum_{\mu \in \delta(\sigma_n)} \sum_{\nu \in \delta(\tau_n)} \xi_{\mu,\nu} \\
&= \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma,\tau} \sum_{(\mu,\nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu,\nu} && (\rho \text{ set-coupling map}) \\
&= \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma,\tau} && (\sum_{(\mu,\nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu,\nu} = 1) \\
&= \sum_{\tau \in S^{|\sigma|}} \Pr_{(s,t)}^\pi(Y \prec \tau \mid X \prec \sigma) && (\text{def. } \kappa_{\sigma,\tau}) \\
&= 1 && (\text{probability})
\end{aligned}$$

So, π_S and π_T are well-defined schedulers for \mathcal{A} . Given the above, we define the relation $R_{\mathcal{C}} \subseteq \Pi \times \Pi$ on schedulers for \mathcal{A} by

$$R_{\mathcal{C}} = \{(\pi_S, \pi_T) \mid \pi \text{ scheduler on } \mathcal{A}_{\mathcal{C}}\}.$$

To better understand the definition of $R_{\mathcal{C}}$, recall that $\mathcal{A}_{\mathcal{C}}$ can be interpreted as the automaton describing the concurrent execution of two copies of \mathcal{A} synchronised according to \mathcal{C} . Then, π_S and π_T can be interpreted as the schedulers on obtained from π , by respectively taking the left and right projections of the executions of $\mathcal{A}_{\mathcal{C}}$ as computations on \mathcal{A} . The relation $R_{\mathcal{C}}$ is given as the collection of these pair of projections.

Next we prove that $R_{\mathcal{C}}$ is a set-coupling for (Π, Π) , that is

$$\{\pi_1 \mid \exists \pi_2 \in \Pi. (\pi_1, \pi_2) \in R_{\mathcal{C}}\} = \Pi \quad \text{and} \quad \{\pi_2 \mid \exists \pi_1 \in \Pi. (\pi_1, \pi_2) \in R_{\mathcal{C}}\} = \Pi.$$

By definition of $R_{\mathcal{C}}$, this is equivalent to prove that for an arbitrary pair of schedulers π_S, π_T for \mathcal{A} we can find a scheduler π for $\mathcal{A}_{\mathcal{C}}$ such that (7.3), (7.4) hold (hence, $(\pi_S, \pi_T) \in R_{\mathcal{C}}$).

Let $\sigma = \sigma_0 \dots \sigma_n$ and $\tau = \tau_0 \dots \tau_n$ be a pair of nonempty finite sequences of the same length over S , and assume $\pi_S(\sigma) = \sum_{\mu \in \delta(\sigma_n)} \alpha_{\mu}^{\sigma} \cdot \mu$ and $\pi_T(\tau) = \sum_{\nu \in \delta(\tau_n)} \beta_{\nu}^{\tau} \cdot \nu$, for some $\alpha_{\mu}^{\sigma}, \beta_{\nu}^{\tau} \in [0, 1]$ such that $\sum_{\mu \in \delta(\sigma_n)} \alpha_{\mu}^{\sigma} = 1$ and $\sum_{\nu \in \delta(\tau_n)} \beta_{\nu}^{\tau} = 1$. We define

$$\pi(\langle \sigma, \tau \rangle) = \sum_{(\mu, \nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu, \nu}^{\sigma, \tau} \cdot f(\mu, \nu) \quad \text{where } \xi_{\mu, \nu}^{\sigma, \tau} = \begin{cases} \alpha_{\mu}^{\sigma} \cdot \beta_{\nu}^{\tau} & \text{if } (\mu, \nu) \in \rho(\sigma_n, \tau_n) \\ 0 & \text{otherwise.} \end{cases}$$

By the fact that $\rho(\sigma_n, \tau_n)$ is a set-coupling in $\mathcal{R}(\delta(\sigma_n), \delta(\tau_n))$ and the definition of $\xi_{\mu, \nu}^{\sigma, \tau}$, it is easy to see that for all $\mu \in \delta(\sigma_n)$ and $\nu \in \delta(\tau_n)$

$$\sum_{\nu \in \delta(\tau_n)} \xi_{\mu, \nu}^{\sigma, \tau} = \alpha_{\mu}^{\sigma} \quad \text{and} \quad \sum_{\mu \in \delta(\sigma_n)} \xi_{\mu, \nu}^{\sigma, \tau} = \beta_{\nu}^{\tau}. \quad (7.5)$$

Next we show that (7.3) holds. Let $\kappa_{\sigma, \tau} = \Pr_{(s, t)}^{\pi}(Y \prec \tau \mid X \prec \sigma)$, then

$$\begin{aligned} & \pi_S(\sigma)(u) \\ &= \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma, \tau} \pi_S(\sigma)(u) && \text{(convex combination)} \\ &= \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma, \tau} \left(\sum_{\mu \in \delta(\sigma_n)} \alpha_{\mu}^{\sigma} \cdot \mu(u) \right) && \text{(eq. (7.3))} \\ &= \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma, \tau} \left(\sum_{\mu \in \delta(\sigma_n)} \sum_{\nu \in \delta(\tau_n)} \xi_{\mu, \nu}^{\sigma, \tau} \cdot \mu(u) \right) && \text{(by (7.5))} \\ &= \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma, \tau} \left(\sum_{\mu \in \delta(\sigma_n)} \sum_{\nu \in \delta(\tau_n)} \xi_{\mu, \nu}^{\sigma, \tau} \sum_{v \in S} f(\mu, \nu)(u, v) \right) && (f(\mu, \nu) \in \Omega(\mu, \nu)) \\ &= \sum_{\tau \in S^{|\sigma|}} \kappa_{\sigma, \tau} \left(\sum_{(\mu, \nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu, \nu}^{\sigma, \tau} \sum_{v \in S} f(\mu, \nu)(u, v) \right) && (\rho \text{ set-coupl. map}) \\ &= \sum_{\tau \in S^{|\sigma|}} \left(\kappa_{\sigma, \tau} \sum_{v \in S} \sum_{(\mu, \nu) \in \rho(\sigma_n, \tau_n)} \xi_{\mu, \nu}^{\sigma, \tau} \cdot f(\mu, \nu)(u, v) \right) \\ &= \sum_{\tau \in S^{|\sigma|}} \left(\Pr_{(s, t)}^{\pi}(Y \prec \tau \mid X \prec \sigma) \sum_{v \in S} \pi(\langle \sigma, \tau \rangle)(u, v) \right) && \text{(def. } \kappa_{\sigma, \tau} \text{ and } \pi) \end{aligned}$$

Equation (7.4) is proven symmetrically.

Part 2: We prove $\Pr_{(s, t)}^{\pi} \in \Omega(\Pr_s^{\pi_S}, \Pr_t^{\pi_T})$ first. Showing the marginal conditions corresponds to prove that, for all nonempty sequences $\sigma = \sigma_0 \dots \sigma_n$ and $\tau = \tau_0 \dots \tau_n$ over S ,

$$\Pr_s^{\pi_S}(X \prec \sigma) = \Pr_{(s, t)}^{\pi}(X \prec \sigma) \quad \text{and} \quad \Pr_t^{\pi_T}(Y \prec \tau) = \Pr_{(s, t)}^{\pi}(Y \prec \tau).$$

We prove only the equality on the left, as the other is similar. We proceed by induction on $n \geq 0$.

- Base case, $n = 0$. Then $\sigma = \sigma_0 \in S$ and

$$\begin{aligned}
\Pr_s^{\pi_S}(X \prec \sigma_0) &= \mathbf{1}_s(\sigma_0) && \text{(def. } \Pr_s^{\pi_S}) \\
&= \sum_{\tau_0 \in S} \mathbf{1}_s(\sigma_0) \cdot \mathbf{1}_t(\tau_0) && \text{(convex combination)} \\
&= \sum_{\tau_0 \in S} \mathbf{1}_{(s,t)}(\sigma_0, \tau_0) && \text{(def. } \mathbf{1}_{(s,t)}) \\
&= \sum_{\tau_0 \in S} \Pr_{(s,t)}^\pi(X \prec \sigma_0, Y \prec \tau_0) && \text{(def. } \Pr_{(s,t)}^\pi) \\
&= \Pr_{(s,t)}^\pi(X \prec \sigma_0). && \text{(additivity)}
\end{aligned}$$

- Inductive step, $n \geq 0$. Let $\sigma = \sigma_0 \dots \sigma_n$ and $s' \in S$, then

$$\begin{aligned}
&\Pr_s^{\pi_S}(X \prec \sigma s') \\
&= \Pr_s^{\pi_S}(X \prec \sigma) \cdot \pi_S(\sigma)(s') && \text{(def. } \Pr_s^{\pi_S}) \\
&= \Pr_{(s,t)}^\pi(X \prec \sigma) \cdot \pi_S(\sigma)(s') && \text{(inductive hp)} \\
&= \Pr_{(s,t)}^\pi(X \prec \sigma) \cdot \sum_{\tau \in S^{|\sigma|}} \left(\Pr_{(s,t)}^\pi(Y \prec \tau \mid X \prec \sigma) \cdot \sum_{t' \in S} \pi(\langle \sigma, \tau \rangle)(s', t') \right) && \text{(eq. (7.3))} \\
&= \sum_{\tau \in S^{|\sigma|}} \sum_{t' \in S} \Pr_{(s,t)}^\pi(X \prec \sigma) \cdot \Pr_{(s,t)}^\pi(Y \prec \tau \mid X \prec \sigma) \cdot \pi(\langle \sigma, \tau \rangle)(s', t') \\
&= \sum_{\tau \in S^{|\sigma|}} \sum_{t' \in S} \Pr_{(s,t)}^\pi(X \prec \sigma, Y \prec \tau) \cdot \pi(\langle \sigma, \tau \rangle)(s', t') && \text{(by (7.1))} \\
&= \sum_{\tau \in S^{|\sigma|}} \sum_{t' \in S} \Pr_{(s,t)}^\pi(X \prec \sigma s', Y \prec \tau t') && \text{(def. } \Pr_{(s,t)}^\pi) \\
&= \Pr_{(s,t)}^\pi(X \prec \sigma s'). && \text{(additivity)}
\end{aligned}$$

The right-marginal condition is proven symmetrically.

Note that the discrepancy $\gamma_1^C(s, t)$ is the maximal probability of reaching a state pair (u, v) such that $\ell(u) \neq \ell(v)$ by starting from the state pair (s, t) in \mathcal{A}_C . That is,

$$\gamma_1^C(s, t) = \sup_{\pi} \Pr_{(s,t)}^\pi(\ell(X) \neq \ell(Y)), \quad (7.6)$$

where π ranges over all schedulers for \mathcal{A}_C . Thus, from $\Pr_{(s,t)}^\pi \in \Omega(\Pr_s^{\pi_S}, \Pr_t^{\pi_T})$ we have

$$\begin{aligned}
\Pr_s^{\pi_S}(E) &= \Pr_{(s,t)}^\pi(\ell(X) \in E) \\
&\geq \Pr_{(s,t)}^\pi(\ell(X) = \ell(Y), \ell(Y) \in E) \\
&= 1 - \Pr_{(s,t)}^\pi(\ell(X) \neq \ell(Y) \cup \ell(Y) \notin E) \\
&\geq 1 - \Pr_{(s,t)}^\pi(\ell(X) \neq \ell(Y)) - \Pr_{(s,t)}^\pi(\ell(Y) \notin E) \\
&= \Pr_{(s,t)}^\pi(\ell(Y) \in E) - \Pr_{(s,t)}^\pi(\ell(X) \neq \ell(Y)) \\
&\geq \Pr_{(s,t)}^\pi(\ell(Y) \in E) - \gamma_1^C(s, t) \\
&= \Pr_{(s,t)}^{\pi_T}(E) - \gamma_1^C(s, t).
\end{aligned}$$

From the above, we conclude that $|\Pr_s^{\pi_S}(\ell(X) \in E) - \Pr_t^{\pi_T}(\ell(Y) \in E)| \leq \gamma_1^C(s, t)$.

The proofs follows immediately by combining Part 1 and 2. \square

Given Lemma 7.3 it is easy to establish the result stated in Theorem 7.1.

Proof of Theorem 7.1. Let $d_{\mathbb{R}}(x, y) = |x - y|$ denote the Euclidean distance on the real line and define $K = \{\Pr_s^\pi(\ell(X) \in E) \mid \pi \in \Pi\}$ and $H = \{\Pr_t^\pi(\ell(X) \in E) \mid \pi \in \Pi\}$.

Then, by [Mém11, Lemma 3.2]

$$\begin{aligned} \mathcal{H}(d_{\mathbb{R}})(K, H) &\geq \max\{|\sup K - \sup H|, |\inf K - \inf H|\} \\ &= \max\{|\text{Max}_s(E) - \text{Max}_t(E)|, |\text{Min}_s(E) - \text{Min}_t(E)|\}. \end{aligned}$$

Next we show that $\mathbf{d}_1(s, t) \geq \mathcal{H}(d_{\mathbb{R}})(K, H)$.

$$\begin{aligned} &\mathcal{H}(d_{\mathbb{R}})(K, H) \\ &= \inf \left\{ \sup_{(\pi, \pi') \in R} |\Pr_s^\pi(\ell(X) \in E) - \Pr_t^{\pi'}(\ell(X) \in E)| \mid R \in \mathcal{R}(\Pi, \Pi) \right\} \quad (\text{Theorem 2.1}) \\ &\leq \inf \{ \gamma_1^{\mathcal{C}} \mid \mathcal{C} \text{ coupling structure for } \mathcal{A} \} \quad (\text{Lemma 7.3}) \\ &= \mathbf{d}_1(s, t). \quad (\text{Theorem 5.7}) \end{aligned}$$

Therefore, $|\text{Max}_s(E) - \text{Max}_t(E)| \leq \mathbf{d}_1(s, t)$ and $|\text{Min}_s(E) - \text{Min}_t(E)| \leq \mathbf{d}_1(s, t)$. \square

Another consequence of Lemma 7.3 is that the bisimilarity distance provides an upper bound of the Hausdorff lifting of the variational distance between sets of distributions induced by the Markov chains obtained by ranging over all possible schedulers. In the theorem we use TV to denote the *total variation distance* between probability measures, defined as $TV(\mu, \nu) = \sup_E |\mu(E) - \nu(E)|$, where E ranges over all measurable subsets.

Theorem 7.4. $\mathcal{H}(TV)(\{\Pr_s^\pi(\ell(X) \in \cdot) \mid \pi \in \Pi\}, \{\Pr_t^\pi(\ell(X) \in \cdot) \mid \pi \in \Pi\}) \leq \mathbf{d}_1(s, t)$.

Proof.

$$\begin{aligned} &\mathcal{H}(TV)(\{\Pr_s^\pi(\ell(X) \in \cdot) \mid \pi \in \Pi\}, \{\Pr_t^\pi(\ell(X) \in \cdot) \mid \pi \in \Pi\}) = \\ &= \inf \left\{ \sup_{(\pi, \pi') \in R} TV(\Pr_s^\pi(\ell(X) \in \cdot), \Pr_t^{\pi'}(\ell(X) \in \cdot)) \mid R \in \mathcal{R}(\Pi, \Pi) \right\} \quad (\text{Theorem 2.1}) \\ &= \inf \left\{ \sup_{(\pi, \pi') \in R} \sup_E |\Pr_s^\pi(\ell(X) \in E) - \Pr_t^{\pi'}(\ell(X) \in E)| \mid R \in \mathcal{R}(\Pi, \Pi) \right\} \quad (\text{def. } TV) \\ &\leq \inf \{ \gamma_1^{\mathcal{C}} \mid \mathcal{C} \text{ coupling structure for } \mathcal{A} \} \quad (\text{Lemma 7.3}) \\ &= \mathbf{d}_1(s, t). \quad (\text{Theorem 5.7}) \quad \square \end{aligned}$$

Theorem 7.4 can be alternatively stated as follows. For any scheduler π there exists a scheduler π' such that $|\Pr_s^\pi(\ell(X) \in E) - \Pr_t^{\pi'}(\ell(X) \in E)| \leq \mathbf{d}_1(s, t)$, for all measurable subsets $E \subseteq L^\omega$.

8. CONCLUSION AND FUTURE WORK

We presented a novel characterization of the probabilistic bisimilarity distance of Deng et al. [DCPP06] as the solution of a simple stochastic game. Starting from it, we designed algorithms for computing the distances based on Condon’s simple policy iteration algorithm. The correctness of Condon’s approach relies on the assumption that the input game is stopping. This may not be the case for our probabilistic bisimilarity games when the discount factor is one. We overcame this problem by means of an improved termination condition based on the notion of self-closed relation due to Fu [Fu12].

As in [TvB16], our simple policy iteration algorithm has exponential worst-case time complexity. Nevertheless, experiments show that our method can compete in practice with the value iteration algorithm by Fu [Fu12] which has theoretical polynomial-time complexity for $\lambda < 1$. To the best of our knowledge, our algorithm is the first practical solution for computing the bisimilarity distance when $\lambda = 1$, performing orders of magnitude faster than the existing solutions based on the existential fragment of the first-order theory of the reals [CdAMR08, CdAMR10, CHL07].

As future work, we plan to improve on the current implementation in the line of [TvB18a], by exploiting the fact that bisimilar states and probabilistic distance one [TvB18b] can be efficiently pre-computed before starting the policy iteration. We believe that this would yield a significant cut down in the time required to compute the discrepancy at each iteration which turned out to be the bottleneck of our algorithms.

More efficient algorithms might lead to the speedup of verification tools for concurrent probabilistic systems, as behavioral distances relate to the satisfiability of logical properties. For the case of labelled Markov chains, in [CvBW12, BBLM15] the variational difference between two states with respect to their probability of satisfying linear-time properties (eg., LTL formulas) is shown to be bound by the (undiscounted) probabilistic bisimilarity distance. In Section 7 we showed that a similar result holds for the case of probabilistic automata with additional subtleties that arise by the need of handling the nondeterminism. In light of this relation it would be interesting to develop approximated techniques to cut down the overall model-checking time of probabilistic automata as briefly discussed in Remark 7.2.

We also plan to extend the work on approximated minimization [BBLM17, BBLM18] to the case of probabilistic automata and explore the possible relation between the probabilistic bisimilarity distance with more expressive logics for concurrent probabilistic systems [CdAMR08, CdAMR10, Mio12].

Acknowledgments. The authors are grateful to the referees for their constructive feedback.

REFERENCES

- [BBL⁺19] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, Radu Mardare, Qiyi Tang, and Franck van Breugel. Computing Probabilistic Bisimilarity Distances for Probabilistic Automata. In *30th International Conference on Concurrency Theory, CONCUR 2019*, volume 140 of *LIPICs*, pages 9:1–9:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- [BBLM13] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, and Radu Mardare. On-the-Fly Exact Computation of Bisimilarity Distances. In *19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2013*, volume 7795 of *LNCS*, pages 1–15, 2013.
- [BBLM15] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, and Radu Mardare. Converging from Branching to Linear Metrics on Markov Chains. In *12th International Colloquium Theoretical Aspects of*

- Computing, ICTAC 2015*, volume 9399 of *Lecture Notes in Computer Science*, pages 349–367. Springer, 2015.
- [BBLM17] Giovanni Bacci, Giorgio Bacci, Kim G. Larsen, and Radu Mardare. On the Metric-Based Approximate Minimization of Markov Chains. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*, volume 80 of *LIPIcs*, pages 104:1–104:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
- [BBLM18] Giovanni Bacci, Giorgio Bacci, Kim G. Larsen, and Radu Mardare. On the metric-based approximate minimization of Markov Chains. *J. Log. Algebr. Meth. Program.*, 100:36–56, 2018.
- [BCD⁺11] Clark Barrett, Christopher L. Conway, Morgan Deters, Liana Hadarean, Dejan Jovanović, Tim King, Andrew Reynolds, and Cesare Tinelli. CVC4. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *23rd International Conference on Computer Aided Verification, CAV 2011*, volume 6806 of *Lecture Notes in Computer Science*, pages 171–177. Springer, 2011.
- [BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.
- [CdAMR08] Krishnendu Chatterjee, Luca de Alfaro, Rupak Majumdar, and Vishwanath Raman. Algorithms for Game Metrics. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2008*, volume 2 of *LIPIcs*, pages 107–118. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2008.
- [CdAMR10] Krishnendu Chatterjee, Luca de Alfaro, Rupak Majumdar, and Vishwanath Raman. Algorithms for Game Metrics (Full Version). *Logical Methods in Computer Science*, 6(3), 2010.
- [CHL07] Taolue Chen, Tingting Han, and Jian Lu. On Behavioral Metric for Probabilistic Systems: Definition and Approximation Algorithm. In *4th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007*, pages 21–25. IEEE Computer Society, 2007.
- [Con90] Anne Condon. On Algorithms for Simple Stochastic Games. In *Advances In Computational Complexity Theory*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 51–72. DIMACS/AMS, 1990.
- [Con92] Anne Condon. The Complexity of Stochastic Games. *Inf. Comput.*, 96(2):203–224, 1992.
- [CvBW12] Di Chen, Franck van Breugel, and James Worrell. On the Complexity of Computing Probabilistic Bisimilarity. In *15th International Conference on Foundations of Software Science and Computational Structures, FOSSACS 2012*, volume 7213 of *Lecture Notes in Computer Science*, pages 437–451. Springer, 2012.
- [DCPP06] Yuxin Deng, Tom Chothia, Catuscia Palamidessi, and Jun Pang. Metrics for Action-labelled Quantitative Transition Systems. *Electr. Notes Theor. Comput. Sci.*, 153(2):79–96, 2006.
- [DGJP04] Josee Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for Labelled Markov Processes. *Theor. Comput. Sci.*, 318(3):323–354, 2004.
- [DLT08] Josée Desharnais, François Laviolette, and Mathieu Tracol. Approximate Analysis of Probabilistic Processes: Logic, Simulation and Games. In *5th International Conference on the Quantitative Evaluation of Systems, QEST 2008*, pages 264–273. IEEE Computer Society, 2008.
- [FKP17] Nathanaël Fijalkow, Bartek Klin, and Prakash Panangaden. Expressiveness of Probabilistic Modal Logics, Revisited. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*, volume 80 of *LIPIcs*, pages 105:1–105:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.
- [Fu12] Hongfei Fu. Computing Game Metrics on Markov Decision Processes. In *39th International Colloquium on Automata, Languages, and Programming, ICALP 2012*, volume 7392 of *Lecture Notes in Computer Science*, pages 227–238. Springer, 2012.
- [Fu14] Hongfei Fu. *Verifying Probabilistic Systems: New Algorithms and Complexity Results*. PhD thesis, RWTH Aachen, Aachen, Germany, Nov 2014.
- [GJS90] Alessandro Giacalone, Chi-Chang Jou, and Scott A. Smolka. Algebraic Reasoning for Probabilistic Concurrent Systems. In *Proceedings of the IFIP WG 2.2/2.3 Working Conference on Programming Concepts and Methods*, pages 443–458. North-Holland, 1990.
- [Hau14] Felix Hausdorff. *Grundzüge der Mengenlehre*. Verlag Von Veit & Comp, Leipzig, 1914.
- [JL91] Bengt Jonsson and Kim G. Larsen. Specification and Refinement of Probabilistic Processes. In *6th Annual Symposium on Logic in Computer Science, LICS 1991*, pages 266–277. IEEE Computer Society, 1991.
- [Jub05] Brendan Juba. On the Hardness of Simple Stochastic Games. Master’s thesis, Carnegie Mellon University, Pittsburgh, PA, USA, May 2005.

- [Kan42] Leonid Vitalevich Kantorovich. On the transfer of masses (in Russian). *Doklady Akademii Nauk*, 5(5-6):1–4, 1942. Translated in *Management Science*, 1958.
- [KM18] Barbara König and Christina Mika-Michalski. (Metric) Bisimulation Games and Real-Valued Modal Logics for Coalgebras. In *29th International Conference on Concurrency Theory, CONCUR 2018*, volume 118 of *LIPIcs*, pages 37:1–37:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.
- [KS95] Peter Kleinschmidt and Heinz Schannath. A Strongly Polynomial Algorithm for the Transportation Problem. *Math. Program.*, 68:1–13, 1995.
- [KW67] Viktor Klee and Christoph Witzgall. Facets and Vertices of Transportation Polytopes. In *5th Summer Seminar on the Mathematics of the Decision Sciences*, volume 11 of *Lectures in Applied Mathematics*, pages 257–282. AMS, 1967.
- [LL69] Thomas Liggett and Steven A. Lippman. Stochastic Games with Perfect Information and Time Average Payoff. *SIAM Review*, 11(4):604–607, 1969.
- [Mém11] Facundo Mémoli. Gromov-Wasserstein Distances and the Metric Approach to Object Matching. *Foundations of Computational Mathematics*, 11(4):417–487, 2011.
- [Mio12] Matteo Mio. On the Equivalence of Game and Denotational Semantics for the Probabilistic μ -calculus. *Logical Methods in Computer Science*, 8(2), 2012.
- [Orl85] James B. Orlin. On the Simplex Algorithm for Networks and Generalized Networks. In *Mathematical Programming Essays in Honor of George B. Dantzig Part I*, pages 166–178. Springer Berlin Heidelberg, 1985.
- [PC19] Gabriel Peyré and Marco Cuturi. Computational Optimal Transport. *Foundations and Trends in Machine Learning*, 11(5-6):355–607, 2019.
- [Put94] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1994.
- [Sch99] Alexander Schrijver. *Theory of Linear and Integer Programming*. Wiley-Interscience series in Discrete Mathematics and Optimization. Wiley, 1999.
- [Seg95] Roberto Segala. *Modeling and Verification of Randomized Distributed Real-time Systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.
- [Sha53] Lloyd S. Shapley. Stochastic Games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.
- [SL94] Roberto Segala and Nancy A. Lynch. Probabilistic Simulations for Probabilistic Processes. In *5th International Conference on Concurrency Theory, CONCUR 1994*, volume 836 of *Lecture Notes in Computer Science*, pages 481–496. Springer, 1994.
- [Str89] James K. Strayer. *Linear Programming and its Applications*. Undergraduate Texts in Mathematics. Springer-Verlag, New York, NY, USA, 1989.
- [Tan18] Qiyi Tang. *Computing Probabilistic Bisimilarity Distances*. PhD thesis, York University, Toronto, Canada, August 2018.
- [TvB16] Qiyi Tang and Franck van Breugel. Computing Probabilistic Bisimilarity Distances via Policy Iteration. In *27th International Conference on Concurrency Theory, CONCUR 2016*, volume 59 of *LIPIcs*, pages 22:1–22:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.
- [TvB18a] Qiyi Tang and Franck van Breugel. Deciding Probabilistic Bisimilarity Distance One for Labelled Markov Chains. In *30th International Conference on Computer Aided Verification, CAV 2018*, volume 10981 of *Lecture Notes in Computer Science*, pages 681–699. Springer, 2018.
- [TvB18b] Qiyi Tang and Franck van Breugel. Deciding Probabilistic Bisimilarity Distance One for Probabilistic Automata. In Sven Schewe and Lijun Zhang, editors, *29th International Conference on Concurrency Theory, CONCUR 2018*, volume 118 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:17. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- [vB12] Franck van Breugel. On Behavioural Pseudometrics and Closure Ordinals. *Inf. Process. Lett.*, 112(19):715–718, 2012.
- [vBSW08] Franck van Breugel, Babita Sharma, and James Worrell. Approximating a Behavioural Pseudometric without Discount for Probabilistic Systems. *Logical Methods in Computer Science*, 4(2), 2008.
- [vBW14] Franck van Breugel and James Worrell. The Complexity of Computing a Bisimilarity Pseudometric on Probabilistic Automata. In *Horizons of the Mind. A Tribute to Prakash Panangaden*

—*Essays Dedicated to Prakash Panangaden on the Occasion of His 60th Birthday*, volume 8464 of *Lecture Notes in Computer Science*, pages 191–213. Springer, 2014.

[Vil08] Cédric Villani. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer, 2008.

[ZP96] Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1/2):343–359, May 1996.