Deciding Probabilistic Bisimilarity Distance One for Probabilistic Automata

Qiyi Tang
Department of Computing, Imperial College, London, United Kingdom

Franck van Breugel
Department of Electrical Engineering and Computer Science, York University, Toronto, Canada

Abstract

Probabilistic bisimilarity, due to Segala and Lynch, is an equivalence relation that captures which states of a probabilistic automaton behave exactly the same. Deng, Chothia, Palamidessi and Pang proposed a robust quantitative generalization of probabilistic bisimilarity. Their probabilistic bisimilarity distances of states of a probabilistic automaton capture the similarity of their behaviour. The smaller the distance, the more alike the states behave. In particular, states are probabilistic bisimilar if and only if their distance is zero.

Although the complexity of computing probabilistic bisimilarity distances for probabilistic automata has already been studied and shown to be in **NP** $\cap$ **coNP** and **PPAD**, we are not aware of any practical algorithm to compute those distances. In this paper we provide several key results towards algorithms to compute probabilistic bisimilarity distances for probabilistic automata. In particular, we present a polynomial time algorithm that decides distance one. Furthermore, we give an alternative characterization of the probabilistic bisimilarity distances as a basis for a policy iteration algorithm.

2012 ACM Subject Classification Mathematics of computing → Markov processes, Theory of computation → Concurrency

Keywords and phrases probabilistic automaton, probabilistic bisimilarity, distance

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2018.9

Funding Natural Sciences and Engineering Research Council of Canada

Acknowledgements The authors would like to thank the referees for their detailed and constructive feedback.

1 Introduction

Behavioural equivalences, such as bisimilarity, are one of the cornerstones of concurrency theory. Recall that a behavioural equivalence $\sim \subseteq S \times S$, where S is the set of states of the model, satisfies

$s \sim s$
if $s \sim t$ then $t \sim s$
if $s \sim t$ and $t \sim u$ then $s \sim u$

for all s, t, $u \in S$. If $s \sim t$ then states s and t behave the same.

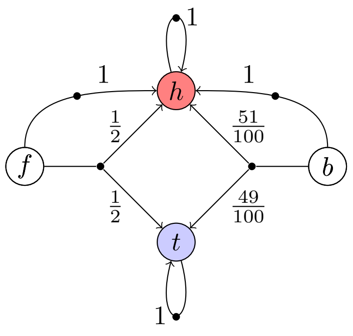
As first observed by Giacalone, Jou and Smolka [17], behavioural equivalences are *not robust* for models that contain quantitative information such as probabilities and time. This lack of robustness is caused by the discrepancy between the discrete nature of behavioural equivalence and the continuous nature of the quantitative information on the which the behavioural equivalence relies. In particular, even small changes to the quantitative information may cause behaviourally equivalent states become inequivalent or vice versa.

Giacalone et al. proposed *behavioural pseudometrics* as a robust quantitative generalization of behavioural equivalences. A behavioural pseudometric $d : S \times S \to [0, 1]$ satisfies

$$d(s,t) = 0 \text{ if and only if } s \sim t$$
$$d(s,t) = d(t,s)$$
$$d(s,u) \leq d(s,t) + d(t,u)$$

for all s, t, $u \in S$. The distance $d(s,t)$ measures the similarity of the behaviour of states s and t. The smaller this distance, the more alike the states behave. Distance zero captures that states are behaviourally equivalent.

In this paper, we focus on *probabilistic automata*. This model was first studied by Segala in [27]. It captures both nondeterminism (and, hence, concurrency) and probabilities. Let us consider a simple example.



The states of a probabilistic automaton are labelled. These labels provide a partition of the states so that states satisfying the same basic properties of interest are in the same partition. In the above example, the labels are represented by colours. Each state has one or more probabilistic transitions. For example, the state t has a single probabilistic transition that takes state t to itself with probability one. State f has two probabilistic transitions. The one takes state f to state h with probability one. The other represents a fair coin toss, that is, it transitions to state h with probability $\frac{1}{2}$ and to state t with probability $\frac{1}{2}$. Also state b has two transitions, one of which represents a biased coin toss.

Segala and Lynch [28] introduced *probabilistic bisimilarity*. This behavioural equivalence for probabilistic automata generalises the one introduced by Larsen and Skou [25]. The latter is applicable to models without nondeterminism, known as labelled Markov chains. States s and t of a probabilistic automaton are probabilistic bisimilar if for each outgoing probabilistic transition of state s there exists a matching outgoing probabilistic transition of state t, and vice versa. Two probabilistic transitions match if they both transition to each probabilistic bisimilarity equivalence class with the exact same probability. States f and b in the above example are not probabilistic bisimilar. Although the transition from state f to state h can be matched by the transition from state b to state h, the probabilistic transitions representing a fair and biased coin toss do not match since the probabilities are slightly different.

Deng, Chothia, Palamidessi and Pang [12] introduced a behavioural pseudometric for probabilistic automata that generalises probabilistic bisimilarity. The Hausdorff metric [18] and the Kantorovich metric [22] are key ingredients of this pseudometric. The former is used to capture nondeterminism. This idea dates back to the work of De Bakker and Zucker [4]. The latter was first used by Van Breugel and Worrell [7] to capture probabilistic behaviour. On the one hand, the behaviours of the states h and t of the above example 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 these states have probabilistic bisimilarity distance $\frac{1}{100}$.

Tracol, Desharnais and Zhioua [34] also introduced a behavioural pseudometric for probabilistic automata. Their probabilistic bisimilarity distances generalise probabilistic bisimilarity as well, but are different from the ones introduced by Deng et al. An example showing the difference can be found in [34, Example 5]. To compute their probabilistic bisimilarity distances, they developed an iterative algorithm. In each iteration, a maximum flow problem needs to be solved. The resulting algorithm is polynomial time.

The complexity of computing the probabilistic bisimilarity distances for probabilistic automata a la Deng et al. was first studied by Fu [15]. He showed that these probabilistic bisimilarity distances are rational. Furthermore, he proved that the problem of deciding whether the distance of two states is smaller than a given rational is in **NP** $\cap$ **coNP**. The proof can be adapted to show that the decision problem is in **UP** $\cap$ **coUP** [16]. Recall that **UP** contains those problems in **NP** with a unique accepting computation. Van Breugel and Worrell [8] have shown that the problem of computing the probabilistic bisimilarity distances is in **PPAD**, which is short for polynomial parity argument in a directed graph.

For the behavioural pseudometric of Deng et al., states are probabilistic bisimilar if and only if they have distance zero. Since probabilistic bisimilarity can be decided in polynomial time, as shown by Baier [2], distance zero can be decided in polynomial time as well. In Section 5 we present a polynomial time algorithm that decides distance one.

As we have already shown in [32] in the context of labelled Markov chains, being able to decide distance zero and distance one in polynomial time has significant impact on computing probabilistic bisimilarity distances. For example, we can determine in polynomial time how many, if any, distances are non-trivial, that is, greater than zero and smaller than one. The technical details in this paper are considerably more involved than those in [32].

Deng et al. define their pseudometric as a least fixed point. In Section 4 we present an alternative characterization of the probabilistic bisimilarity distances. This characterization is similar to the one presented for labelled Markov chains by Chen, Van Breugel and Worrell [9]. The latter characterization provided the foundation for the policy iteration algorithm to compute the probabilistic bisimilarity distances for labelled Markov chains by Bacci, Bacci, Larsen and Mardare [1] (see also [31]). Our alternative characterization plays a key role in the correctness proof of our algorithm.

2 Order and Distances

In this section, we provide some definitions and results from the literature about orders and distances that we will use in the remainder of this paper. For more details we refer the reader to, for example, [11] and [3]. Given a set S, we denote the set of functions from $S \times S$ to $[0, 1]$ by $[0, 1]^{S \times S}$. As in the work of Desharnais et al. [13], we endow the set $[0, 1]^{S \times S}$ with the following natural order.

▶ **Definition 1.** The relation $\sqsubseteq \subseteq [0, 1]^{S \times S} \times [0, 1]^{S \times S}$ is defined by

$$d \sqsubseteq e \text{ if } d(s, t) \leq e(s, t) \text{ for all } s, t \in S.$$

▶ **Proposition 2.** $\langle [0, 1]^{S \times S}, \sqsubseteq \rangle$ is a complete lattice.

Proof. See, for example, [13, Lemma 3.2]. ◀

Let $\langle X, \leq \rangle$ be an ordered set. Let $f : X \to X$. Following [11, Definition 8.14], we define the following three notions:

- $x \in X$ is a *fixed point* of f if $f(x) = x$,
- $x \in X$ is a *pre-fixed point* of f if $f(x) \leq x$, and
- $x \in X$ is a *post-fixed point* of f if $x \leq f(x)$.

A function $f : X \to X$ is *monotone* if for all x, $y \in X$, $x \leq y$ implies $f(x) \leq f(y)$. The following result is known as the Knaster-Tarski fixed point theorem [24, 33].

▶ **Theorem 3.** *Let X be a complete lattice and let $f : X \to X$ be a monotone function.*
(a) *f has a greatest fixed point.*
(b) *The greatest fixed point of f is the greatest post-fixed point of f.*
(c) *f has a least fixed point.*
(d) *The least fixed point of f is the least pre-fixed point of f.*

Proof. See, for example, [11, Theorem 2.35] and [11, Theorem 8.20]. ◀

We denote the greatest and least fixed point of a function f by $\boldsymbol{\nu} f$ and $\boldsymbol{\mu} f$, respectively. Given a set X, we denote the set of subsets of X by 2^X. The correctness of our iterative algorithm to decide distance one relies on the following theorem.

▶ **Theorem 4.** *Let X be a finite set and let $\Phi : 2^X \to 2^X$ be a monotone function.*
(a) *$\boldsymbol{\mu}\Phi = \Phi^n(\emptyset)$ for some $n \in \mathbb{N}$.*
(b) *$\boldsymbol{\nu}\Phi = \Phi^n(X)$ for some $n \in \mathbb{N}$.*
(c) *If $Y \subseteq \boldsymbol{\mu}\Phi$ then $\boldsymbol{\mu}\Phi = \Phi^n(Y)$ for some $n \in \mathbb{N}$.*

Proof. See, for example, [10, Lemma 8]. ◀

The set $[0,1]^{S \times S}$ also carries the following natural metric.

▶ **Definition 5.** The function $\| \cdot - \cdot \| : [0,1]^{S \times S} \times [0,1]^{S \times S} \to [0,1]$ is defined by

$$\|d - e\| = \sup_{s,t \in S} |d(s,t) - e(s,t)|.$$

▶ **Proposition 6.** $\langle [0,1]^{S \times S}, \| \cdot - \cdot \| \rangle$ is a nonempty complete metric space.

Proof. See, for example, [3, Section 1.1.2]. ◀

Let $\langle X, d \rangle$ be a metric space and $c \in (0,1]$. A function $f : X \to X$ is *c-Lipschitz* if for all x, $y \in X$, $d(f(x), f(y)) \leq c\, d(x,y)$. A 1-Lipschitz function is also called *nonexpansive*. A function is *contractive* if it is c-Lipschitz for some $c \in (0,1)$. The following result is known as Banach's fixed point theorem [5].

▶ **Theorem 7.** *Let X be a nonempty complete metric space and $f : X \to X$ a contractive function. Then f has a unique fixed point.*

Proof. See, for example, [3, Theorem 1.34]. ◀

The Hausdorff metric [18] is defined as follows.

▶ **Definition 8.** The function $H : [0,1]^{X \times X} \to [0,1]^{2^X \times 2^X}$ is defined by

$$H(d)(M,N) = \max \left\{ \max_{\mu \in M} \min_{\nu \in N} d(\mu,\nu), \max_{\nu \in N} \min_{\mu \in M} d(\mu,\nu) \right\}.$$

Given a nonempty finite set X, we denote the set of probability distributions on X by $\mathit{Distr}(X)$. For $\mu \in \mathit{Distr}(X)$, we define its support by $\mathrm{support}(\mu) = \{ x \in X \mid \mu(x) > 0 \}$.

▶ **Definition 9.** Let μ, $\nu \in \mathit{Distr}(X)$. The set $\Omega(\mu, \nu)$ of *couplings* of μ and ν is defined by

$$\Omega(\mu,\nu) = \left\{ \omega \in \mathit{Distr}(X \times X) \;\middle|\; \sum_{x \in X} \omega(x,y) = \mu(y) \text{ and } \sum_{y \in X} \omega(x,y) = \nu(x) \right\}.$$

In general, the set $\Omega(\mu, \nu)$ is infinite. The set of vertices of the convex polytope $\Omega(\mu, \nu)$ is denoted by $V(\Omega(\mu, \nu))$. The latter set is finite (see, for example, [23, page 259]). This fact will be crucial in the proof of Lemma 20. The Kantorovich metric [22] is defined as follows.

▶ **Definition 10.** The function $K : [0,1]^{X \times X} \to [0,1]^{\mathit{Distr}(X) \times \mathit{Distr}(X)}$ is defined by

$$K(d)(\mu,\nu) = \min_{\omega \in V(\Omega(\mu,\nu))} \sum_{u,v \in S} \omega(u,v)\, d(u,v).$$

The Hausdorff metric and the Kantorovich metric are key ingredients of the definition of the probabilistic bisimilarity distances, as we will see in the next section.

3 Probabilistic Automata

Also in this section, we recall some definitions and results from the literature. In particular, we introduce the model of interest, probabilistic automata, its best known behavioural equivalence, probabilistic bisimilarity, and its quantitative generalization. Probabilistic automata were first studied in the context of concurrency by Segala [27].

▶ **Definition 11.** A *probabilistic automaton* is a tuple $\langle S, L, \to, \ell \rangle$ consisting of

- a nonempty finite set S of *states*,
- a nonempty finite set L of *labels*,
- a finitely branching *transition relation* $\to \subseteq S \times \mathit{Distr}(S)$, and
- a *labelling function* $\ell : S \to L$.

Instead of $(s, \mu) \in \to$, we write $s \to \mu$. A transition relation is finitely branching if for all $s \in S$, the set $\{ \mu \in \mathit{Distr}(S) \mid s \to \mu \}$ is *nonempty* and finite. For the remainder of this paper we fix a probabilistic automaton $\langle S, L, \to, \ell \rangle$.

In order to define probabilistic bisimilarity, we first show how a relation on states can be lifted to a relation on distributions over states. This notion of lifting is due to Jonsson and Larsen [21].

▶ **Definition 12.** The *lifting* of a relation $\mathcal{R} \subseteq S \times S$ is the relation $\mathcal{R}\!\uparrow \subseteq \mathit{Distr}(S) \times \mathit{Distr}(S)$ defined by $(\mu, \nu) \in \mathcal{R}\!\uparrow$ if there exists $\omega \in V(\Omega(\mu, \nu))$ such that $\mathrm{support}(\omega) \subseteq \mathcal{R}$.

Probabilistic bisimilarity, a notion due to Segala and Lynch [28], is introduced next. States are probabilistic bisimilar if they have the same label and each probabilistic transition of the one state can be matched by a probabilistic transition of the other state, and vice versa. Two probabilistic transitions match if they transition with *exactly* the same probability to states that behave *exactly* the same.

▶ **Definition 13.** An equivalence relation $\mathcal{R} \subseteq S \times S$ is a *probabilistic bisimulation* if for all s, $t \in S$, if $(s, t) \in \mathcal{R}$ then

- $\ell(s) = \ell(t)$,

- for all $s \to \mu$ there exists $t \to \nu$ such that $(\mu, \nu) \in \mathcal{R}\!\uparrow$ and
- for all $t \to \nu$ there exists $s \to \mu$ such that $(\nu, \mu) \in \mathcal{R}\!\uparrow$.

Probabilistic bisimilarity, denoted $\sim$, is the largest probabilistic bisimulation.

For a proof that a largest probabilistic bisimulation exists, we refer the reader to, for example, [6, Proposition 4.3]. Relying on exact matching is the cause for a lack of robustness. To address this shortcoming, we define a quantitative generalization of probabilistic bisimilarity, the probabilistic bisimilarity distances, as the least fixed point of the function Δ_1. To prove an alternative characterization of the probabilistic bisimilarity distances in the next section, we also introduce a family of discounted versions of Δ_1, namely Δ_c with $c \in (0, 1)$.

▶ **Definition 14.** Let $c \in (0, 1]$. The function $\Delta_c : [0, 1]^{S \times S} \to [0, 1]^{S \times S}$ is defined by

$$\Delta_c(d)(s, t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ c\, H(K(d))(\{\, \mu \mid s \to \mu \,\}, \{\, \nu \mid t \to \nu \,\}) & \text{otherwise.} \end{cases}$$

▶ **Proposition 15.** For all $c \in (0, 1]$, the function Δ_c is monotone.

Proof. See [12, Lemma 2.10]. ◀

Since $\langle [0, 1]^{S \times S}, \sqsubseteq \rangle$ is a complete lattice according to Proposition 2 and Δ_c is a monotone function by Proposition 15, we can conclude from Theorem 3(c) that Δ_c has a least fixed point $\boldsymbol{\mu}\Delta_c$. The fact that the probabilistic bisimilarity distances $\boldsymbol{\mu}\Delta_1$ provide a quantitative generalization of probabilistic bisimilarity is captured by the following theorem due to Deng et al. [12].

▶ **Theorem 16.** *For all s, $t \in S$, $\boldsymbol{\mu}\Delta_1(s, t) = 0$ if and only if $s \sim t$.*

Proof. See [12, Corollary 2.14]. ◀

4 An Alternative Characterization

In the previous section, we defined the probabilistic bisimilarity distances as a least fixed point. Next, we present an alternative characterization. This generalizes the characterization of probabilistic bisimilarity distances for labelled Markov chains due to Chen et al. [9, Theorem 8]. First, we partition the set of state pairs as follows.

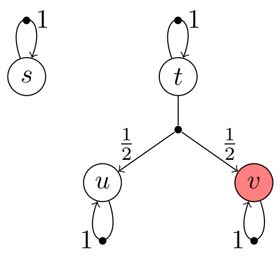
$$\begin{array}{lcl} S^2_0 & = & \{\, (s, t) \in S \times S \mid s \sim t \,\} \\ S^2_1 & = & \{\, (s, t) \in S \times S \mid \ell(s) \neq \ell(t) \,\} \\ S^2_? & = & (S \times S) \setminus (S^2_0 \cup S^2_1) \end{array}$$

Note that, due to Theorem 16 the state pairs in S^2_0 have distance zero. From Definition 14 we can infer that the state pairs in S^2_1 have distance one. The state pairs in $S^2_?$ cannot have distance zero, again due to Theorem 16, but can have any distance in the interval $(0, 1]$, including distance one.

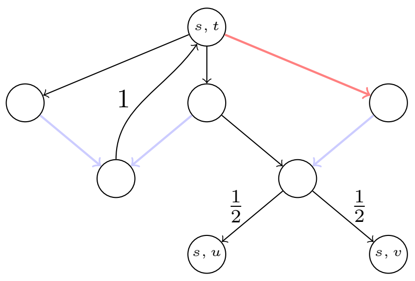
The characterization can be viewed as a two player game, a max player and a min player, similar to the one presented in [8]. The game can be considered a quantitative generalization of the game that characterizes bisimilarity (see [30]). In this turn based game, starting in a pair of states (s, t), the max player chooses a probabilistic transition from either s or t. Subsequently, the min player chooses a probabilistic transition from the other state and also chooses a coupling. For example, if the max player picks $s \to \mu$ and the min player picks $t \to \nu$, then the min player also has to choose $\omega \in V(\Omega(\mu, \nu))$. This will be formalized in

Definition 17. Recall that such a coupling ω is a probability distribution on $S \times S$. From a coupling ω the game moves to state pair (u, v) with probability $\omega(u, v)$.

Consider, for example, the following probabilistic automaton.



Note that the states s and u are probabilistic bisimilar. The corresponding game graph can be depicted as follows.



Since the game will be used to characterize the probabilistic bisimilarity distances, the state pairs for which we can easily determine their distance have no outgoing edges in the game graph. In particular, state pairs with different labels, which have distance one, and state pairs that are probabilistic bisimilar, which have distance zero, have no outgoing edges.

The objective of the max player is to maximize the expectation of reaching a state pair with different labels. The min player tries to minimize this expectation. In the above example, the max player tries to reach the state pair (s, v), whereas the min player tries to avoid that from happening. The policies, also known as strategies, for the max and min player are introduced next.

▶ **Definition 17.** The set $\mathcal{A}$ of *max policies* is defined by

$$\mathcal{A} = \left\{ A \in (S^2_? \to (S \times \mathit{Distr}(S))) \;\middle|\; \begin{array}{l} \forall (s,t) \in S^2_? : \\ (\exists \nu \in \mathit{Distr}(S) : A(s,t) = (s,\nu) \wedge t \to \nu) \vee \\ (\exists \mu \in \mathit{Distr}(S) : A(s,t) = (t,\mu) \wedge s \to \mu) \end{array} \right\}.$$

The set $\mathcal{I}$ of *min policies* is defined by

$$\mathcal{I} = \left\{ I \in ((S \times \mathit{Distr}(S)) \to \mathit{Distr}(S \times S)) \;\middle|\; \begin{array}{l} \forall (s,\nu) \in S \times \mathit{Distr}(S) : \exists \mu \in \mathit{Distr}(S) : \\ I(s,\nu) \in V(\Omega(\mu,\nu)) \wedge s \to \mu \end{array} \right\}.$$

Given a policy A for the max player and a policy I for the min player, we define the value function as the least fixed point of the function $\Gamma^{A,I}_1$. This least fixed point captures the expectation of reaching a state pair with different labels if both players use the given policies. We also introduce a family of discounted versions of $\Gamma^{A,I}_1$, namely $\Gamma^{A,I}_c$ with $c \in (0, 1)$, that we will use later in this section.

▶ **Definition 18.** Let $A \in \mathcal{A}$, $I \in \mathcal{I}$ and $c \in (0,1]$. The function $\Gamma_c^{A,I} : [0,1]^{S \times S} \to [0,1]^{S \times S}$ is defined by

$$\Gamma_c^{A,I}(d)(s,t) = \begin{cases} 0 & \text{if } (s,t) \in S_0^2 \\ 1 & \text{if } (s,t) \in S_1^2 \\ c \displaystyle\sum_{u,v \in S} I(A(s,t))(u,v)\, d(u,v) & \text{otherwise.} \end{cases}$$

▶ **Proposition 19.** For all $A \in \mathcal{A}$, $I \in \mathcal{I}$ and $c \in (0,1]$, the function $\Gamma_c^{A,I}$ is monotone and c-Lipschitz.

From Theorem 3(c) we can conclude that $\Gamma_c^{A,I}$ has a least fixed point, which we denote by $\boldsymbol{\mu}\Gamma_c^{A,I}$. In the remainder of this section we will show that there exist an optimal max policy A^* and an optimal min policy I^* such that the corresponding value function captures the probabilistic bisimilarity distances. In the above game graph, the red edge represents the optimal max policy and the blue edges represent the optimal min policy. The proof of $\boldsymbol{\mu}\Delta_1 = \boldsymbol{\mu}\Gamma_1^{A^*,I^*}$ consists of two parts. First, we prove that there exists an optimal min policy.

▶ **Lemma 20.** $\exists I \in \mathcal{I} : \forall A \in \mathcal{A} : \boldsymbol{\mu}\Gamma_1^{A,I} \sqsubseteq \boldsymbol{\mu}\Delta_1$.

Proof. Towards the construction of $I^* \in \mathcal{I}$, let $s \in S$ and $\nu \in \mathit{Distr}(S)$. Since we restrict our attention to finitely branching probabilistic automata,

$$\mu_{s,\nu} = \operatorname*{argmin}_{s \to \mu} K(\boldsymbol{\mu}\Delta_1)(\mu,\nu) \tag{1}$$

exists. Because the set $V(\Omega(\mu_{s,\nu},\nu))$ is nonempty and finite, we can define

$$I^*(s,\nu) = \operatorname*{argmin}_{\omega \in V(\Omega(\mu_{s,\nu},\nu))} \sum_{u,v \in S} \omega(u,v)\, \boldsymbol{\mu}\Delta_1(u,v). \tag{2}$$

By construction $I^* \in \mathcal{I}$.

Let $A \in \mathcal{A}$. Since $\boldsymbol{\mu}\Gamma_1^{A,I^*}$ is the least pre-fixed point of Γ_1^{A,I^*} according to Theorem 3(d), to conclude that $\boldsymbol{\mu}\Gamma_1^{A,I^*} \sqsubseteq \boldsymbol{\mu}\Delta_1$ it suffices to show that $\boldsymbol{\mu}\Delta_1$ is a pre-fixed point of Γ_1^{A,I^*}, that is, $\Gamma_1^{A,I^*}(\boldsymbol{\mu}\Delta_1) \sqsubseteq \boldsymbol{\mu}\Delta_1$. Let $s,t \in S$. We distinguish three cases.

- If $(s,t) \in S_0^2$, then

$$\begin{aligned} \Gamma_1^{A,I^*}(\boldsymbol{\mu}\Delta_1)(s,t) &= 0 \\ &= \boldsymbol{\mu}\Delta_1(s,t) \qquad \text{[Theorem 16]} \end{aligned}$$

- If $(s,t) \in S_1^2$, then

$$\begin{aligned} \Gamma_1^{A,I^*}(\boldsymbol{\mu}\Delta_1)(s,t) &= 1 \\ &= \Delta_1(\boldsymbol{\mu}\Delta_1)(s,t) \\ &= \boldsymbol{\mu}\Delta_1(s,t). \end{aligned}$$

- Otherwise, $(s,t) \in S_?^2$. Without any loss of generality, we assume that $A(s,t) = (s,\nu)$

with $t \to \nu$. Then

$$\begin{aligned}
\Gamma_1^{A,I^*}(\boldsymbol{\mu}\Delta_1)(s,t) &= \sum_{u,v\in S} I^*(A(s,t))(u,v)\,\boldsymbol{\mu}\Delta_1(u,v)\\
&= \sum_{u,v\in S} I^*(s,\nu)(u,v)\,\boldsymbol{\mu}\Delta_1(u,v) \qquad [A(s,t)=(s,\nu)]\\
&= \min_{\omega\in V(\Omega(\mu_{s,\nu},\nu))} \sum_{u,v\in S} \omega(u,v)\,\boldsymbol{\mu}\Delta_1(u,v) \qquad [(2)]\\
&= K(\boldsymbol{\mu}\Delta_1)(\mu_{s,\nu},\nu)\\
&= \min_{s\to\mu} K(\boldsymbol{\mu}\Delta_1)(\mu,\nu) \qquad [(1)]\\
&\leq \max_{t\to\nu}\min_{s\to\mu} K(\boldsymbol{\mu}\Delta_1)(\mu,\nu)\\
&\leq H(K(\boldsymbol{\mu}\Delta_1))(\{\,\mu \mid s\to\mu\,\},\{\,\nu \mid t\to\nu\,\})\\
&= \Delta_1(\boldsymbol{\mu}\Delta_1)(s,t)\\
&= \boldsymbol{\mu}\Delta_1(s,t).
\end{aligned}$$ ◀

In the remainder of this paper, we denote the optimal min policy constructed in the above proof by I^*. It remains to prove that there exists an optimal max policy. The proof of this second part turns out to be more involved than the proof of the first part contained in above lemma. The proof has the following three major components.

- For all $A \in \mathcal{A}$ and $I \in \mathcal{I}$, the value function $\boldsymbol{\mu}\Gamma_1^{A,I}$ is the limit of the discounted value functions $\boldsymbol{\mu}\Gamma_c^{A,I}$. This result is inspired by [14, Theorem 4.4.1].
- Similarly, the probabilistic bisimilarity distances captured by $\boldsymbol{\mu}\Delta_1$ are the limit of their discounted counterparts represented by $\boldsymbol{\mu}\Delta_c$.
- There exists an optimal max policy in the discounted setting.

Combining the above three components, we arrive at an optimal max policy. The first two components are formalized next.

▶ **Proposition 21.** For all $A \in \mathcal{A}$ and $I \in \mathcal{I}$, $\lim_{c\uparrow 1} \boldsymbol{\mu}\Gamma_c^{A,I} = \boldsymbol{\mu}\Gamma_1^{A,I}$ and $\lim_{c\uparrow 1} \boldsymbol{\mu}\Delta_c = \boldsymbol{\mu}\Delta_1$.

The major component of the proof consists of showing that there exists an optimal max policy in the discounted setting.

▶ **Proposition 22.** For all $c \in (0,1)$, $\exists A \in \mathcal{A} : \forall I \in \mathcal{I} : \boldsymbol{\mu}\Delta_c \sqsubseteq \boldsymbol{\mu}\Gamma_c^{A,I}$.

Proof. Let $c \in (0,1)$. Let s, $t \in S$. If

$$\max_{s\to\mu}\min_{t\to\nu} K(\boldsymbol{\mu}\Delta_c)(\mu,\nu) \geq \max_{t\to\nu}\min_{s\to\mu} K(\boldsymbol{\mu}\Delta_c)(\mu,\nu) \tag{3}$$

then we define $A_c^*(s,t)$ by

$$A_c^*(s,t) = \left(t, \operatorname*{argmax}_{s\to\mu}\min_{t\to\nu} K(\boldsymbol{\mu}\Delta_c)(\mu,\nu)\right).$$

Because the probabilistic automaton is finitely branching, the above exists. Otherwise, we define $A_c^*(s,t)$ by

$$A_c^*(s,t) = \left(s, \operatorname*{argmax}_{t\to\nu}\min_{s\to\mu} K(\boldsymbol{\mu}\Delta_c)(\mu,\nu)\right).$$

By construction, $A_c^* \in \mathcal{A}$.

Let $I \in \mathcal{I}$. Since $\langle [0,1]^{S \times S}, \| \cdot - \cdot \| \rangle$ is a nonempty complete metric space according to Proposition 6 and the function $\Gamma_c^{A_c^*,I}$ is contractive by Proposition 19, we can conclude from Theorem 7 that $\Gamma_c^{A_c^*,I}$ has a unique fixed point. Therefore, $\boldsymbol{\mu}\Gamma_c^{A_c^*,I}$ is not only the least fixed point but also the greatest fixed point of $\Gamma_c^{A_c^*,I}$. According to Theorem 3(b), $\boldsymbol{\mu}\Gamma_c^{A_c^*,I}$ is the greatest post-fixed point of $\Gamma_c^{A_c^*,I}$. Hence, to conclude that $\boldsymbol{\mu}\Delta_c \sqsubseteq \boldsymbol{\mu}\Gamma_c^{A_c^*,I}$ it suffices to show that $\boldsymbol{\mu}\Delta_c$ is a post-fixed point of $\Gamma_c^{A_c^*,I}$, that is, $\boldsymbol{\mu}\Delta_c \sqsubseteq \Gamma_c^{A_c^*,I}(\boldsymbol{\mu}\Delta_c)$. Let s, $t \in S$. We distinguish three cases.

- If $(s,t) \in S_0^2$, then

$$\begin{aligned}\boldsymbol{\mu}\Delta_c(s,t) &\leq \boldsymbol{\mu}\Delta_1(s,t)\\ &= 0 \qquad \text{[Theorem 16]}\\ &= \Gamma_c^{A_c^*,I}(\boldsymbol{\mu}\Delta_c)(s,t).\end{aligned}$$

- If $(s,t) \in S_1^2$, then

$$\begin{aligned}\boldsymbol{\mu}\Delta_c(s,t) &= \Delta_c(\boldsymbol{\mu}\Delta_c)(s,t)\\ &= 1\\ &= \Gamma_c^{A_c^*,I}(\boldsymbol{\mu}\Delta_c)(s,t).\end{aligned}$$

- Otherwise, $(s,t) \in S_?^2$. Without loss of any generality, assume that $A_c^*(s,t) = (t,\mu)$. This assumption implies that (3) and

$$\Delta_1(\boldsymbol{\mu}\Delta_c)(s,t) = \min_{t \to \nu} K(\boldsymbol{\mu}\Delta_c)(\mu,\nu). \tag{4}$$

  Hence,

$$\begin{aligned}\boldsymbol{\mu}\Delta_c(s,t) &= \Delta_c(\boldsymbol{\mu}\Delta_c)(s,t)\\ &= c\,\Delta_1(\boldsymbol{\mu}\Delta_c)(s,t)\\ &= c \min_{t \to \nu} K(\boldsymbol{\mu}\Delta_c)(\mu,\nu) \qquad [(4)]\\ &\leq c \sum_{u,v \in S} I(A_c^*(s,t))(u,v)\,\boldsymbol{\mu}\Delta_c(u,v)\\ &= c\,\Gamma_1^{A_c^*,I}(\boldsymbol{\mu}\Delta_c)(s,t)\\ &= \Gamma_c^{A_c^*,I}(\boldsymbol{\mu}\Delta_c)(s,t).\end{aligned}$$ ◀

Combining the above three components, we obtain the second part of the proof.

▶ **Lemma 23.** $\exists A \in \mathcal{A} : \forall I \in \mathcal{I} : \boldsymbol{\mu}\Delta_1 \sqsubseteq \boldsymbol{\mu}\Gamma_1^{A,I}$.

Proof. According to Proposition 22,

$$\forall n \in \mathbb{N} : \exists A_n \in \mathcal{A} : \forall I \in \mathcal{I} : \boldsymbol{\mu}\Delta_{\frac{n}{n+1}} \sqsubseteq \boldsymbol{\mu}\Gamma_{\frac{n}{n+1}}^{A_n,I}. \tag{5}$$

Since the set $\mathcal{A}$ is finite, the sequence $(A_n)_{n \in \mathbb{N}}$ has a subsequence $(A_{\sigma(n)})_{n \in \mathbb{N}}$ that is constant, that is, there exists $A^* \in \mathcal{A}$ such that for all $n \in \mathbb{N}$, $A_{\sigma(n)} = A^*$. From Proposition 21 we can conclude that

$$\lim_{n \in \mathbb{N}} \boldsymbol{\mu}\Delta_{\frac{\sigma(n)}{\sigma(n)+1}} = \boldsymbol{\mu}\Delta_1 \text{ and } \lim_{n \in \mathbb{N}} \boldsymbol{\mu}\Gamma_{\frac{\sigma(n)}{\sigma(n)+1}}^{A,I} = \boldsymbol{\mu}\Gamma_1^{A,I}.$$

From (5) we can deduce that $\forall I \in \mathcal{I} : \boldsymbol{\mu}\Delta_1 \sqsubseteq \boldsymbol{\mu}\Gamma_1^{A^*,I}$. ◀

In the remainder of this paper, we denote the optimal max policy that satisfies Lemma 23 by A^*. Combining Lemma 20 and 23, we arrive at the following alternative characterization of the probabilistic bisimilarity distances.

▶ **Theorem 24.** $\boldsymbol{\mu}\Delta_1 = \boldsymbol{\mu}\Gamma_1^{A^*,I^*}$.

5 Deciding Distance One

In this section, we present an algorithm to compute the set D_1 of state pairs that have distance one, that is

$$D_1 = \{\, (s,t) \in S \times S \mid \boldsymbol{\mu}\Delta_1(s,t) = 1 \,\}.$$

The key ingredient of our algorithm is the following function.

▶ **Definition 25.** The function $\Lambda : 2^{S\times S} \times 2^{S\times S} \to 2^{S\times S}$ is defined by

$$\Lambda(X,Y) = S_1^2 \cup \left\{ (s,t) \in S_?^2 \;\middle|\; \begin{array}{l} \exists s \to \mu : \forall t \to \nu : \forall \omega \in V(\Omega(\mu,\nu)) : \\ \qquad \mathrm{support}(\omega) \subseteq X \wedge \mathrm{support}(\omega) \cap Y \neq \emptyset \vee \\ \exists t \to \nu : \forall s \to \mu : \forall \omega \in V(\Omega(\mu,\nu)) : \\ \qquad \mathrm{support}(\omega) \subseteq X \wedge \mathrm{support}(\omega) \cap Y \neq \emptyset \end{array} \right\}.$$

The set $\Lambda(X,Y)$ contains all state pairs with different labels and those state pairs for which there exists a move by the max player so that every subsequent move of the min player always ends up in X and with some positive probability in Y. The function Λ has the following monotonicity properties.

▶ **Proposition 26.** For all X, Y, $Z \subseteq S \times S$ with $X \subseteq Y$,
(a) $\Lambda(Z,X) \subseteq \Lambda(Z,Y)$.
(b) $\boldsymbol{\mu}Z.\Lambda(X,Z) \subseteq \boldsymbol{\mu}Z.\Lambda(Y,Z)$.

Since $\langle 2^{S\times S}, \subseteq\rangle$ is a complete lattice and for each $X \subseteq S \times S$ the function $\boldsymbol{\lambda}Y.\Lambda(X,Y)$ is monotone, the least fixed point $\boldsymbol{\mu}Y.\Lambda(X,Y)$ exists according to Theorem 3(c). The set $\boldsymbol{\mu}Y.\Lambda(X,Y)$ contains all state pairs (s,t) for which there exists a max policy such that for all min policies, (s,t) can reach a state pair with different labels and all state pairs reachable from (s,t) are element of X.

Since the function $\boldsymbol{\lambda}X.\boldsymbol{\mu}Y.\Lambda(X,Y)$ is monotone as well, we can conclude from Theorem 3(a) that the greatest fixed point $\boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X,Y)$ exists. The set $\boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X,Y)$ contains all state pairs (s,t) for which there exists a max policy such that for all min policies, all state pairs reachable from (s,t) can reach a state pair with different labels. In the next section, we will prove that $\boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X,Y)$ captures the set D_1. According to Theorem 4(a) and (b), these greatest and least fixed points can be obtained iteratively as follows.

```
$X_c = S \times S$
do
  $Y_c = \emptyset$
  do
    $Y_p = Y_c$
    $Y_c = \Lambda(X_c, Y_p)$
  while $Y_p \neq Y_c$
  $X_p = X_c$
  $X_c = Y_c$
while $X_p \neq X_c$
```

The inner loop (line 3–7) computes the least fixed point $\boldsymbol{\mu}Y.\Lambda(X_c, Y)$. The outer loop (line 1–10) computes the greatest fixed point $\boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X, Y)$, which equals D_1 as we will prove in the next section. Due to the monotonicity of Λ we can conclude that both the inner and outer loop terminate after at most $|S|^2$ iterations. To conclude that the above algorithm is polynomial time, it remains to show that $\Lambda(X_c, Y_p)$ in line 6 can be computed in polynomial time.

▶ **Proposition 27.** For all μ, $\nu \in \mathit{Distr}(S)$ and $X \subseteq S \times S$,

$$\forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq X \text{ if and only if } K(d)(\mu, \nu) = 1$$

and

$$\forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \cap X \neq \emptyset \text{ if and only if } K(d)(\mu, \nu) > 0$$

where

$$d(s, t) = \begin{cases} 1 & \text{if } (s, t) \in X \\ 0 & \text{otherwise.} \end{cases}$$

Computing $K(d)(\mu, \nu)$ boils down to solving a minimum cost network flow problem, where d captures the cost. This problem can be solved in polynomial time using, for example, Orlin's network simplex algorithm [26]. Hence, $\Lambda(X_c, Y_p)$ can be computed in polynomial time.

6 Correctness Proof

To conclude that the algorithm presented in the previous section is correct, it remains to show that $\boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X, Y)$ equals D_1. We start by providing an iterative characterization of $\boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X, Y)$.

▶ **Definition 28.** For each $i \in \mathbb{N}$, the set $X_i \subseteq S \times S$ is defined by

$$X_i = \begin{cases} S \times S & \text{if } i = 0 \\ \boldsymbol{\mu}Y.\Lambda(X_{i-1}, Y) & \text{otherwise.} \end{cases}$$

For each i, $j \in \mathbb{N}$, the set $Y_i^j \subseteq S \times S$ is defined by

$$Y_i^j = \begin{cases} D_1 & \text{if } j = 0 \\ \Lambda(X_i, Y_i^{j-1}) & \text{otherwise.} \end{cases}$$

The above definition differs from the iterative algorithm presented in the previous section in that $Y_i^0 = D_1$ whereas the algorithm starts its iteration towards the least fixed point from $\emptyset$.

▶ **Proposition 29.**

(a) $X_m = \boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X, Y)$ for some $m \in \mathbb{N}$.

(b) $Y_m^n = \boldsymbol{\mu}Y.\Lambda(X_m, Y)$ for some $n \in \mathbb{N}$.

(c) $X_m = Y_m^n$.

Proof sketch. Part (a) follows from Theorem 4(b) and Proposition 26(b). Part (b) can be proved as follows. First, we observe that

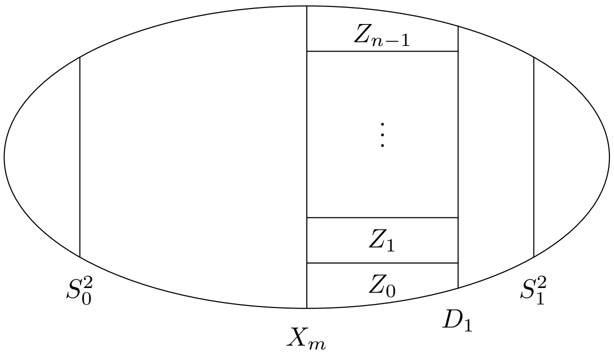
$$D_1 \subseteq \boldsymbol{\mu}Y.\Lambda(X_m, Y) = X_m \tag{6}$$

by part (a). The desired result follows from the latter fact and Theorem 4(c) and Proposition 26(a). Part (c) follows from part (a) and (b). ◀

From part (a) of the above proposition and (6) we can conclude that it suffices to prove $X_m \subseteq D_1$.

▶ **Definition 30.** For each $0 \leq i < n$, the set $Z_i \subseteq S \times S$ is defined by

$$Z_i = Y_m^{i+1} \setminus Y_m^i.$$



▶ **Proposition 31.**
(a) For all $0 \leq i < n$, $Z_i \subseteq S_?^2$.
(b) For all $0 \leq i < j < n$, $Z_i \cap Z_j = \emptyset$.
(c) $\bigcup_{0 \leq i < n} Z_i = X_m \setminus D_1$.
(d) For all $0 \leq i \leq n$, $Y_m^i = D_1 \cup \bigcup_{0 \leq j < i} Z_j$.

According to Proposition 31(b) and (c), the sets $Z_0, \ldots, Z_{n-1}$ form a partition of $X_m \setminus D_1$.

▶ **Proposition 32.** For all $0 \leq i < n$ and $(s, t) \in Z_i$,

$$\exists s \to \mu : \forall t \to \nu : \forall \omega \in V(\Omega(\mu, \nu)) : \text{support}(\omega) \subseteq X_m \wedge \text{support}(\omega) \cap Y_m^i \neq \emptyset \vee \tag{7}$$
$$\exists t \to \nu : \forall s \to \mu : \forall \omega \in V(\Omega(\nu, \mu)) : \text{support}(\omega) \subseteq X_m \wedge \text{support}(\omega) \cap Y_m^i \neq \emptyset \tag{8}$$

Based on the above proposition, we construct a max policy A'.

▶ **Definition 33.** The function $A' : S_?^2 \to (S \times \mathit{Distr}(S))$ is defined by

$$A'(s,t) = \begin{cases} (t, \mu) & \text{if } (s,t) \in Z_i \text{ and } (7) \\ (s, \nu) & \text{if } (s,t) \in Z_i \text{ and } (8) \\ A^*(s,t) & \text{if } (s,t) \in S_?^2 \setminus (X_m \setminus D_1). \end{cases}$$

Given the max policy A' and an arbitrary min policy I, from Proposition 31(d) and 32 we can conclude that each state pair in Z_i can reach a state pair in D_1 or Z_j with $j < i$. Consequently, each state pair in Z_i can reach a state pair in D_1. Given the max policy A' and the optimal min policy I^*, we define the function Ψ as follows.

▶ **Definition 34.** The function $\Psi : [0,1]^{S \times S} \to [0,1]^{S \times S}$ is defined by

$$\Psi(d)(s,t) = \begin{cases} \Gamma_1^{A',I^*}(d)(s,t) & \text{if } (s,t) \in X_m \\ 0 & \text{otherwise} \end{cases}$$

▶ **Proposition 35.** The function Ψ is monotone.

Since $\langle [0,1]^{S \times S}, \sqsubseteq \rangle$ is a complete lattice and Ψ is monotone, Ψ has a greatest fixed point $\boldsymbol{\nu}\Psi$ and a least fixed point $\boldsymbol{\mu}\Psi$ by Theorem 3(a) and (c). Next, we will show that Ψ has a unique fixed point.

▶ **Proposition 36.** Ψ has a unique fixed point.

Proof sketch. It is sufficient to prove that $\boldsymbol{\mu}\Psi = \boldsymbol{\nu}\Psi$. Let

$$\begin{aligned} m &= \max\{\, \boldsymbol{\nu}\Psi(s,t) - \boldsymbol{\mu}\Psi(s,t) \mid (s,t) \in S \times S \,\} \\ M &= \{\, (s,t) \in S \times S \mid \boldsymbol{\nu}\Psi(s,t) - \boldsymbol{\mu}\Psi(s,t) = m \,\} \end{aligned}$$

We can show that $m = 0$ and, hence, we can conclude that $\boldsymbol{\mu}\Psi = \boldsymbol{\nu}\Psi$. ◀

From the fact that Ψ has a unique fixed point and the alternative characterization of the probabilistic bisimilarity distances presented in the previous section, we can infer the main result of this section.

▶ **Theorem 37.** $D_1 = \boldsymbol{\nu}X.\boldsymbol{\mu}Y.\Lambda(X,Y)$.

Proof sketch. We can show that the function $d \in S \times S \to [0,1]$ defined by

$$d(s,t) = \begin{cases} 1 & \text{if } (s,t) \in X_m \\ 0 & \text{otherwise} \end{cases}$$

is a fixed point of Ψ. Let $(s,t) \in X_m$. Then

$$\begin{aligned} \boldsymbol{\mu}\Delta_1(s,t) &\geq \boldsymbol{\mu}\Gamma_1^{A',I^*}(s,t) && \text{[Lemma 20]} \\ &= \boldsymbol{\mu}\Psi(s,t) && [(s,t) \in X_m] \\ &= d(s,t) && [d \text{ is a fixed point of } \Psi \text{ and Proposition 36}] \\ &= 1 && [(s,t) \in X_m] \end{aligned}$$

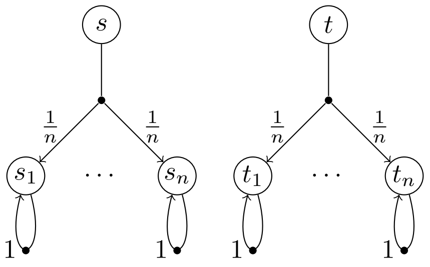
Hence, $(s,t) \in D_1$. Therefore, $X_m \subseteq D_1$. According to (6), $D_1 \subseteq X_m$. Thus, $X_m = D_1$. Proposition 29(a) completes the proof. ◀

7 Conclusion

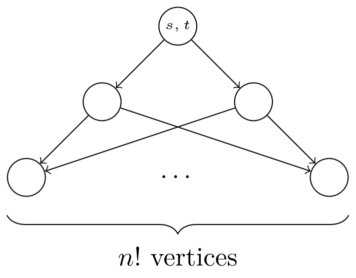
Chen et al. [9] have provided an alternative characterization of the probabilistic bisimilarity distances for labelled Markov chains. This characterization forms the basis for the algorithm to compute the probabilistic bisimilarity distances for labelled Markov chains by Bacci et al. [1]. Their algorithm is similar to Howard's policy iteration algorithm [20]. In this paper we have presented an alternative characterization of the probabilistic bisimilarity distances for probabilistic automata. In future work, we plan to use this characterization as the foundation for an algorithm to compute the probabilistic bisimilarity distances for probabilistic automata based on the policy iteration algorithm due to Hoffman and Karp [19].

As shown by Baier [2], probabilistic bisimilarity distance zero for probabilistic automata can be decided in polynomial time. In this paper we have shown that distance one can also be decided in polynomial time. As a consequence, we can determine in polynomial time how many, if any, distances are non-trivial, that is, greater than zero and smaller than one. As we have already shown in [32] in the context of labelled Markov chains, being able to decide distance zero and distance one in polynomial time has significant impact on computing probabilistic bisimilarity distances for labelled Markov chains. The algorithm by Bacci et al. [1], that does not decide distance one before computing the non-trivial distances using policy iteration, can compute distances for labelled Markov chains up to 150 states. For one such labelled Markov chain, their algorithm takes more than 49 hours. Our algorithm that we present in [32] decides distance zero and distance one before using policy iteration to compute the non-trivial distances. Our algorithm takes 13 milliseconds instead of 49 hours. Furthermore, our algorithm can compute distances for labelled Markov chains with more than 10,000 states in less than 50 minutes.

Consider the following probabilistic automaton.



This probabilistic automaton induces the following game graph.



If μ and ν are both the uniform distribution on n elements, then the vertices of $\Omega(\mu, \nu)$ can be viewed as permutations (see, for example, [29, Theorem 8.4]). As a result, from the state pair (s, t) after one move by the max player and one move by the min player, $n!$ vertices can be reached. Hence, we may encounter an exponential blow-up when we transform a probabilistic automaton into a game. As a consequence, it is not immediately obvious which results from game theory can be transferred to our setting. We leave this for future research.

To prove Lemma 23, which provides the second part of the proof of the alternative characterization of the probabilistic bisimilarity distances, we rely on the discounted functions Δ_c and $\Gamma_c^{A_c^*,I}$ for $c \in (0,1)$. In particular, in the proof of Proposition 22 we use the fact that $\Gamma_c^{A_c^*,I}$ has a unique fixed point. If we were able to prove that $\Gamma_1^{A^*,I}$ has a unique fixed point, then we would be able to give a proof of Lemma 23 that does not rely on discounted functions. We also leave that for future research.

References

1 Giorgio Bacci, Giovanni Bacci, Kim Larsen, and Radu Mardare. On-the-fly exact computation of bisimilarity distances. In Nir Piterman and Scott Smolka, editors, *Proceedings of the 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 7795 of *Lecture Notes in Computer Science*, pages 1–15, Rome, Italy, 2013. Springer-Verlag.

2 Christel Baier. Polynomial time algorithms for testing probabilistic bisimulation and simulation. In Rajeev Alur and Thomas Henzinger, editors, *Proceedings of the 8th International Conference on Computer Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 50–61, New Brunswick, NJ, USA, 1996. Springer-Verlag.

3 Jaco de Bakker and Erik de Vink. *Control flow semantics*. MIT Press, Cambridge, MA, USA, 1996.

4 Jaco de Bakker and Jeffery Zucker. Denotational semantics of concurrency. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*, pages 153–158, San Francisco, 1982. ACM.

5 Stefan Banach. Sur les opérations dans les ensembles abstraits et leurs applications aux equations intégrales. *Fundamenta Mathematicae*, 3:133–181, 1922.

6 Franck van Breugel. Probabilistic bisimilarity distances. *SIGLOG News*, 4(4):33–51, 2017.

7 Franck van Breugel and James Worrell. Towards quantitative verification of probabilistic systems. In Fernando Orejas, Paul Spirakis, and Jan van Leeuwen, editors, *Proceedings of 28th International Colloquium on Automata, Languages and Programming*, volume 2076 of *Lecture Notes in Computer Science*, pages 421–432, Crete, 2001. Springer-Verlag.

8 Franck van Breugel and James Worrell. The complexity of computing a bisimilarity pseudometric on probabilistic automata. In Franck van Breugel, Elham Kashefi, Catuscia Palamidessi, and Jan Rutten, editors, *Horizons of the Mind – A Tribute to Prakash Panangaden*, volume 8464 of *Lecture Notes in Computer Science*, pages 191–213. Springer-Verlag, 2014.

9 Di Chen, Franck van Breugel, and James Worrell. On the complexity of computing probabilistic bisimilarity. In Lars Birkedal, editor, *Proceedings of the 15th International Conference on Foundations of Software Science and Computational Structures*, volume 7213 of *Lecture Notes in Computer Science*, pages 437–451, Tallinn, Estonia, 2012. Springer-Verlag.

10 Edmund Clarke, Orna Grumberg, and Doron Peled. *Model checking.* MIT Press, Cambridge, MA, USA, 1999.

11 Brian Davey and Hilary Priestley. *Introduction to lattices and order.* Cambridge University Press, Cambridge, United Kingdom, 2002.

12 Yuxin Deng, Tom Chothia, Catuscia Palamidessi, and Jun Pang. Metrics for action-labelled quantitative transition systems. In Antonio Cerone and Herbert Wiklicky, editors, *Proceedings of the 3rd Workshop on Quantitative Aspects of Programming Languages*, volume 153(2) of *Electronic Notes in Theoretical Computer Science*, pages 79–96, Edinburgh, Scotland, 2005. Elsevier.

13 Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science*, pages 413–422, Copenhagen, Denmark, 2002. IEEE.

14 Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes.* Springer-Verlag, New York, NY, USA, 1997.

15 Hongfei Fu. Computing game metrics on Markov decision processes. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Proceedings of the 39th International Colloquium on Automata, Languages, and Programming*, volume 7392 of *Lecture Notes in Computer Science*, pages 227–238, Warwick, UK, 2012. Springer-Verlag.

16 Hongfei Fu. Personal communication, 2013.

17 Alessandro Giacalone, Chi-Chang Jou, and Scott 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, Sea of Gallilee, Israel, 1990. North-Holland.

18 Felix Hausdorff. *Grundzüge der Mengenlehre.* Von Veit & Comp., Leipzig, 1914.

19 Alan Hoffman and Richard Karp. On nonterminating stochastic games. *Management Science*, 12(5):359–370, 1966.

20 Ronald Howard. *Dynamic Programming and Markov Processes.* The MIT Press, Cambridge, MA, USA, 1960.

21 Bengt Jonsson and Kim Larsen. Specification and refinement of probabilistic processes. In *Proceedings of the 6th Annual Symposium on Logic in Computer Science*, pages 266–277, Amsterdam, The Netherlands, 1991. IEEE.

22 Leonid Kantorovich. On the transfer of masses (in Russian). *Doklady Akademii Nauk*, 5(1):1–4, 1942. Translated in *Management Science*, 5(1):1–4, 1958.

23 Viktor Klee and Christoph Witzgall. Facets and vertices of transportation polytopes. In George Dantzig and Arthur Veinott, editors, *Proceedings of 5th Summer Seminar on the Mathematis of the Decision Sciences*, volume 11 of *Lectures in Applied Mathematics*, pages 257–282, Stanford, CA, USA, 1967. AMS.

24 Bronisław Knaster. Un théorème sur les fonctions d'ensembles. *Annales de la Société Polonaise de Mathématique*, 6:133–134, 1928.

25 Kim Larsen and Arne Skou. Bisimulation through probabilistic testing. In *Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages*, pages 344–352, Austin, TX, USA, 1989. ACM.

26 James Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(2):109–129, 1997.

27 Roberto Segala. *Modeling and verification of randomized distributed real-time systems*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

28 Roberto Segala and Nancy Lynch. Probabilistic simulations for probabilistic processes. In Bengt Jonsson and Joachim Parrow, editors, *Proceedings of the 5th International Conference on Concurrency Theory*, volume 836 of *Lecture Notes in Computer Science*, pages 481–496, Uppsala, Sweden, 1994. Springer-Verlag.

29 Denis Serre. *Matrices: theory and applications*. Springer-Verlag, New York, NY, USA, 2010.

30 Colin Stirling. Bisimulation, modal logic and model checking games. *Logic Journal of the IGPL*, 7(1):103–124, 1999.

31 Qiyi Tang and Franck van Breugel. Computing probabilistic bisimilarity distances via policy iteration. In Josée Desharnais and Radha Jagadeesan, editors, *Proceedings of the 27th International Conference on Concurrency Theory*, volume 59 of *Leibniz International Proceedings in Informatics*, pages 22:1–22:15, Quebec City, QC, Canada, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

32 Qiyi Tang and Franck van Breugel. Deciding probabilistic bisimilarity distance one for labelled Markov chains. In Hana Chockler and Georg Weissenbacher, editors, *Proceedings of the 30th International Conference on Computer Aided Verification*, volume 10981 of *Lecture Notes in Computer Science*, pages 681–699, Oxford, UK, 2018. Springer-Verlag.

33 Alfred Tarski. A lattice-theoretic fixed point theorem and its applications. *Pacific Journal of Mathematics*, 5(2):285–309, 1955.

34 Mathieu Tracol, Josée Desharnais, and Abir Zhioua. Computing distances between probabilistic automata. In Mieke Massink and Gethin Norman, editors, *Proceedings 9th Workshop on Quantitative Aspects of Programming Languages*, volume 57 of *Electronic Proceedings in Theoretical Computer Science*, pages 148–162, Saarbrücken, Germany, 2011. Elsevier.