On the Complexity of Computing Probabilistic Bisimilarity

Di Chen[1], Franck van Breugel[2,⋆], and James Worrell[1,⋆⋆]

[1] Department of Computer Science, University of Oxford, UK
[2] Department of Computer Science and Engineering, York University, Canada

Abstract. Probabilistic bisimilarity is a fundamental notion of equivalence on labelled Markov chains. It has a natural generalisation to a probabilistic bisimilarity pseudometric, whose definition involves the Kantorovich metric on probability distributions. The pseudometric has discounted and undiscounted variants, according to whether one discounts the future in observing discrepancies between states.

This paper is concerned with the complexity of computing probabilistic bisimilarity and the probabilistic bisimilarity pseudometric on labelled Markov chains. We show that the problem of computing probabilistic bisimilarity is **P**-hard by reduction from the monotone circuit value problem. We also show that the discounted pseudometric is rational and can be computed exactly in polynomial time using the network simplex algorithm and the continued fraction algorithm. In the undiscounted case we show that the pseudometric is again rational and can be computed exactly in polynomial time using the ellipsoid algorithm. Finally, using the notion of couplings on Markov chains, we show that the pseudometric can be used to compute bounds on the variational distance of trace distributions, which is **NP**-hard to compute directly.

1 Introduction

Probabilistic bisimilarity is a notion of equivalence for probabilistic labelled transition systems, introduced by Larsen and Skou [21]. It is based on Park and Milner's classical notion of bisimilarity for (non-deterministic) labelled transition systems [23]. A very similar and widely used concept on Markov chains, called *lumpability*, can be found as far back as the classical text of Kemeny and Snell [20]. A system and its probabilistic bisimilarity quotient can be considered indistinguishable, and quotienting by probabilistic bisimilarity is a widely used compression technique in verification and performance analysis [18,19].

The first part of this paper concerns the complexity of computing probabilistic bisimilarity. It is known that this can be done in polynomial time, e.g., by partition refinement [2,11,32]. Our first result shows that probabilistic bisimilarity is **P**-hard, and therefore **P**-complete. As a consequence probabilistic bisimilarity

⋆ Supported by NSERC and the Leverhulme Trust.

⋆⋆ Supported by the EPSRC.

L. Birkedal (Ed.): FOSSACS 2012, LNCS 7213, pp. 437–451, 2012.

is not in **NC** unless **P** = **NC**. (Recall that **NC** is a subclass of **P** comprising problems that can be solved in polylogarithmic time using PRAMs of polynomial size [16]. Informally such problems are considered to be efficiently parallelisable.) By contrast, language equivalence of probabilistic automata is in **NC** [31], as are related equivalence problems such as tree isomorphism [16].

For (non-deterministic) labelled transition systems it is known that computing bisimilarity is **P**-complete [4,26]. However the proof in the probabilistic case requires a different construction than in *op. cit.*

For probabilistic systems it is natural to generalise from bivalent notions of equivalence, such as probabilistic bisimilarity or language equivalence [30], to quantitative measures of similarity. As well as being more informative, such measures are more meaningful in the presence of rounding errors in computation and modelling (see, for example, [15]).

In the second part of this paper we consider a *probabilistic bisimilarity pseudometric* on labelled Markov chains. This generalises the notion of probabilistic bisimilarity by assigning a *similarity distance* to pairs of states of a labelled Markov chain. The smaller the distance, the more alike the states, with states at zero distance if and only if they are probabilistic bisimilar. This pseudometric was first introduced in [12] and, together with closely related notions, has subsequently been studied in the context of systems biology [28], games [9], planning [10] and security [8], among others. The definition of the pseudometric is based on the classical Kantorovich metric on probability distributions. The pseudometric has discounted versions, which discount the future in observing discrepancies between states.

We show that for labelled Markov chains with rational transition probabilities the discounted probabilistic bisimilarity pseudometric is rational and can be computed exactly by a polynomial-time algorithm. In particular, we show that the distances can be approximated by using the network simplex algorithm repeatedly and the exact distances can be obtained from the approximated ones by means of the continued fraction algorithm. In the undiscounted case we also obtain a polynomial-time algorithm to exactly compute the pseudometric, this time using the heavier machinery of the ellipsoid algorithm. In combination with our lower bound on computing probabilistic bisimilarity we conclude that computing the pseudometric is **P**-complete. These results go beyond previous work which only showed how to approximate the pseudometric up to some desired level of precision [7]. In the undiscounted case it was only known how to approximate the pseudometric using polynomial space [6]. We use the notion of couplings of Markov chains to show that the pseudometric is an upper bound on the variational distance between the trace distributions generated by states of the Markov chain, which is **NP**-hard to compute directly [22].

Fu [14] shows that the complexity of approximating a bisimilarity pseudometric on probabilistic automata, which generalise labelled Markov chains, lies in the intersection of **NP** and **coNP**. Even more general than probabilistic automata are stochastic games. A generalisation of the bisimilarity pseudometric from labelled Markov chains to stochastic games has been shown to be as hard as the sum-of-square-roots problem [9], a problem not known even to be in **NP**.

2 Probabilistic Bisimilarity

In this section we introduce labelled Markov chains and probabilistic bisimilarity, and we show that computing probabilistic bisimilarity is **P**-hard.

A *labelled Markov chain* is a tuple $\mathcal{M} = (S, \Sigma, \pi, \ell)$ consisting of a finite set of *states* S, a finite set of *labels* Σ, a rational *transition matrix* π such that $\sum_{t \in S} \pi_{s,t} = 1$ for all $s \in S$, and a *labelling function* $\ell : S \to \Sigma$.

A *probabilistic bisimulation* on $\mathcal{M}$ is an equivalence relation $R \subseteq S \times S$ such that if $s \mathrel{R} t$ then $\ell(s) = \ell(t)$ and $\sum_{u \in E} \pi_{s,u} = \sum_{u \in E} \pi_{t,u}$ for each R-equivalence class E, i.e., related states have the same label and the same probability to transition into any given equivalence class. It is a standard result that there is a largest probabilistic bisimulation on $\mathcal{M}$ and that this relation is an equivalence relation (see, e.g., [25, Section 7.6]). The maximum probabilistic bisimulation is called *probabilistic bisimilarity* and is denoted $\sim$. From now on, we mostly refer to probabilistic bisimilarity as simply bisimilarity.

We are interested in the problem of computing bisimilarity $\sim$ on $\mathcal{M}$. The decision version of the problem asks whether $s \sim t$ for two designated states $s, t \in S$.

The above formulation of the bisimilarity problem is convenient for our hardness proof, however variations, such as replacing state labels with labels on transitions, can easily be accommodated. It is also not difficult to reduce the problem above to the restricted case in which the set of labels has two elements.

For a state s, let $succ(s) = \{u : \pi_{s,u} > 0\}$. We say that a transition matrix π is *uniform* if for all $s \in S$ and u, $v \in succ(s)$, $\pi_{s,u} = \pi_{s,v}$. That is, the transition probability out of each state is a uniform distribution over its support.

Lemma 1 (Matching Lemma). *Assume that $|succ(s)| = |succ(t)|$ and π is uniform. Then $s \sim t$ if and only if $\ell(s) = \ell(t)$ and there exists a bijection $f : succ(s) \to succ(t)$ with $u \sim f(u)$ for each $u \in succ(s)$.*

Proof. Suppose that $s \sim t$. Since $\sim$ is a bisimulation, $\ell(s) = \ell(t)$ and for each $\sim$-equivalence class E,

$$\textstyle\sum_{x \in E} \pi_{s,x} = \sum_{x \in E} \pi_{t,x}. \tag{1}$$

Since $|succ(s)| = |succ(t)|$ and π is uniform, $|E \cap succ(s)| = |E \cap succ(t)|$ for each $\sim$-equivalence class E. Hence there exists a bijection $f : succ(s) \to succ(t)$ with $u \sim f(u)$ for all $u \in succ(s)$.

Conversely, assume that $\ell(s) = \ell(t)$ and suppose that f is a bijection as above. To conclude that $s \sim t$ we prove that the smallest equivalence relation containing $\sim \cup \{(s,t)\}$, which we denote by R, is a bisimulation.

Since $\sim$ is a bisimulation and $\ell(s) = \ell(t)$, R only relates states with the same label. Moreover, since every R-equivalence class is a union of $\sim$-equivalence classes, it suffices to show (1) for $\sim$-equivalences classes only.

Assume uRv. We distinguish three cases. First, let $u = s$ and $v = t$. Because of the existence of the bijection f, we have that $|E \cap succ(s)| = |E \cap succ(t)|$ for each $\sim$-equivalence class E. Because π is uniform, (1) holds for each $\sim$-equivalence class E. Second, let $u \sim s$ and $v \sim t$. Recall that $\sim$ is a bisimulation. Hence, for each $\sim$-equivalence class E,

$$\textstyle\sum_{x \in E} \pi_{u,x} = \sum_{x \in E} \pi_{s,x} = \sum_{x \in E} \pi_{t,x} = \sum_{x \in E} \pi_{v,x},$$

where we use $u \sim s$, the previous case, and $v \sim t$. The third and final case, $u \sim v$, follows immediately from the fact that $\sim$ is a bisimulation. □

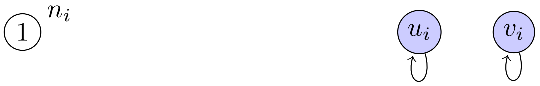
Theorem 2. *Deciding probabilistic bisimilarity is* ***P****-hard.*

Proof. We reduce from the Monotone Circuit Value problem which is **P**-hard [16, Theorem 6.2.2]. Recall that a monotone circuit is a finite directed acyclic graph in which nodes have in-degree either two or zero. Nodes with in-degree two are labelled $\wedge$ or $\vee$; nodes with in-degree zero, called input nodes, are labelled either true (1) or false (0). There is a distinguished output node with out-degree zero. The Monotone Circuit Value problem is to compute the output of a given monotone circuit.

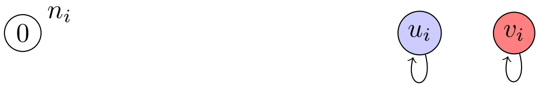
Given a circuit C, we define a Markov chain $\mathcal{M}(C)$ with a uniform transition matrix. For each node n_i of C and its incoming edges, we include a gadget consisting of states and their outgoing transitions in $\mathcal{M}(C)$. Note that the transitions of the Markov chain go in the opposite direction of the edges of the circuit. Each gadget contains states u_i and v_i. We will prove that $u_i \sim v_i$ if and only if n_i evaluates to true. We define the labelling function ℓ such that states have the same label if and only if they belong to the same gadget and the gadget does not represent an input node that is labelled false. In the diagrams below, states have the same label if and only if they have the same index and the same colour.

We describe $\mathcal{M}(C)$ by giving gadgets for each input node, *and*-gate and *or*-gate of C.

The gadget for input node labelled true is shown below.

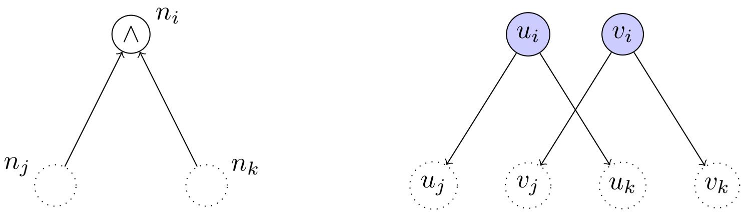


The gadget for input node labelled false is shown below.



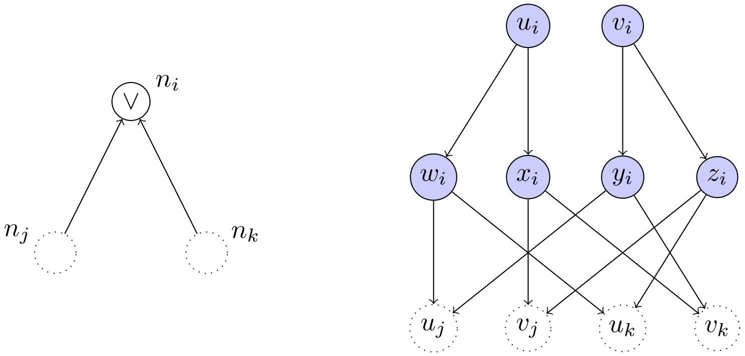
Note that u_i and v_i have the same label if and only if n_i is labelled true and therefore $u_i \sim v_i$ if and only if n_i is labelled true.

The gadget for an *and*-gate is shown below.



Note that u_j, v_j and u_k, v_k are states of the gadgets corresponding to the nodes n_j and n_k. The correctness of this gadget amounts to showing that $u_i \sim v_i$ if and only if both $u_j \sim v_j$ and $u_k \sim v_k$. This follows immediately from the Matching Lemma and the fact that the definition of ℓ precludes $u_j \sim v_k$ and $v_j \sim u_k$ in case n_j and n_k are different nodes. If n_j and n_k are one and the same node, the *and*-gate can be removed from the circuit.

The gadget for an *or*-gate is shown below.



The correctness of this gadget amounts to showing that $u_i \sim v_i$ if and only if $u_j \sim v_j$ or $u_k \sim v_k$.

$$\begin{aligned} u_i \sim v_i \text{ iff } & (w_i \sim y_i \wedge x_i \sim z_i) \vee (w_i \sim z_i \wedge x_i \sim y_i) \quad \text{[Matching Lemma]} \\ \text{ff } & u_j \sim v_j \vee u_k \sim v_k \quad \text{[Matching Lemma]} \end{aligned}$$

In the last step we use again the fact that the definition of ℓ precludes $u_j \sim v_k$ and $v_j \sim u_k$ in case n_j and n_k are different nodes. If n_j and n_k are one and the same node, the *or*-gate can be removed from the circuit.

This completes the description of the gadgets. The Markov chain $\mathcal{M}(C)$ is obtained by composing the gadgets for each node of C. The transduction of a circuit to a Markov chain is done gate by gate. To produce the output gadget corresponding to each circuit gate one only needs to store the indices of the gate and its two inputs, and the states of the output gadget. Thus the reduction can be done in deterministic logarithmic space. □

The proofs of **P**-hardness of ordinary bisimilarity for labelled transition systems by Balcázar, Gabarró and Sántha [4] and Sawa and Jančar [26] are also by reduction from MONOTONE CIRCUIT VALUE. However in the probabilistic case disjunction cannot be translated directly as in the non-deterministic case. Interestingly, a formally identical gadget to the above disjunction gadget appears in Toran's proof of **DET**-hardness of graph isomorphism [29]. However **DET** is a subclass of **P** and the graph isomorphism problem is not known to be **P**-hard.

3 The Bisimilarity Pseudometric

In this section we recall the definition of a bisimilarity pseudometric on labelled Markov chains. We first give a logical characterisation, due to Desharnais, Gupta, Jagadeesan and Panangaden [12], based on a real-valued semantics for Larsen and Skou's probabilistic modal logic [21]. This characterisation illustrates the sense in which states that are close in the pseudometric satisfy similar behavioural properties. In the next section we give a more abstract fixed-point characterisation of the pseudometric, which will be used in our algorithms.

The logic $\mathcal{L}$ is defined by the grammar

$$\varphi ::= \sigma \mid \varphi \vee \varphi \mid \neg\varphi \mid \varphi \ominus q \mid \Diamond\varphi \tag{2}$$

where $\sigma \in \Sigma$ and $q \in [0,1]$ is rational.

We consider a real-valued semantics of $\mathcal{L}$, which is parameterised by a *discount factor* $c \in (0,1]$. The smaller the value of c, the more the future is discounted, with $c = 1$ being the *undiscounted* case. Given a labelled Markov chain $\mathcal{M} = (S, \Sigma, \pi, \ell)$, the interpretation of a formula φ is a function $[\![\varphi]\!] : S \to [0,1]$ defined by the following clauses:

$$\begin{aligned}
[\![\sigma]\!](s) &= \begin{cases} 1 \text{ if } \ell(s) = \sigma \\ 0 \text{ otherwise} \end{cases} \\
[\![\varphi \vee \psi]\!](s) &= \max([\![\varphi]\!](s), [\![\psi]\!](s)) \\
[\![\neg\varphi]\!](s) &= 1 - [\![\varphi]\!](s) \\
[\![\varphi \ominus q]\!](s) &= \max([\![\varphi]\!](s) - q, 0) \\
[\![\Diamond\varphi]\!](s) &= c \cdot \sum_{t \in S} \pi_{s,t} \cdot [\![\varphi]\!](t)
\end{aligned}$$

A *pseudometric* is a relaxation of the notion of an ordinary metric in which different states can have distance zero. Formally a (1-bounded) pseudometric on a set S is a map $d : S \times S \to [0,1]$ such that for all $s, t, u \in S$, $d(s,s) = 0$, $d(s,t) = d(t,s)$ and $d(s,u) \leq d(s,t) + d(t,u)$.

Given a discount factor $c \in (0,1]$ the function $\mathsf{d}_c : S \times S \to [0,1]$ assigns a distance to every pair of states of a labelled Markov chain according to the following definition:

$$\mathsf{d}_c(s,t) = \sup_{\varphi \in \mathcal{L}} \left| [\![\varphi]\!](s) - [\![\varphi]\!](t) \right| . \tag{3}$$

It is straightforward that, with this definition, d_c is a pseudometric. The following theorem justifies our description of d_c as a bisimilarity pseudometric.

Theorem 3. [25, Section 8.2] $\mathsf{d}_c(s,t)=0$ *if and only if* $s \sim t$.

In [9], Chatterjee, de Alfaro, Majumdar and Raman enriched the logic $\mathcal{L}$ by the addition of fixed-point operators, yielding a *quantitative μ-calculus* $\mathcal{L}_\mu$ which can express reachability and ω-regular specifications. For example, the $\mathcal{L}_\mu$-formula $\mu x.(\sigma \vee \Diamond x)$ represents the probability to reach a σ-labelled state, while $\nu y.\mu x.((\sigma \wedge y) \vee \Diamond x)$ represents the probability to infinitely often visit a σ-labelled state. It is shown in [9] that $\mathsf{d}_c(s,t) = \sup_{\varphi \in \mathcal{L}_\mu} |[\![\varphi]\!](s) - [\![\varphi]\!](t)|$ for any pair of states $s,t \in S$; thus d_c can equivalently be defined in terms of the more powerful logic $\mathcal{L}_\mu$.

4 Matchings, Couplings and the Kantorovich Metric

In this section we give a fixed-point characterisation of the probabilistic bisimilarity pseudometric. Based on this we relate the pseudometric to the classical notion of *couplings* on Markov chains.

Say that a probability distribution ω on $S \times S$ is a *matching* of probability distributions μ, ν on S if

$$\begin{array}{ll} \sum_{v \in S} \omega(u,v) = \mu(u) & \text{for all } u \in S \\ \sum_{u \in S} \omega(u,v) = \nu(v) & \text{for all } v \in S\,. \end{array}$$

In other words, ω is a joint probability distribution whose marginals are μ and ν.

Suppose that (S,d) is a finite metric space. The *Kantorovich metric d_K* on the set of probability distributions on S is defined by

$$d_K(\mu,\nu) = \min_{\omega \in \Omega_{\mu,\nu}} \sum_{u,v \in S} d(u,v) \cdot \omega(u,v)\,,$$

where $\Omega_{\mu,\nu}$ is the set of matchings of μ and ν.

Informally we can characterise the bisimilarity pseudometric $\mathsf{d}_c(s,t)$ as the distance between the distributions $\pi_{s,-}$ and $\pi_{t,-}$ in the Kantorovich metric over (S, d_c). This characterisation is recursive, and accordingly we will show that d_c is a fixed point of a functional Δ_c based on the Kantorovich metric.

Define $\Delta_c : [0,1]^{S \times S} \to [0,1]^{S \times S}$ as follows. If $\ell(s) \neq \ell(t)$ then $\Delta_c(d)(s,t) = 1$ and if $\ell(s) = \ell(t)$ then

$$\Delta_c(d)(s,t) = c \cdot \min_{\omega \in \Omega_{s,t}} \sum_{u,v \in S} d(u,v) \cdot \omega(u,v)\,, \tag{4}$$

where $\Omega_{s,t}$ is the set of matchings of $\pi_{s,-}$ and $\pi_{t,-}$.

The set $[0,1]^{S \times S}$ is a complete lattice in the pointwise order. It is shown in [5, Proposition 38] that Δ_c is a monotone selfmap on $[0,1]^{S \times S}$ and thus, by Tarski's fixed point theorem, has a least fixed point. Since the least element of $[0,1]^{S \times S}$ is a pseudometric, Δ_c maps a pseudometric to a pseudometric, and the least upper-bound of a set of pseudometrics is a pseudometric, we can conclude that the least fixed point of Δ_c is a pseudometric as well. This turns out to be the pseudometric d_c.

Theorem 4. *[6, Theorem 4.6]* d_c *is the least fixed point of* Δ_c*.*

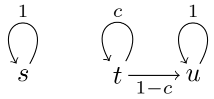
Remark 5. In the relational setting it is traditional to view bisimilarity as a greatest fixed point. Intuitively the situation is opposite in the pseudometric setting because the bottom element of $[0, 1]$ represents relatedness.

Theorem 6. *If* $c < 1$ *then* d_c *is the unique fixed point of* Δ_c*.*

Proof sketch. We can show that Δ_c is c-Lipschitz. From Banach's fixed point theorem we can conclude that the fixed point is unique. □

However, Δ_1 need not have a unique fixed point. For example, consider the labelled Markov chain with a single state. Then Δ_1 is the identity mapping.

Example 7. Consider the Markov chain below, where $\ell(s) = \ell(t) \neq \ell(u)$:



For $c < 1$, $\mathsf{d}_c(s, t) = \frac{c-c^2}{1-c^2}$. The pseudometric $\mathbf{0}$ assigns to every pair of states distance zero. For all $n \in \mathbb{N}$, $\Delta_c^n(\mathbf{0})(s, t) \leq \frac{c-c^{2n+1}}{1+c}$. This shows that the fixed point may not be reached by a finite number of iterations of Δ_c.

For each $s, t \in S$, let $\omega_{(s,t),(-,-)}$ be a matching of $\pi_{s,-}$ and $\pi_{t,-}$. Then the Markov chain $\mathcal{C} = (S \times S, \omega)$ is a coupling (see, e.g., [24, Chapter 11] for a discussion of couplings). Such a coupling can be seen as two copies of $\mathcal{M}$ running synchronously, although not necessarily independently. Couplings are typically used to give upper bounds on convergence to stationary distributions. Here we use them to a slightly different end. Given a coupling $\mathcal{C}$, as above, define the *discrepancy* of a state $(s, t) \in S \times S$, denoted $d_{\mathcal{C}}(s, t)$, to be the probability that a trajectory of $\mathcal{C}$ starting in state (s, t) reaches a state (u, v) with $\ell(u) \neq \ell(v)$.

Formally, given a coupling $\mathcal{C}$, we define $\Gamma_{\mathcal{C}} : [0, 1]^{S \times S} \to [0, 1]^{S \times S}$ as follows. If $\ell(s) \neq \ell(t)$ then $\Gamma_{\mathcal{C}}(d)(s, t) = 1$ and if $\ell(s) = \ell(t)$ then

$$\Gamma_{\mathcal{C}}(d)(s, t) = \sum_{u,v \in S} d(u, v) \cdot \omega_{(s,t),(u,v)}.$$

We leave it to the reader to check that $\Gamma_{\mathcal{C}}$ is a monotone selfmap on $[0, 1]^{S \times S}$. By Tarski's fixed point theorem, $\Gamma_{\mathcal{C}}$ has a least fixed point, which we denote by $d_{\mathcal{C}}$. As we will show next, $d_{\mathcal{C}}$ is closely related to our bisimilarity pseudometric d_1.

Theorem 8. $\mathsf{d}_1 = \min\{\, d_{\mathcal{C}} : \mathcal{C}$ *is a coupling*$\,\}$.

As a consequence of the above theorem, the bisimilarity pseudometric d_1 corresponds to the minimal coupling. Next, we will show that two states have discrepancy zero in some coupling if and only if they are bisimilar.

Proposition 9. *$d_{\mathcal{C}}(s,t) = 0$ for some coupling $\mathcal{C}$ if and only if $s \sim t$.*

Proof. From Theorem 8 we can conclude that $d_{\mathcal{C}}(s,t) = 0$ for some coupling $\mathcal{C}$ if and only if $\mathsf{d}_1(s,t) = 0$. By Theorem 3 this gives us the desired result. □

The following simple lemma shows that the discrepancy can be used to bound the variational distance between trace distributions. This can be seen as a quantitative version of the folklore that bisimilar states satisfy the same linear-time properties. In the lemma we use $\mathrm{Pr}_{\mathcal{M},s}(A)$ to denote the probability that a run of the labelled Markov chain $\mathcal{M}$ started in state s is in the set A. For a formal definition of $\mathrm{Pr}_{\mathcal{M},s}(A)$ and a definition of measurable subset of the set Σ^ω of infinite sequences over Σ, we refer the reader to, e.g., [3, Chapter 10].

Lemma 10 (Coupling Lemma). *Let $\mathcal{C}$ be a coupling of the labelled Markov chain $\mathcal{M} = (S, \Sigma, \pi, \ell)$. Then for any measurable set $A \subseteq \Sigma^\omega$ and $s, t \in S$,*

$$|\mathrm{Pr}_{\mathcal{M},s}(A) - \mathrm{Pr}_{\mathcal{M},t}(A)| \le d_{\mathcal{C}}(s,t)\,.$$

As a consequence of the Coupling Lemma and Theorem 8, our bisimilarity pseudometric is an upper bound for the variational distance between trace distributions.

Corollary 11. *For any measurable set $A \subseteq \Sigma^\omega$ and $s, t \in S$,*

$$|\mathrm{Pr}_{\mathcal{M},s}(A) - \mathrm{Pr}_{\mathcal{M},t}(A)| \le \mathsf{d}_1(s,t)\,.$$

Whereas the variational distance between trace distributions is **NP**-hard to compute, as shown by Lyngsø and Pedersen in [22], we will show that our bisimilarity pseudometric can be computed in polynomial time.

5 Algorithms for Bisimilarity Pseudometrics

5.1 The Discounted Case

Let $c < 1$ be a fixed rational discount factor. Given a labelled Markov chain $\mathcal{M}$, we show that d_c is rational and can be computed exactly in time polynomial in size($\mathcal{M}$) and size(c).[1]

In Theorem 4 we have characterised d_c as the least fixed point of Δ_c. While the stipulation that d_c be the least fixed point is essential in the undiscounted case, it is redundant in the discounted case. In the latter case, Δ_c has a unique fixed point (see Theorem 6). As a consequence, d_c is also the greatest fixed point of Δ_c for $c < 1$. Thus, by Tarski's fixed point theorem, we have

$$\mathsf{d}_c = \bigsqcup \{\, d \in \mathbb{R}^{S \times S} : d \le \Delta_c(d) \wedge 0 \le d \le 1 \,\}\,. \tag{5}$$

[1] We denote by size(X) the size of the representation of an object X. We represent rational numbers as quotients of integers written in binary. For example, the size of a rational number is the sum of the bit lengths of its numerator and denominator and the size of a matrix is the sum of the sizes of its entries.

This simple change in perspective is fruitful because the characterisation (5) directly yields a translation of the problem of computing d_c to the following linear program:

$$\begin{array}{lll} \text{maximise} & \sum_{s,t \in S} d_{s,t} & \\ \text{such that} & d_{s,t} \ \leq \ c \cdot \sum_{u,v \in S} d_{u,v} \cdot \omega(u,v) & \omega \in \Omega_{s,t},\ \ell(s) \neq \ell(t) \\ & d_{s,t} \ = \ 1 & \ell(s) = \ell(t) \\ & 0 \leq d_{s,t} \leq 1 & \end{array} \tag{6}$$

As we will see, the linear program (6) can be solved in polynomial time using the ellipsoid algorithm. We pursue this option in the undiscounted setting below. However, here we do not require such powerful techniques. Instead we just use the linear programming formulation to observe that the fixed point of Δ_c is rational and bounded in size by a polynomial in size($\mathcal{M}$) and size(c). We then approximate the fixed point by repeating the network simplex algorithm, obtaining the exact solution by rounding by means of the continued fraction algorithm.

Recall that the set of matchings $\Omega_{s,t}$ is a polytope in $\mathbb{R}^{S \times S}$ defined by the following constraints:

$$\textstyle\sum_{v \in S} \omega(u,v) = \pi_{s,u} \text{ and } \sum_{u \in S} \omega(u,v) = \pi_{t,v} \text{ and } \omega(u,v) \geq 0 \tag{7}$$

In general, $\Omega_{s,t}$ is infinite and therefore the set of constraints in (6) is infinite also. However, for each fixed d the linear function mapping a matching ω to $c \cdot \sum_{u,v} d(u,v) \cdot \omega(u,v)$ achieves its minimum on $\Omega_{s,t}$ at some vertex. Thus, writing $V(\Omega_{s,t})$ for the (finite) set of vertices of $\Omega_{s,t}$, we can replace $\Omega_{s,t}$ with $V(\Omega_{s,t})$ in (6), obtaining a linear program with the same feasible region. We denote the polytope defined by the set of constraints of this linear program by D. To prove that the distances are rational, we first observe the following.

Proposition 12. *Each $\omega \in V(\Omega_{s,t})$ is rational of size polynomial in* size($\mathcal{M}$).

Proof sketch. Since a vertex of $\Omega_{s,t}$ is by definition an intersection of hyperplanes given by the (in)equalities defining $\Omega_{s,t}$ and the coefficients of the (in)equalities are rationals bounded in size by size($\mathcal{M}$), we can conclude that each $\omega \in V(\Omega_{s,t})$ is rational of size polynomial in size($\mathcal{M}$). □

Proposition 13. d_c *is rational of size polynomial in* size($\mathcal{M}$) *and* size(c).

Proof sketch. Along a similar line of reasoning as used in the proof of Proposition 12, we can conclude that the vertices of polytope D are rational of size polynomial in size($\mathcal{M}$) and size(c).

Since the function mapping any d of the polytope D to $\sum_{s,t \in S} d_{s,t}$ is linear, it attains its maximum, d_c, at some vertex of D, which, as we have just shown, is rational of size polynomial in size($\mathcal{M}$) and size(c). □

Note that the proofs of Proposition 12 and 13 are also valid for $c = 1$ and, hence, d_1 is rational as well. Having established that d_c is rational, we now give a simple iterative algorithm to approximate d_c starting from the pseudometric $\mathbf{0}$.

Proposition 14. *For all* $n \in \mathbb{N}$, $\Delta_c^n(\mathbf{0})$ *is rational of size polynomial in* $\text{size}(\mathcal{M})$ *and* $\text{size}(c)$ *and can be computed in time polynomial in* $\text{size}(\mathcal{M})$ *and* $\text{size}(c)$.

Proof sketch. We prove this property by induction on n. Obviously, the property holds for $n = 0$. Let $n > 0$. Obviously, the property holds when $\ell(s) \neq \ell(t)$. Otherwise, $\Delta_c^n(\mathbf{0})(s,t) = c \cdot \min_{\omega \in \Omega_{s,t}} \sum_{u,v \in S} \Delta_c^{n-1}(\mathbf{0})(u,v) \cdot \omega(u,v)$. The above minimum is attained at a vertex of $\Omega_{s,t}$. As we have seen in Proposition 12, these vertices are rationals of size polynomial in $\text{size}(\mathcal{M})$. Furthermore, by induction, $\Delta_c^{n-1}(\mathbf{0})$ is rational of size polynomial in $\text{size}(\mathcal{M})$ and $\text{size}(c)$. Hence, $\Delta_c^n(\mathbf{0})(s,t)$ is a rational of size polynomial in $\text{size}(\mathcal{M})$ and $\text{size}(c)$. Computing $\Delta_c(d)(s,t)$ is a minimum-cost flow problem for which there are versions of the network simplex algorithm that are strongly polynomial time [1]. □

To get ϵ-close to d_c, we need to iterate $\lceil \log_c(\epsilon) \rceil$ times.

Proposition 15. *For all* $\epsilon > 0$, $\|\Delta_c^{\lceil \log_c(\epsilon) \rceil}(\mathbf{0}) - \mathsf{d}_c\| \leq \epsilon$.

From Proposition 14 and 15 we can conclude that we can approximate d_c in time polynomial in $\text{size}(\mathcal{M})$, $\text{size}(c)$ and $\log_2(\frac{1}{\epsilon})$. Once we have iterated close enough to d_c, we can use the continued fraction algorithm (see, e.g. [17, Section 5.1]) to obtain d_c.

Theorem 16. *The pseudometric* d_c *can be computed in time polynomial in* $\text{size}(\mathcal{M})$ *and* $\text{size}(c)$.

Proof sketch. This follows now immediately from the observation made by Etessami and Yannakakis [13, page 2540] that for problems whose solutions are rational, of size polynomial in the input size, if we can solve the approximation problem in polynomial time, then we can also solve the exact computation problem in polynomial time by using the continued fraction algorithm. □

5.2 The Undiscounted Case

Throughout this section we refer to the undiscounted bisimilarity pseudometric as d, rather than d_1 and likewise use Δ instead of Δ_1.

In the previous section we gave a reduction of the problem of computing d_c to linear programming by characterising d_c as the greatest fixed point of Δ_c for $c < 1$. However, recall from Section 4 that d is not in general the greatest fixed point of Δ. Nevertheless we can recover a greatest-fixed-point characterisation of d by separately handling the set of bisimilar states, i.e. the states at distance zero. This will allow us to use linear programming to compute d.

As a first step we define $\Delta' : [0,1]^{S \times S} \to [0,1]^{S \times S}$ by

$$\Delta'(d)(s,t) = \begin{cases} \Delta(d)(s,t) & \text{if } s \not\sim t \\ 0 & \text{if } s \sim t. \end{cases}$$

Proposition 17. Δ' *has a unique fixed point.*

Proof sketch. Since Δ is monotone, we can easily deduce that Δ' is monotone as well. According to Tarski's fixed point theorem, Δ' has a least and a greatest fixed point. Hence, it is sufficient to prove that if $d \leq d'$ are both fixed points of Δ' then $d = d'$.

To this end, let $m = \max\{\, d'(s,t) - d(s,t) : s, t \in S \,\}$ and let M be the set of pairs $\{\, (s,t) \in S \times S : d'(s,t) - d(s,t) = m \,\}$ which maximise the discrepancy between d' and d. We will show that $m = 0$, which implies that $d = d'$. We distinguish two cases.

Assume that $(s,t) \in M$ such that $\ell(s) \neq \ell(t)$. Then

$$d'(s,t) - d(s,t) = \Delta'(d')(s,t) - \Delta'(d)(s,t) = 1 - 1 = 0$$

and, hence, $m = 0$.

Otherwise, for all $(s,t) \in M$ we have that $\ell(s) = \ell(t)$. In this case, we claim that $M \subseteq \sim$. From this claim it follows that for all $(s,t) \in M$,

$$d'(s,t) - d(s,t) = \Delta'(d')(s,t) - \Delta'(d)(s,t) = 0 - 0 = 0$$

and, hence, $m = 0$. It just remains to prove the claim.

By Proposition 9 it suffices to define a coupling $\mathcal{C}$ such that $d_{\mathcal{C}}(s,t) = 0$ for all $(s,t) \in M$. To define $\mathcal{C}$ we must specify a matching $\omega \in \Omega_{s,t}$ for each pair of states $(s,t) \in S \times S$. For $(s,t) \notin M$ any matching will do. For $(s,t) \in M$ we show that we can choose a matching $\omega \in \Omega_{s,t}$ whose support is contained in M. Then in the coupling $\mathcal{C}$ no pair in $(s,t) \in M$ can reach a pair (u,v) with $\ell(u) \neq \ell(v)$, that is, $d_{\mathcal{C}}(s,t) = 0$.

Let $(s,t) \in M$. Suppose $\Delta'(d)(s,t) = \sum_{u,v \in S} d(u,v) \cdot \omega(u,v)$, where $\omega \in \Omega_{s,t}$. Then

$$\begin{aligned}
m &= d'(s,t) - d(s,t) \\
&= \Delta'(d')(s,t) - \Delta'(d)(s,t) \\
&= \left(\min_{\omega' \in \Omega_{s,t}} \sum_{u,v \in S} d'(u,v) \cdot \omega'(u,v) \right) - \sum_{u,v \in S} d(u,v) \cdot \omega(u,v) \\
&\leq \sum_{u,v \in S} d'(u,v) \cdot \omega(u,v) - \sum_{u,v \in S} d(u,v) \cdot \omega(u,v) \\
&= \sum_{u,v \in S} (d'(u,v) - d(u,v)) \cdot \omega(u,v).
\end{aligned}$$

Since $d'(u,v) - d(u,v) \leq m$ and $\sum_{u,v \in S} \omega(u,v) = 1$, we can conclude from $\sum_{u,v \in S} (d'(u,v) - d(u,v)) \cdot \omega(u,v) \geq m$ that $d'(u,v) - d(u,v) = m$ whenever $\omega(u,v) > 0$. □

Corollary 18. d *is the unique fixed point of* Δ'.

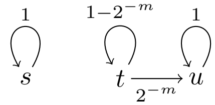
Proof. It is enough to prove that d is a fixed point of Δ'. On the one hand, suppose that $s \sim t$. Then $\mathsf{d}(s,t) = 0 = \Delta'(d)(s,t)$ by Theorem 3. On the other hand, suppose that $s \not\sim t$. Then $\mathsf{d}(s,t) = \Delta(\mathsf{d})(s,t) = \Delta'(\mathsf{d})(s,t)$. □

Corollary 18 implies that d is the greatest fixed point of Δ'. Thus, following the development in Section 5.1, we can compute d as the solution to the following linear program:

$$\begin{array}{lrcll} \text{maximise} & \sum_{s,t\in S} d_{s,t} & & & \\ \text{such that} & d_{s,t} & = & 0 & s \sim t \\ & d_{s,t} & = & 1 & \ell(s) \neq \ell(t) \\ & d_{s,t} & \leq & \sum_{u,v\in S} d_{u,v} \cdot \omega(u,v) & \omega \in V(\Omega_{s,t}), s \not\sim t, \ell(s) = \ell(t) \end{array} \tag{8}$$

Unfortunately we cannot solve (8) using the iterative method adopted in the discounted case. The reason is that it may require exponentially many iterations of Δ' to achieve a sufficiently close approximation to d.

Example 19. Consider the Markov chain below, where $\ell(s) = \ell(t) \neq \ell(u)$:



Then $\mathsf{d}(s,t) = 1$ and $(\Delta')^n(\mathbf{0})(s,t) \leq n \cdot 2^{-m}$ for all $n \in \mathbb{N}$. This shows that it may require exponentially many iterations in size($\mathcal{M}$) to approximate the fixed point of Δ'.

Instead we use the ellipsoid algorithm (see, e.g. [27, Chapter 14]) to solve the linear program (8). According to Proposition 12 (which also holds for $c = 1$), the coefficients of the constraints of the linear program (8) are rational of size polynomial in size($\mathcal{M}$). By, e.g. [27, Corollary 14.1a], to conclude that the linear program (8) can be solved in time polynomial in size($\mathcal{M}$), it suffices to show that there exists a polynomial time separation algorithm. In our setting, given a $d \in \mathbb{R}^{S\times S}$ rational of size polynomial in size($\mathcal{M}$), the separation algorithm has to decide whether d satisfies the constraints of (8) or not, and, in the latter case, find in time polynomial in size($\mathcal{M}$) a separating hyperplane, i.e., an $\alpha \in \mathbb{Q}^{S\times S}$ such that

$$\textstyle\sum_{u,v\in S} d(u,v) \cdot \alpha(u,v) < \sum_{u,v\in S} d'(u,v) \cdot \alpha(u,v) \tag{9}$$

for all $d' \in \mathbb{R}^{S\times S}$ that satisfy the constraints of (8).

Let $d \in \mathbb{R}^{S\times S}$ be rational of size polynomial in size($\mathcal{M}$). For each pair of states $s, t \in S$, we consider the following linear program:

$$\begin{array}{ll} \text{minimise} & \sum_{u,v\in S} d(u,v) \cdot \omega_{u,v} \\ \text{such that} & \sum_{v\in S} \omega_{u,v} = \pi_{s,u} \text{ and } \sum_{u\in S} \omega_{u,v} = \pi_{t,v} \text{ and } \omega_{u,v} \geq 0 \end{array} \tag{10}$$

This linear program is a minimum-cost flow problem for which there are versions of the network simplex algorithm that can compute an $(\omega_{u,v})_{u,v\in S}$, which satisfies the constraints of (10) and minimizes the objective function, and that are strongly polynomial time [1].

Note that d satisfies the constraints of (10) if and only if $d(s,t)$ is smaller than or equal to the optimal value of (10) for each pair of states $s, t \in S$. Otherwise, there exists a pair of states $s, t \in S$ such that $d(s,t)$ is greater than the optimal value of (10). Let $\omega \in V(\Omega_{s,t})$ be a vertex that realizes the optimal value of (10). As we have seen in Proposition 12, ω is rational of size polynomial in size($\mathcal{M}$).

It remains to define an α that satisfies (9). We define α in terms of ω as follows:

$$\alpha(u,v) = \begin{cases} \omega(u,v) - 1 & \text{if } (u,v) = (s,t) \\ \omega(u,v) & \text{otherwise.} \end{cases}$$

Proposition 20. *Assume that d does not satisfy the constraints of (10). Then for all $d' \in \mathbb{R}^{S\times S}$ that satisfy the constraints of (10), we have (9).*

Proof. Since d does not satisfy the constraints of (10), there exists a pair of states $s, t \in S$ such that $d(s,t) > \sum_{u,v\in S} d(u,v) \cdot \omega(u,v)$. Hence,

$$\textstyle\sum_{u,v\in S} d(u,v) \cdot \alpha(u,v) < 0. \tag{11}$$

Assume that $d' \in \mathbb{R}^{S\times S}$ satisfies the constraints of (10). Then we have that $d'(s,t) \leq \sum_{u,v\in S} d'(u,v) \cdot \omega(u,v)$. Hence,

$$\textstyle\sum_{u,v\in S} d'(u,v) \cdot \alpha(u,v) \geq 0. \tag{12}$$

From (11) and (12) we can immediately conclude (9). □

6 Conclusion

The linear program (6) shows *inter alia* that the problem of computing probabilistic bisimilarity can naturally be reduced to linear programming. It would be interesting to relate the resulting procedure for computing probabilistic bisimilarity to the classical partition refinement algorithm.

The problem of computing probabilistic bisimilarity bears some similarity to the graph isomorphism problem. While the latter is not known to be in **P**, for certain restricted graph classes, such as graphs of bounded degree or with bounded colour classes, it is in **DET** (a subclass of **P**). By contrast, deciding probabilistic bisimilarity is **P**-hard already for labelled Markov chains with branching degree at most two, in which at most four states share the same label.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network flows – theory, algorithms and applications (1993)
2. Baier, C.: Polynomial Time Algorithms for Testing Probabilistic Bisimulation and Simulation. In: Alur, R., Henzinger, T.A. (eds.) CAV 1996. LNCS, vol. 1102, pp. 50–61. Springer, Heidelberg (1996)
3. Baier, C., Katoen, J.-P.: Principles of Model Checking (2008)
4. Balcázar, J.L., Gabarró, J., Sántha, M.: Deciding bisimilarity is P-complete. FAC 4(6A), 638–648 (1992)
5. van Breugel, F., Hermida, C., Makkai, M., Worrell, J.: Recursively defined metric spaces without contraction. TCS 380(1-2), 171–197 (2007)

6. van Breugel, F., Sharma, B., Worrell, J.: Approximating a behavioural pseudometric without discount for probabilistic systems. LMCS 4(2:2) (2008)
7. van Breugel, F., Worrell, J.: Approximating and computing behavioural distances in probabilistic transition systems. TCS 360(1-3), 373–385 (2006)
8. Cai, X., Gu, Y.: Measuring Anonymity. In: Bao, F., Li, H., Wang, G. (eds.) ISPEC 2009. LNCS, vol. 5451, pp. 183–194. Springer, Heidelberg (2009)
9. Chatterjee, K., de Alfaro, L., Majumdar, R., Raman, V.: Algorithms for game metrics. In: FSTTCS, pp. 107–118 (2008)
10. Comanici, G., Precup, D.: Basis function discovery using spectral clustering and bisimulation metrics. In: AAAI, pp. 325–330 (2011)
11. Derisavi, S., Hermanns, H., Sanders, W.H.: Optimal state-space lumping in Markov chains. IPL 87(6), 309–315 (2003)
12. Desharnais, J., Gupta, V., Jagadeesan, R., Panangaden, P.: Metrics for labelled Markov processes. TCS 318(3), 323–354 (2004)
13. Etessami, K., Yannakakis, M.: On the complexity of Nash equilibria and other fixed points. SIAM Journal on Computing 39(6), 2531–2597 (2010)
14. Fu, H.: The complexity of deciding a behavioural pseudometric on probabilistic automata. Technical Report AIB-2011-26, RWTH Aachen (2011)
15. Giacalone, A., Jou, C.-C., Smolka, S.A.: Algebraic reasoning for probabilistic concurrent systems. In: PROCOMET, pp. 443–458 (1990)
16. Greenlaw, R., James Hoover, H., Ruzzo, W.L.: Limits to Parallel Computation: P-Completeness Theory (1995)
17. Grötschel, M., Lovász, L., Schrijver, A.: Geometric Algorithms and Combinatorial Optimization (1988)
18. Hillston, J.: A Compositional Approach to Performance Modelling (1996)
19. Katoen, J.-P., Kemna, T., Zapreev, I.S., Jansen, D.N.: Bisimulation Minimisation Mostly Speeds Up Probabilistic Model Checking. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 87–101. Springer, Heidelberg (2007)
20. Kemeny, J.G., Laurie Snell, J.: Finite Markov Chains (1960)
21. Larsen, K.G., Skou, A.: Bisimulation through probabilistic testing. I&C 94(1), 1–28 (1991)
22. Lyngsø, R.B., Pedersen, C.N.S.: The consensus string problem and the complexity of comparing hidden Markov models. JCSS 65(3), 545–569 (2002)
23. Milner, R.: Communication and Concurrency (1989)
24. Mitzenmacher, M., Upfal, E.: Probability and Computing (2005)
25. Panangaden, P.: Labelled Markov Processes (2009)
26. Sawa, Z., Jančar, P.: Behavioural equivalences on finite-state systems are PTIME-hard. Computers and Artificial Intelligence 24(5), 513–528 (2005)
27. Schrijver, A.: Theory of Linear and Integer Programming (1986)
28. Thorsley, D., Klavins, E.: Approximating stochastic biochemical processes with Wasserstein pseudometrics. IET Systems Biology 4(3), 193–211 (2010)
29. Torán, J.: On the hardness of graph isomorphism. SIAM Journal on Computing 33(5), 1093–1108 (2004)
30. Tzeng, W.-G.: A polynomial-time algorithm for the equivalence of probabilistic automata. SIAM Journal on Computing 21(2), 216–227 (1992)
31. Tzeng, W.-G.: On path equivalence of nondeterministic finite automata. IPL 58(1), 43–46 (1996)
32. Valmari, A., Franceschinis, G.: Simple $O(m \log n)$ Time Markov Chain Lumping. In: Esparza, J., Majumdar, R. (eds.) TACAS 2010. LNCS, vol. 6015, pp. 38–52. Springer, Heidelberg (2010)