Robust Probabilistic Bisimilarity for Labelled Markov Chains

No Author Given

No Institute Given

Abstract. Despite its prevalence, probabilistic bisimilarity suffers from a lack of robustness under minuscule perturbations of the transition probabilities. This can lead to discontinuities in the probabilistic bisimilarity distance function, undermining its reliability in practical applications where transition probabilities are often approximations derived from experimental data. Motivated by this limitation, we introduce the notion of robust probabilistic bisimilarity for labelled Markov chains, which ensures the continuity of the probabilistic bisimilarity distance function. We also propose an efficient algorithm for computing robust probabilistic bisimilarity and show that it performs well in practice, as evidenced by our experimental results.

Keywords: (probabilistic) model checking · labelled Markov chain · probabilistic bisimilarity · behavioural pseudometric.

1 Introduction

In the analysis and verification of probabilistic systems, one of the foundational concepts is identifying and merging system states that are behaviourally indistinguishable. Kemeny and Snell [28] introduced the notion of lumpability for Markov chains and it was adapted to the setting of labelled Markov chains by Larsen and Skou [31], known as probabilistic bisimulation. State-of-the-art probabilistic verification tools [30,22] implement a variety of methods for minimizing the state space of the system by collapsing probabilistically bisimilar states. This can significantly improve verification efficiency in some cases [27].

However, due to the sensitivity of behavioural equivalences to small changes in the transition probabilities, Giacalone et al. [20] proposed using behavioural pseudometrics to capture the behavioural similarity of states in a probabilistic system. Instead of classifying states as either equivalent or inequivalent, the pseudometric maps each pair of states to a real value in the unit interval, thus also quantifying the behavioral difference between non-equivalent states. Behavioural pseudometrics have been studied in the context of systems biology [42], games [7], planning [10] and security [6], among others.

In probabilistic verification, the most widely studied example of such a behavioural pseudometric is the *probabilistic bisimilarity distance*. It generalizes probabilistic bisimilarity quantitatively; in particular, the distance between two states is zero if and only if they are probabilistically bisimilar. The probabilistic bisimilarity distance was introduced by Desharnais et al. [14], based on a real-valued semantics for Larsen and Skou's probabilistic modal logic [31]. A formula φ of this logic maps any state s to a number $[\![\varphi]\!](s) \in [0,1]$. The probabilistic bisimilarity distance between two states s,t can be characterized as $\delta(s,t) = \sup_\varphi |[\![\varphi]\!](s) - [\![\varphi]\!](t)| \in [0,1]$, where φ ranges over all formulas. The lower the distance between two states, the more similar their behaviour. As shown by Van Breugel and Worrell [5], the probabilistic bisimilarity distance can also be characterized as a fixed point of a function (we use this definition in this paper).

However, as pointed out by Jaeger et al. [24], probabilistic bisimilarity distances are sometimes not continuous, leading to unexpected and abrupt changes in behaviour between two states when the transition probabilities are perturbed. Since the probabilities of the labelled Markov chain are usually obtained experimentally and, therefore, are often an approximation [41,16,33,36,39], the lack of robustness of probabilistic bisimilarity is a serious drawback. This inconsistency undermines the reliability of probabilistic bisimilarity as a measure of system equivalence and can be particularly problematic when used in practical applications where approximate models are prevalent.

Example 1. Consider Figure 1a on page 5. When $\varepsilon = 0$, states h_0 and h_1 are probabilistically bisimilar; i.e., their distance $\delta_0(h_0,h_1)$ equals 0 (the subscript of δ indicates ε). For $\varepsilon > 0$ we have $\delta_\varepsilon(h_0,h_1) > 0$; i.e., h_0 and h_1 are no longer bisimilar. However, when ε is small then $\delta_\varepsilon(h_0,h_1)$ is small. In fact, one can show that $\delta_\varepsilon(h_0,h_1) \le 2\varepsilon$, which implies that $\lim_{\varepsilon\to 0} \delta_\varepsilon(h_0,h_1) = 0$. This means that in this example, the distance is continuous in ε. One may say that states h_0, h_1 are not only probabilistically bisimilar, but also robustly so, in that they remain "almost" bisimilar when the transition probabilities are perturbed. Intuitively, states h_0 and h_1 behave similarly even for small positive ε: both states carry a blue label and perform a geometrically distributed number of self-loops (about two in expectation) before transitioning to state t.

Example 2. Consider Figure 1b on page 5. When $\varepsilon = 0$, states h_2 and h_3 are probabilistically bisimilar; i.e., their distance $\delta_0(h_2,h_3)$ equals 0. But for any $\varepsilon{>}0$ we have $\delta_\varepsilon(h_2,h_3) = 1$; i.e., h_2 and h_3 behave "maximally" differently in terms of the probabilistic bisimilarity distance. We have $\lim_{\varepsilon\to 0} \delta_\varepsilon(h_2,h_3) = 1$; so, in this example, the distance is discontinuous in ε. One may say that although states h_2, h_3 are probabilistically bisimilar, they are not robustly so, because upon perturbing the transition probabilities the behaviour of h_3 changes completely. For any positive ε, state h_2 remains in a self-loop forever, whereas h_3 eventually reaches the (red-labelled) state t_3 with probability 1.

In this paper, we address this issue by introducing the notion of *robust probabilistic bisimilarity* for labelled Markov chains. Robust probabilistic bisimilarity is a particular probabilistic bisimulation, implying that robust probabilistic bisimilarity is a subset of probabilistic bisimilarity. Crucially, we show that our definition ensures the continuity of the probabilistic bisimilarity distance function. This means that for any two states that are robustly probabilistically bisimilar, their probabilistic bisimilarity distance remains small even after small perturbations of any transition probabilities.

Secondly, we develop a polynomial-time algorithm for computing robust probabilistic bisimilarity. It is suitable for large-scale verification tasks, opening the door to checking probabilistic models from the literature for (lack of) robustness of their probabilistic bisimilarity relation. Thus, one can identify pairs of states that may be dangerous to merge if the transition probabilities are not known precisely. We present experimental results on the applicability and efficiency of an implementation of our algorithm on models from the Quantitative Verification Benchmark Set (QVBS) [21] and the examples included in the Java PathFinder extension jpf-probabilistic [18].

The rest of the paper is structured as follows. Section 2 introduces the model of interest, namely a labelled Markov chain, and probabilistic bisimilarity. In Section 3, we formally define probabilistic bisimilarity distances and further examine how the bisimilarity distance changes when the transition function is varied. Section 4 describes robust probabilistic bisimilarity and demonstrates that it ensures the continuity of the bisimilarity distance function. In Section 5, we present a polynomial-time algorithm for computing robust probabilistic bisimilarity. Section 6 reports experimental results on the algorithm's implementation. Finally, Section 7 concludes the paper and discusses directions for future research. Omitted proofs can be found in the appendix.

2 Labelled Markov Chains and Probabilistic Bisimilarity

In this section, we present some fundamental concepts that underpin this paper.

Let X be a nonempty finite set. A function $\mu : X \to [0, 1]$ is a *subprobability distribution* on X if $\sum_{x \in X} \mu(x) \leq 1$. We denote the set of subprobability distributions on X by $\mathcal{S}(X)$. For $\mu \in \mathcal{S}(X)$ and $A \subseteq X$, we often write $\mu(A)$ instead of $\sum_{x \in A} \mu(x)$. For a distribution $\mu \in \mathcal{S}(X)$ we define the *support* of μ by $\text{support}(\mu) = \{ x \in X \mid \mu(x) > 0 \}$. A subprobability distribution μ on X is a *probability distribution* if $\mu(X) = 1$. We denote the set of probability distributions on X by $\mathcal{D}(X)$.

A *Markov chain* is a pair $\langle S, \tau \rangle$ consisting of a finite set S of states and a transition probability function $\tau : S \to \mathcal{D}(S)$. A *labelled Markov chain* is a tuple $\langle S, L, \tau, \ell \rangle$ where $\langle S, \tau \rangle$ is a Markov chain, L is a finite set of labels and $\ell : S \to L$ is a labelling function. A *path* in a Markov chain $\langle S, \tau \rangle$ is a sequence of states s_0, s_1, $s_2 \ldots$ such that $s_i \in S$ and $\tau(s_i)(s_{i+1}) > 0$ for all $i \geq 0$.

For the remainder, we fix a labelled Markov chain $\langle S, L, \tau, \ell \rangle$, and we will study perturbations of the transition probability function τ.

For all μ, $\nu \in \mathcal{D}(X)$, the set $\Omega(\mu, \nu)$ of *couplings* of μ and ν is defined by

$$\Omega(\mu, \nu) = \{ \omega \in \mathcal{D}(X \times X) \mid \forall x \in X : \omega(x, X) = \mu(x) \wedge \omega(X, x) = \nu(x) \}.$$

We write $\omega(x, X)$ for $\sum_{y \in X} \omega(x, y)$.

Definition 1. *An equivalence relation $R \subseteq S \times S$ is a* probabilistic bisimulation *(or just bisimulation) if for all $(s,t) \in R$, $\ell(s) = \ell(t)$ and there exists $\omega \in \Omega(\tau(s), \tau(t))$ such that* support$(\omega) \subseteq R$. *States s and t are* bisimilar*, denoted $s \sim t$, if $(s,t) \in R$ for some bisimulation R.*

If $|\ell(S)| = 1$ then $\sim = S \times S$. In the remainder, we assume that the labelled Markov chain contains states with different labels, that is, $|\ell(S)| \geq 2$. Hence, we also have that $|S| \geq 2$.

Definition 1 [25, Definition 4.3] differs from the standard definition [31, Definition 6.3] which defines a bisimulation as an equivalence relation $R \subseteq S \times S$ such that for all $(s,t) \in R$, $\ell(s) = \ell(t)$ and for all R-equivalence classes C, $\tau(s)(C) = \tau(t)(C)$, where $\tau(s)(C) = \sum_{t \in C} \tau(s)(t)$. Nevertheless, an equivalence relation R is a bisimulation by Definition 1 if and only if it is a bisimulation as per the standard definition (see [25, Theorem 4.6]).

3 Probabilistic Bisimilarity Distances

Definition 2. *The* probabilistic bisimilarity distance *(or just bisimilarity distance), $\delta_\tau : S \times S \to [0,1]$, is the least fixed point of the function $\Delta_\tau : (S \to \mathcal{D}(S)) \to (S \times S \to [0,1]) \to (S \times S \to [0,1])$ defined by*

$$\Delta_\tau(d)(s,t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \inf\limits_{\omega \in \Omega(\tau(s),\tau(t))} \sum\limits_{u,v \in S} \omega(u,v)\, d(u,v) & \text{otherwise.} \end{cases}$$

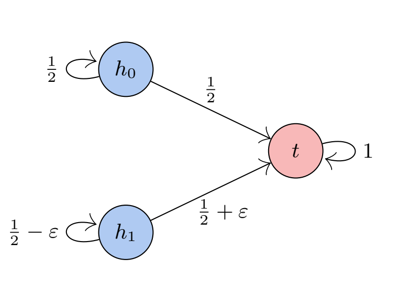
Intuitively, the smaller the distance between two states, the more similar they behave.

Theorem 1 ([15, Theorem 4.10]). *For all s, $t \in S$, $s \sim t$ if and only if $\delta_\tau(s,t) = 0$.*

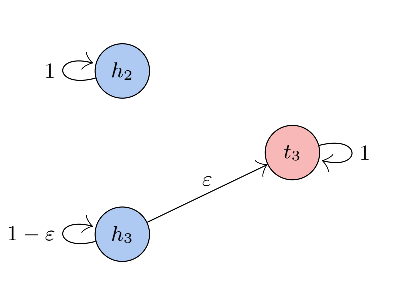
Quantitative μ-calculus [29,1,32] is an expressive modal logic that uses fixed-point operators to define properties of transition systems. It supports the concise representation of a wide range of properties, including reachability, safety, and the probability of satisfying a general ω-regular specification. We use the syntax described in [7], except that we use the operator next instead of pre_1 and pre_2. Let $q\mu$ denote the set of quantitative μ-calculus formulae, then a formula $\varphi \in q\mu$ maps states to a numerical value within $[0,1]$, that is, $[\![\varphi]\!] : S \to [0,1]$. The bisimilarity distances can be characterized in terms of the quantitative μ-calculus [7, Equation 2.3] as $\delta_\tau(s,t) = \sup_{\varphi \in q\mu} |[\![\varphi]\!](s) - [\![\varphi]\!](t)|$.

3.1 Examples

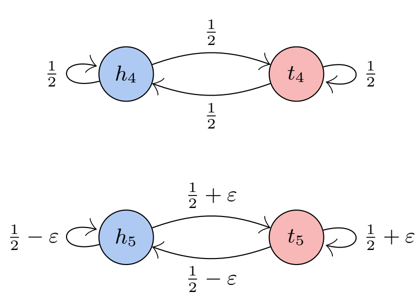
We now investigate how the bisimilarity distance changes when the transition function varies. In the following, let $\varepsilon \in [0, \frac{1}{2}]$. Define τ_ε as shown in Figure 1. For example, $\tau_{1/6}(h_5)(t_5) = \frac{2}{3}$. Then $\tau_\cdot : [0, \frac{1}{2}] \to (S \to \mathcal{D}(S))$ and $\delta_{\tau_\cdot} : [0, \frac{1}{2}] \to (S \times S \to [0,1])$.



(a) Repeated tosses of a fair coin (top) and a biased coin (bottom) until each lands on tails.



(b) Single toss of a rigged coin (top) and repeated tosses of an extremely biased coin until it lands on tails (bottom).



(c) Repeated tosses of a fair coin (top) and a biased coin (bottom).

Fig. 1: Various examples featuring fair and biased coins. States labeled with *heads* are shown in blue, while states labeled with *tails* are shown in red.

Example 3. Consider Figure 1a. As ε increases, h_1 becomes more biased and the distance between h_0 and h_1 increases proportionally. One can show that $\delta_{\tau_\varepsilon}(h_0, h_1) = \frac{\varepsilon}{0.5+\varepsilon} \leq 2\varepsilon$. Note that if ε is small then the distance is also small and $\lim_{\varepsilon \to 0} \delta_{\tau_\varepsilon}(h_0, h_1) = 0$.

The formula $\varphi = \mu V. \operatorname{next}(\mathit{tails} \vee V) \ominus 0.5$ distinguishes the states h_0 and h_1 the most, that is $\delta_{\tau_\varepsilon}(h_0, h_1) = \frac{\varepsilon}{0.5+\varepsilon} = |[\![\varphi]\!](h_0) - [\![\varphi]\!](h_1)|$. The quantifier μ denotes the least fixed-point of the recursive formula involving the variable V. Intuitively, a state satisfies V if the next state is *tails* or satisfies V with probability greater than a half. More precisely, considering state h_1, $[\![\varphi]\!](h_1)$ is the expected value of $\max([\![\varphi]\!](s), [\![\mathit{tails}]\!](s)) - \frac{1}{2}$, where s denotes the random successor state of h_1. Then, $[\![\varphi]\!](h_1)$ evaluates to $\sum_{n=0}^{\infty} \varepsilon(\frac{1}{2} - \varepsilon)^n = \frac{\varepsilon}{0.5+\varepsilon}$. Each summand in the series represents the probability of reaching state t in n steps, starting from state h_1, with 0.5 subtracted at each step. On the other hand, $[\![\varphi]\!](h_0) = 0$.

Example 4. In Figure 1b, when $\varepsilon = 0$, the states h_2 and h_3 are bisimilar with $\delta_{\tau_0}(h_2, h_3) = 0$. However, if $\varepsilon > 0$, then $\delta_{\tau_\varepsilon}(h_2, h_3) = 1$. This difference is evident when considering the probability of eventually reaching a state labelled with *tails* when starting in h_2 compared to h_3. In the first Markov chain, $P(\Diamond \mathit{tails}) = 0$, while in the second Markov chain, $P(\Diamond \mathit{tails}) = 1$. This property can be expressed as the quantitative μ-calculus formula $\mu V. \operatorname{next}(\mathit{tails} \vee V)$. This example was also presented in [24].

Example 5. The first Markov chain in Figure 1c represents fair coin flips, while the second Markov chain represents potentially biased coin flips. When $\varepsilon = 0$, the states h_4 and h_5 are bisimilar with $\delta_{\tau_0}(h_4, h_5) = 0$. However, if $\varepsilon > 0$, one can show that $\delta_{\tau_\varepsilon}(h_4, h_5) = 1$. Intuitively, this is because small differences in probabilities can compound and lead to qualitative differences in the long-run behaviour.

Let us illustrate this. Assume that a point is awarded each time the coin lands on *tails* and a point is deducted each time it lands on *heads*. Let us examine the limit behaviour of the Markov chains. Observe that the Markov chains behave like a random walk on the integer number line, $\mathbb{Z}$, starting at 0. At each step, the first Markov chain goes up by one with probability $\frac{1}{2}$ and down by one with probability $\frac{1}{2}$. On the other hand, at each step, the second Markov chain goes up by one with probability $\frac{1}{2} + \varepsilon$ and down by one with probability $\frac{1}{2} - \varepsilon$. Let $Y_1, Y_2, Y_3, \ldots$ be the sequence of independent random variables, where Y_i denotes the i^{th} step taken by the random walk, with $Y_i = 1$ for a step up and $Y_i = -1$ for a step down. Define $S_n = \sum_{i=1}^{n} Y_i$. In the first Markov chain, $P(\liminf_{n\to\infty} S_n = -\infty) = 1$ and $P(\limsup_{n\to\infty} S_n = \infty) = 1$, by the Hewitt-Savage zero-one law [9, Example 5.19]. In contrast, in the second Markov chain, we have $P(\lim_{n\to\infty} S_n = \infty) = 1$, by the law of large numbers [38]. Thus, in the first Markov chain, with equal chances of gaining or losing points at each step, the random walk almost surely oscillates infinitely. In contrast, in the second Markov chain, the upward bias introduced by $\varepsilon > 0$ guarantees that the total number of points will eventually diverge to $+\infty$.

We see that small changes in the transition probabilities can lead to significant changes in the behaviour and, thus, in the distances between states. This example is similar to the one presented in [24].

In the remainder, we conservatively assume that the transition function can be varied arbitrarily, that is, changes to the transition function are not restricted to specific transitions with constrained variables as in the examples. Therefore, we are interested in the continuity of the function $\delta_{\cdot}(s,t) : (S \to \mathcal{D}(S)) \to [0,1]$. This is more general than the approach of [24].

The bisimilarity distance function $\delta_{\tau_{\cdot}}$ is *lower semi-continuous* at (s,t) if for any sequence $(\tau_n)_n$ converging to τ we have $\liminf_n \delta_{\tau_n}(s,t) \geq \delta_\tau(s,t)$ and *upper semi-continuous* at (s,t) if we have $\limsup_n \delta_{\tau_n}(s,t) \leq \delta_\tau(s,t)$. Lastly, $\delta_{\tau_{\cdot}}$ is *continuous* at (s,t) if it is both upper semi-continuous and lower semi-continuous at (s,t). The Appendix, especially Appendix A, gives formal meaning to the notion of convergence used here and the underlying distances.

Observe that in all examples in Figure 1, the bisimilarity distance function $\delta_{\tau_{\cdot}}$ is lower semi-continuous at 0. The following proposition shows that this holds in general, even for $\delta_{\cdot}$, that is, allowing for arbitrary modifications of τ.

Proposition 1. *For all s, $t \in S$, the function $\delta_{\cdot}(s,t) : (S \to \mathcal{D}(S)) \to [0,1]$ is lower semi-continuous, that is, if $(\tau_n)_n$ converges to τ then $\liminf_n \delta_{\tau_n}(s,t) \geq \delta_\tau(s,t)$.*

In Figure 1c, the bisimilarity distance function is not upper semi-continuous. Specifically, $\limsup_{\epsilon \to 0} \delta_{\tau_\epsilon}(h_4,h_5) = 1$, while $\delta_{\tau_0}(h_4,h_5) = 0$. As a result, small perturbations of τ cause a jump in the distance from 0 to 1. The main goal of this paper to characterize and identify the continuity of the bisimilarity distance function for bisimilar pairs of states.

The following subsets of $S \times S$ play a key role in the subsequent discussion.

Definition 3. *The sets S^2_Δ, $S^2_{0,\tau}$, S^2_1, $S^2_{?,\tau}$, and $S^2_{0?}$ are defined by*

$$\begin{aligned}
S^2_\Delta =& \{\, (s,s) \mid s \in S \,\} \\
S^2_{0,\tau} =& \{\, (s,t) \in S \times S \mid s \neq t \wedge s \sim t \,\} \\
S^2_1 =& \{\, (s,t) \in S \times S \mid \ell(s) \neq \ell(t) \,\} \\
S^2_{?,\tau} =& (S \times S) \setminus (S^2_\Delta \cup S^2_{0,\tau} \cup S^2_1) \\
S^2_{0?} =& S^2_{0,\tau} \cup S^2_{?,\tau}
\end{aligned}$$

The first four sets form a partition of $S \times S$. Observe that the sets $S^2_{0,\tau}$ and $S^2_{?,\tau}$ depend on τ and may, therefore, change when we perturb τ, whereas the sets S^2_Δ and S^2_1 stay the same. Note that $S^2_{0?} = (S \times S) \setminus (S^2_\Delta \cup S^2_1)$. Hence, this set also stays the same if we perturb τ. Furthermore, note that $\sim \, = S^2_\Delta \cup S^2_{0,\tau}$ and for all $(s,t) \in S^2_1$, we have $\delta_\tau(s,t) = 1$.

Definition 4. *Let $\tau : S \to \mathcal{D}(S)$. The set $\mathcal{P}_\tau$ of* policies *for τ is defined by*

$$\mathcal{P}_\tau = \left\{ P : S \times S \to \mathcal{D}(S \times S) \;\middle|\; \begin{array}{l} \forall (s,t) \in S^2_{0?} : P(s,t) \in \Omega(\tau(s),\tau(t)) \\ \forall (s,t) \in S^2_\Delta \cup S^2_1 : \mathrm{support}(P(s,t)) = \{(s,t)\} \end{array} \right\}.$$

Note that a policy $P \in \mathcal{P}_\tau$ induces a Markov chain $\langle S \times S, P\rangle$. The subscript τ is omitted when clear from the context. The following proposition characterizes δ_τ in terms of policies.

Proposition 2. *For all s, $t \in S$, $\delta_\tau(s,t) = \min_{P \in \mathcal{P}} \gamma_P$, where γ_P is the probability with which (s,t) reaches S^2_1 in $\langle S \times S, P\rangle$.*

Proof Sketch. The proof follows from [2, Theorem 10.15] and [8, Theorem 8]. □

Example 6. Consider the labelled Markov chain in Figure 1a when $\varepsilon = \frac{1}{8}$. Then the probability with which (h_0, h_1) reaches S^2_1 for any policy $P \in \mathcal{P}$ is $\geq \frac{1}{5}$. The policy P that achieves the minimum probability of $\frac{1}{5}$ is defined such that $P(h_0, h_1) = \{(h_0, h_1) \mapsto \frac{3}{8}, (h_0, t) \mapsto \frac{1}{8}, (t, t) \mapsto \frac{1}{2}\}$. The Markov chain induced by P is illustrated in Figure 4 in the Appendix. Thus, $\delta_{\tau_\varepsilon}(h_0, h_1) = \frac{1}{5}$.

4 Robust Probabilistic Bisimilarity

We aim to define a notion of robust bisimilarity which is a bisimulation that is robust against perturbations of the transition function τ. As we will see in Theorem 2 below, the following definition fulfills this requirement.

Definition 5. Robust probabilistic bisimilarity *(or just robust bisimilarity), denoted $\simeq$, is defined for s, $t \in S$ as $s \simeq t$ if there exists a policy $P \in \mathcal{P}$ such that (s,t) reaches S^2_Δ with probability 1 in $\langle S \times S, P\rangle$.*

One can show using Proposition 2 that robust bisimilarity is a bisimulation (see Proposition 16 in the Appendix). Therefore, $\simeq \subseteq \sim$ and, by Theorem 1, for any s, $t \in S$ such that $s \simeq t$ we have $\delta_\tau(s,t) = 0$.

Example 7. In Figure 1a, when $\varepsilon = 0$, then $h_0 \simeq h_1$, since there exists a policy $P \in \mathcal{P}$ such that (h_0, h_1) reaches $(t,t) \in S^2_\Delta$ with probability 1 in $\langle S \times S, P\rangle$. Indeed, take $P(h_0, h_1) = \{(h_0, h_1) \mapsto \frac{1}{2}, (t,t) \mapsto \frac{1}{2}\}$ as shown in Figure 5 in the Appendix. Hence, $h_0 \simeq h_1$. Note, however, that $h_2 \not\simeq h_3$ and $h_4 \not\simeq h_5$.

The following theorem provides the rationale behind the term robust bisimilarity. It establishes that for all robust bisimilar pairs of states, small perturbations of τ result in a correspondingly small change in the distance between them.

Theorem 2. *For all s, $t \in S$, if $s \simeq t$ then the function $\delta_.(s,t) : (S \to \mathcal{D}(S)) \to [0,1]$ is continuous at τ, that is, for any sequence $(\tau_n)_n$ converging to τ we have $\lim_n \delta_{\tau_n}(s,t) = 0$.*

Proof Sketch. To build some intuition behind this theorem, we first outline the underlying idea. Let $P \in \mathcal{P}$ be the policy such that (s,t) reaches S^2_Δ with probability 1 in $\langle S \times S, P\rangle$. Then, for some k, the probability of (s,t) reaching S^2_Δ within k steps is almost one, say $1-x$, where $x>0$ is a small value. When the transition function of the Markov chain is perturbed by a small ε, the transitions of the product Markov chain $\langle S\times S, P\rangle$ are also only perturbed slightly, say by ε'. Therefore, (s,t) still reaches S^2_Δ with high probability.

To argue the last point in slightly more detail, observe that if $\varepsilon > 0$ is small enough so that $(1-\varepsilon')^k \geq 1-x$ then the probability, say p, of any individual path of length at most k from (s,t) to S^2_Δ remains at least $(1-x)\cdot p$ after the perturbation. It follows that the probability of *all* paths of length at most k from (s,t) to S^2_Δ remains at least $(1-x)\cdot(1-x) \geq 1-2x$ after the perturbation.

In the Appendix, we provide a different, formal proof using matrix norms. There we construct a graph consisting of the closed communication classes of $\langle S\times S, P\rangle$ that are reachable from (s,t). Let $P_n \in \mathcal{P}_{\tau_n}$. We then show that for all closed communication classes C reachable from (s,t) and for all pairs $(u,v)\in C$, it holds that $\lim_n \gamma_{P_n}(u,v) = \gamma_P(u,v) = 0$, by induction on the length of the longest path from C.

By Proposition 2, we have $\lim_n \delta_{\tau_n}(s,t) \leq \lim_n \gamma_{P_n}(s,t)$. Using the above result, we conclude that $\lim_n \delta_{\tau_n}(s,t) \leq \gamma_P(s,t) = 0$. □

Towards an algorithm for computing $\simeq$, let us develop another characterization of robust bisimilarity. Let $\tau : S \to \mathcal{D}(S)$. Given a policy $P \in \mathcal{P}$, we say that a set $R \subseteq S\times S$ *supports* a path $(u_1,v_1)\dots(u_n,v_n)$ in $\langle S\times S, P\rangle$ if for all $1\leq i\leq n$ we have $(u_i,v_i)\in R$ and $\mathrm{support}(P(u_i,v_i)) \subseteq R$.

Definition 6. *A* robust bisimulation *is a bisimulation $R \subseteq S\times S$ such that for all $(s,t)\in R$, there exists a policy $P\in\mathcal{P}$ such that R supports a path from (s,t) to S^2_Δ in $\langle S\times S, P\rangle$.*

Proposition 3. *For all $\tau : S\to\mathcal{D}(S)$, $\simeq$ is a robust bisimulation.*

Proof Sketch. Clearly, $\simeq$ is reflexive and symmetric. We prove in the Appendix that $\simeq$ is transitive as well and, therefore, an equivalence relation.

Next we show that $\simeq$ is a bisimulation. Assume that $s\simeq t$. Let $P\in\mathcal{P}$ be the policy such that (s,t) reaches S^2_Δ with probability 1 in $\langle S\times S, P\rangle$. Evidently, (s,t) cannot reach S^2_1 in $\langle S\times S, P\rangle$. It follows from Proposition 2 that $\delta_\tau(s,t)=0$. Thus, by Theorem 1, $s\sim t$ and we know that $\ell(s)=\ell(t)$.

Moreover, observe that for all (u,v) reachable from (s,t), (u,v) must reach S^2_Δ with probability 1 in $\langle S\times S, P\rangle$. Consequently, $\mathrm{support}(P(s,t)) \subseteq \simeq$. In fact, $\simeq$ supports a path from (s,t) to S^2_Δ in $\langle S\times S, P\rangle$, and we can conclude that $\simeq$ is a robust bisimulation. □

Proposition 4. *For any robust bisimulation $R\subseteq S\times S$, we have $R \subseteq \simeq$.*

Proof Sketch. We construct a policy $P\in\mathcal{P}$, called a maximal R-support policy in the Appendix, such that for every $(s,t)\in R$, R supports a path from (s,t) to S^2_Δ in $\langle S\times S, P\rangle$. Note that P is designed to simultaneously ensure that all pairs in R have an R-supported path to S^2_Δ in $\langle S\times S, P\rangle$. It follows from a standard result in Markov chain theory that all $(s,t)\in R$ reach S^2_Δ with probability 1 in $\langle S\times S, P\rangle$. □

It follows from Propositions 3 and 4 that $\simeq$, that is, robust bisimilarity, is the greatest robust bisimulation. This is analogous to ordinary bisimulation, where bisimilarity is the greatest bisimulation.

5 Algorithm

In this section, we present an efficient algorithm to compute robust bisimilarity; see Algorithm 1. The key idea is to define a function, Refine, such that robust bisimilarity is a fixed point (in fact, the greatest fixed point) of Refine.

Algorithm 1: Computing robust bisimilarity for labelled Markov chains

Input: A labelled Markov chain with a finite set S of states and a transition probability function $\tau : S \to \mathcal{D}(S)$, and the set of pairs of bisimilar states $\sim = S^2_{0,\tau} \cup S^2_\Delta$

Output: The set of pairs of robustly bisimilar states $R = \simeq$

$R \leftarrow \sim$
repeat
  $R_{\text{old}} \leftarrow R$
  $R \leftarrow \text{Refine}(R)$ `/* see Algorithm 2 */`
until $R = R_{\text{old}}$
return R

For any L, U with $L \subseteq U \subseteq S \times S$, write $[L, U] = \{ R \subseteq S \times S \mid L \subseteq R \subseteq U \}$ and $[L, U]_\mathcal{B} = \{ R \in [L, U] \mid R \text{ is a bisimulation} \}$.

The function Filter : $[S^2_\Delta, \sim]_\mathcal{B} \to [S^2_\Delta, \sim]$ is defined as $\text{Filter}(R) = \{ (s, t) \in R \mid \exists P \in \mathcal{P}$ such that R supports a path from (s, t) to S^2_Δ in $\langle S \times S, P \rangle \}$. The function Prune : $[S^2_\Delta, \sim] \to [S^2_\Delta, \sim]$ is defined as $\text{Prune}(R) = \{ (s, t) \in R \mid \forall (t, u) \in R : (s, u) \in R$ and $\forall (u, s) \in R : (u, t) \in R \}$. The function Bisim : $[S^2_\Delta, \sim] \to [S^2_\Delta, \sim]_\mathcal{B}$ is defined as $\text{Bisim}(R)$ being the largest bisimulation R' with $R' \subseteq R$. Given an equivalence relation R, $\text{Bisim}(R)$ can be computed in polynomial time (see Proposition 23 in the Appendix). Lastly, the function Refine : $[S^2_\Delta, \sim]_\mathcal{B} \to [S^2_\Delta, \sim]_\mathcal{B}$ is defined as $\text{Refine}(R) = \text{Bisim}(\text{Prune}(\text{Filter}(R)))$.

Proposition 5. Bisim *and* Filter *are monotone with respect to* $\subseteq$. *However,* Prune *is not.*

Proof Sketch. A counterexample for Prune is as follows. Let $S = \{s, t, u\}$, $A = \{(s, s), (t, t), (u, u), (s, t), (t, s)\}$ and $B = \{(s, s), (t, t), (u, u), (s, t), (t, s), (t, u), (u, t)\}$. A and B are symmetric and reflexive and, thus, can be visualized as an undirected graph as shown in Figure 2. Observe that $A \subseteq B$, however, $\text{Prune}(A) = A \not\subseteq \text{Prune}(B) = \{(s, s), (t, t), (u, u)\}$. □

Note that Algorithm 1 is not a typical fixed point iteration, since we do not know whether Refine is monotone.

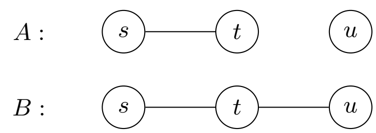


Fig. 2: Graph visualization of the relations A and B used in the proof of Proposition 5

Algorithm 2: Refine

Input: A set $R \in [S^2_\Delta, \sim]_\mathcal{B}$
Output: Refine(R)
$R \leftarrow$ Filter(R) `/* see Algorithm 3 */`
$R \leftarrow$ Prune(R) `/* see Algorithm 4 */`
$R \leftarrow$ Bisim(R)
return R

Proposition 6. *Any relation $R \subseteq S \times S$ is a robust bisimulation if and only if it is a fixed point of* Refine*.*

Proof Sketch. Let $R \subseteq S \times S$. Assume that R is a robust bisimulation. By definition, Refine(R) $\subseteq R$. Since R is a robust bisimulation, $R \subseteq$ Refine(R).

Assume that R is a fixed point of Refine, then R is a bisimulation and for every $(s, t) \in R$ there exists a policy P such that R supports a path from (s, t) to S^2_Δ in $\langle S \times S, P \rangle$. Therefore, R is a robust bisimulation. □

It follows from Propositions 4 and 6 that every fixed point of Refine is a subset of $\simeq$. Furthermore, by Propositions 3 and 6, $\simeq$ is a fixed point of Refine. Therefore, $\simeq$ is the greatest fixed point of Refine.

Let $Q \subseteq S \times S$ and s, t, u, $v \in S$. We use the following notation below: $\text{Post}((s,t)) = \text{support}(\tau(s)) \times \text{support}(\tau(t))$ and $\text{Pre}(Q) = \{ (s,t) \in S \times S \mid \text{Post}((s,t)) \cap Q \neq \varnothing \}$.

Proposition 7. *Given $R \in [\simeq, \sim]_\mathcal{B}$, for all (s,t), $(t,u) \in$ Filter(R), if $s \simeq t$ or $t \simeq u$ then $(s,u) \in$ Filter(R).*

Proof Sketch. We show that if $t \simeq u$ then $(s,u) \in$ Filter(R). The case $s \simeq t$ is similar. Write $s_1 = s$ and $t_1 = t$ and $u_1 = u$.

The idea behind the proof is that since Filter(R) $\subseteq R$, we have (s,t), $(t,u) \in R$. Since R is an equivalence relation, $(s,u) \in R$. We define P to be a maximal R-support policy, which means that for all $(s,t) \in R \cap S^2_{0?}$, $\text{support}(P(s,t)) = \text{Post}((s,t)) \cap R$. We then show that since $(s,t) \in$ Filter(R), there exists a path $(s_1,t_1), \ldots, (s_n,t_n)$ in $\langle S \times S, P \rangle$, where $s_n = t_n$.

Assume that $(t,u) \in \simeq$. Recall that $\simeq$ is a bisimulation. Since $t_1, \ldots, t_n$ is a path in the original Markov chain $\langle S, \tau \rangle$, there is also a path $u_1, \ldots, u_n$ in $\langle S, \tau \rangle$ such that $(t_i, u_i) \in \simeq$ for all $1 \leq i \leq n$. Since $\simeq \subseteq R$, there exists a path

Algorithm 3: Filter

Input: A set $R \in [S^2_\Delta, \sim]_\mathcal{B}$
Output: Filter(R)
$Q \leftarrow S^2_\Delta$
$n \leftarrow 0$
repeat
  $Q_{\text{old}} \leftarrow Q$
  foreach $(s, t) \in (R \cap \text{Pre}(Q_{\text{old}})) \setminus Q_{\text{old}}$ **do**
    $Q \leftarrow Q \cup \{(s, t)\}$
  end
  $n \leftarrow n + 1$
until $Q = Q_{\text{old}}$
return Q

Algorithm 4: Prune

Input: A set $Q \in [S^2_\Delta, \sim]$
Output: Prune(Q)
$E \leftarrow Q$
foreach $(s, t) \in Q$ **do**
  foreach $u \in S : (t, u) \in Q$ **do**
    if $(s, u) \notin Q$ **then**
      $E \leftarrow E \setminus \{(s, t), (t, u)\}$
    end
  end
end
return E

$(t_1, u_1), \ldots, (t_n, u_n)$ in $\langle S \times S, P \rangle$. Note that $(s_i, u_i) \in R$ for all $1 \leq i \leq n$. Hence, there exists a path $(s_1, u_1), \ldots, (s_n, u_n) = (t_n, u_n)$ in $\langle S \times S, P \rangle$. See Figure 3.

Since $(t_n, u_n) \in \simeq$, we know that (t_n, u_n) reaches S^2_Δ with probability 1. Therefore, there is a path $(t_n, u_n), \ldots, (t_m, u_m)$, with $t_m = u_m$ in $\langle S \times S, P \rangle$ and $(t_i, u_i) \in \simeq$ for all $n \leq i \leq m$. Thus, there exists paths $(s_1, u_1), \ldots, (s_n, u_n)$ and $(t_n, u_n), \ldots, (t_m, u_m)$ in $\langle S \times S, P \rangle$, with $(s_n, u_n) = (t_n, u_n)$. By the definition of P, R supports the same path in $\langle S \times S, P \rangle$. Hence, $(s, u) \in \text{Filter}(R)$. □

Proposition 7 allows us to prove the following Proposition.

Proposition 8. *$R \in [\simeq, \sim]_\mathcal{B}$ is a loop invariant of Algorithm 1.*

Proof Sketch. R is initialized to $\sim$, so the loop invariant holds before the loop.

Assume that the loop invariant holds before an iteration of the loop. Since $\simeq \subseteq R$, Filter is monotone and $\simeq$ is a fixed point of Refine, we have that $\simeq$ is a subset of Filter(R).

If $s \simeq t$, then $(s, t) \in \text{Filter}(R)$. Then, by Proposition 7, for all $(t, u) \in \text{Filter}(R)$ we have $(s, u) \in \text{Filter}(R)$ and for all $(u, s) \in \text{Filter}(R)$ we have $(u, t) \in$

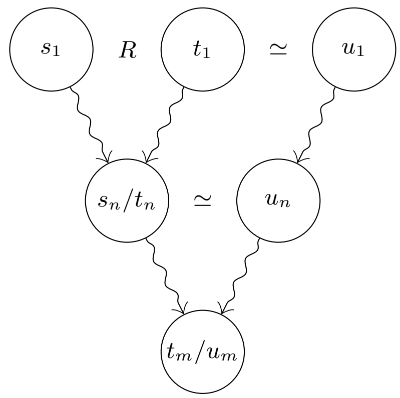


Fig. 3: Illustration of the proof of Proposition 7

Filter(R). Hence, $(s,t) \in$ Prune(Filter(R)), and we have that $\simeq$ is a subset of Prune(Filter(R)).

Bisim is monotone, therefore, $\simeq$ is a subset of Bisim(Prune(Filter(R))) and Refine(R). By the definition of Bisim, Refine(R) $\in [\simeq, \sim]_{\mathcal{B}}$. Thus, the loop invariant is maintained in each iteration of the loop. $\square$

Using the loop invariant established in Proposition 8, we can now prove the correctness of Algorithm 1.

Theorem 3. *Algorithm 1 computes the set $\simeq$.*

Proof Sketch. It is immediate from the definitions of Bisim, Filter and Prune that Refine(R) $\subseteq R$ holds for all $R \subseteq S \times S$. Therefore, Algorithm 1 is a standard fixed-point iteration. By Proposition 8, $\simeq \subseteq R$, thus, it computes a fixed point of Refine greater than or equal to $\simeq$. Since $\simeq$ is the greatest fixed point of Refine, we can conclude that Algorithm 1 computes $\simeq$. $\square$

In the Appendix, we show that Algorithm 1 has a time complexity of $\mathcal{O}(n^6)$, where $n = |S|$. The computational bottleneck is the function Filter (see Proposition 26).

6 Experiments

To evaluate the efficiency and usefulness of our robust bisimilarity algorithm, we implemented it in the widely used probabilistic model checker PRISM [30], an open-source tool providing quantitative verification and analysis of several types of probabilistic models, including labelled Markov chains.

Implementation. PRISM's implementation of the traditional (i.e., non-robust) bisimilarity algorithm Bisim is a standard partition-refinement approach which uses the signature-based method of Derisavi [12]. Let P be the current partition and E_P be the set of equivalence classes in P. Then the new partition is computed as $\{ (s,t) \in P \mid \forall B \in E_P : \tau(s)(B) = \tau(t)(B) \}$.

We implemented Algorithm 1 in Java as part of PRISM's explicit-state model checking engine. Each state and equivalence class (referred to as a block) is represented by an integer ID. The current partition of the state space is tracked by an array that is indexed by state IDs and contains the corresponding block IDs. To store the list of successors for each state, we use a map. Bisim is run on the input Markov chain to obtain the set of bisimilar states.

The function Filter first constructs R from the current partition and initializes Q to S^2_Δ. In our approach, R is implemented as an array indexed by block IDs, with each block containing a list of states. Conversely, Q is implemented as an array indexed by state IDs, with each state storing the set of states related to it. Predecessors of Q in R are added to Q until a fixed point is reached. A pair of states $(s,t) \in R \setminus Q$ is a predecessor of Q if they have some successors that are related in Q. Specifically, there must exists successors s' and t' of s and t, respectively, such that $t' \in Q[s']$ and vice versa.

Prune constructs a new partition of the state space by grouping states in the same block if they have the same neighbourhood in Q, that is, they are related to the same states. In other words, s and t are placed in the same block if $Q[s] = Q[t]$ holds. Bisim is then called with the current partition passed as the initial partition. This process continues until no further refinement is possible, resulting in the set of robustly bisimilar states. Finally, the minimized Markov chain is constructed.

Experimental setup. We evaluated our algorithm by applying it to all discrete-time Markov chains from the Quantitative Verification Benchmark Set (QVBS) [21], a comprehensive collection of probabilistic models which is designed as a benchmark suite for quantitative verification and analysis tools and is the foundation of the Quantitative Verification Competition (QComp), which compares the performance, versatility, and usability of such tools.

For an additional source of models, we also use jpf-probabilistic [18]. Java PathFinder (JPF) [43] is the most popular model checker for Java code, and the JPF extension jpf-probabilistic provides Java implementations of sixty randomized algorithms [18]. JPF can be used in tandem with PRISM to check properties of these algorithms and supplement JPF's qualitative results with quantitative information. A description of the subset of these algorithms utilized in our study is provided in Appendix J.

In order to explore both the benefits and the efficiency of our algorithm, we run both the robust and traditional bisimilarity algorithms on all models. For the latter, we use PRISM's existing implementation, in order to provide a comparable implementation. Our experiments were run on a MacBook with an M1 chip and 16GB memory, and with the Java virtual machine limited to 8GB.

Results. Table 1 shows results for all benchmarks where the minimized models obtained by traditional bisimilarity and robust bisimilarity differ. These are of

Benchmark				Bisimilarity		Robust Bisimilarity	
Name (prop.)	Parameters		States	Min	Time	Min	Time
brp (p1)	N=32	MAX=2	1349	646	0.036	901	0.054
		MAX=3	1766	871	0.043	1127	0.062
		MAX=4	2183	1096	0.051	1353	0.068
		MAX=5	2600	1321	0.058	1579	0.075
	N=64	MAX=2	2693	1286	0.080	1797	0.105
		MAX=3	3526	1735	0.084	2247	0.119
		MAX=4	4359	2184	0.103	2697	0.130
		MAX=5	5192	2633	0.132	3147	0.167
brp (p4)	N=32	MAX=2	1349	10	0.012	711	3.690
		MAX=3	1766	12	0.013	937	6.291
		MAX=4	2183	14	0.018	1163	9.331
		MAX=5	2600	16	0.021	1389	13.952
	N=64	MAX=2	2693	10	0.017	1415	27.299
		MAX=3	3526	12	0.015	1865	45.031
		MAX=4	4359	14	0.016	2315	69.949
		MAX=5	5192	16	0.018	2765	102.941
crowds	CS=5	TR=3	1198	41	0.018	505	0.231
(positive)		TR=4	3515	61	0.021	1484	1.304
		TR=5	8653	81	0.038	3659	7.575
		TR=6	18817	101	0.071	7969	34.765
	CS=10	TR=3	6563	41	0.024	2320	8.296
		TR=4	30070	61	0.078	10524	196.233
		TR=5	111294	81	0.190	38770	2946.840
oscillators	T=6	N=3	57	28	0.007	38	0.009
(power)	T=8	N=6	1717	1254	0.037	1255	0.037
		N=8	6436	5148	0.100	5149	0.122
oscillators	T=6	N=3	57	28	0.007	38	0.008
(time)	T=8	N=6	1717	1254	0.032	1255	0.036
		N=8	6436	5148	0.111	5149	0.115
set isolation	U=13	ST=3	8196	19	0.029	27	21.885
(good sample)		ST=4	8196	20	0.029	26	24.325
		ST=5	8196	21	0.032	25	24.330
		ST=6	8196	22	0.031	24	25.162

Table 1: The results of the benchmarks for which the minimized models differ.

particular interest because they are instances where our algorithm identifies that a model minimized in traditional fashion is not robust. The property used for each benchmark dictates the labelling used for the model. In the table, *Min* denotes the number of states in the minimized model and *Time* denotes the amount of time taken (in seconds) to compute bisimilarity.

The results are promising: robust bisimilarity, although (unsurprisingly) slower than traditional bisimilarity, remains practical across a wide range of standard benchmarks. Table 2 displays some of the largest models per benchmark along

Benchmark			Robust Bisimilarity	
Name	Property	States	Min States	Time
crowds	positive	111294	38770	2946.84
egl	messages	115710	131	153.01
herman	steps	32768	612	25.29
oscillators	power	24311	17877	0.42

Table 2: Models with the maximum state space per benchmark.

Benchmark			Average % Increase	
Name	Property	Instances	States	Time
brp	p1	12	27.93	28.95
	p2	12	7.76	80.54
	p4	12	9193.43	142193.57
crowds	positive	7	12306.72	273258.24
egl	messages	6	-	20693.83
erdös-rényi model	connected	18	-	799.07
fair biased coin	heads	9	-	0.00
has majority element	incorrect	24	-	16.08
herman	steps	7	-	297.99
leader-sync	elected & time	18	-	518.98
haddad-monmege	target	3	-	0.00
oscillators	power & time	14	5.12	11.73
pollards factorization	input	8	-	0.00
queens	success	6	-	1193.93
set isolation	good sample	4	25.06	79035.95
Total		160	1231.68	25589.22

Table 3: Summary of all benchmarks with the change due to robust bisimilarity.

Benchmark					Bisimilarity	
Name	Property	Parameters		States	Min States	Time
crowds	positive	CS=10	TR=6	352535	101	0.57
egl	messages	N=5	L=8	156670	171	0.87
	unfair	N=5	L=2	33790	229	0.10
			L=4	74750	469	0.26
			L=6	115710	709	0.40
			L=8	156670	949	0.70
nand	reliable	N=20	K=1	78332	39982	0.81
			K=2	154942	102012	1.89
			K=3	231552	164042	3.67
queens	success		N=10	23492	527	0.08

Table 4: Models for which robust bisimilarity results in an `OutOfMemoryError`.

with the time required to compute robust bisimilarity. The longest time recorded is 2947.84 seconds (49.13 minutes) for the *crowds* benchmark.

The complete set of experiments includes 170 models, of which 160 are aggregated in Table 3. This table presents the average percentage increase in both the state space of the minimized model and the computation time for robust bisimilarity compared to traditional bisimilarity. The *crowds* benchmark exhibits the largest average percentage increase for both metrics. The reported values may seem large, however, it is important to note that the traditional bisimilarity algorithm required a maximum of 2.14 seconds per model in this table.

Furthermore, robust bisimilarity was successfully computed in less than a minute for 152 models (over 89%). Of the total set of models, the remaining 10 (approximately 6%), listed in Table 4, could only be minimized using traditional bisimilarity, as the robust bisimilarity computation ran out of available memory before completion. This issue occurred with all instances of the *nand* benchmark and half of the instances of the *egl* benchmark.

Ultimately, robust bisimilarity proves feasible for large models, despite needing more resources than traditional bisimilarity. Furthermore, it offers a more reliable method of determining equivalence, which can be particularly beneficial in mission-critical applications, which require a higher level of precision.

7 Conclusions and Future Work

To address the lack of robustness of probabilistic bisimilarity, we have introduced the concept of robust bisimilarity for labelled Markov chains. Robust bisimilarity ensures that the distance function remains continuous even under perturbations of transition probabilities. Additionally, we have presented a computationally efficient algorithm, with experimental results demonstrating that robust bisimilarity is plausible for large-scale verification tasks.

Our work opens new avenues for future exploration. First, a logical characterization of robust bisimilarity could provide deeper insights. Second, while we have established in Theorem 2 that robust bisimilarity is a sufficient condition for continuity, we conjecture that for bisimilar states, robust bisimilarity is in fact also a necessary condition for continuity. We also aim to define continuity for non-bisimilar state pairs, to complete the theoretical characterization of robustness. Thirdly, in [24] it was shown that when the distances are discounted (i.e., differences that manifest themselves later count less), the distance function becomes continuous. This raises the question: can we identify the properties for which the discontinuity is relevant? The examples suggest that these are long-term properties. Finally, we plan to investigate specific types of perturbations of the transition probabilities, such as those that do not introduce new transitions, preserving the graph structure, as seen in Figures 1a and 1c, unlike the perturbation shown in Figure 1b which adds a new transition.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Appendix

A Metric Topology

Definition 7. *The function* $d_E : [0,1] \times [0,1] \to [0,1]$ *is defined by*

$$d_E(r,s) = |r - s|.$$

Definition 8. *The function* $d_F : (X \to Y) \times (X \to Y) \to [0,1]$ *is defined by*

$$d_F(f,g) = \max_{x \in X} d(f(x), g(x)).$$

Definition 9. *The function* $d_{TV} : \mathcal{D}(X) \times \mathcal{D}(X) \to [0,1]$ *is defined by*

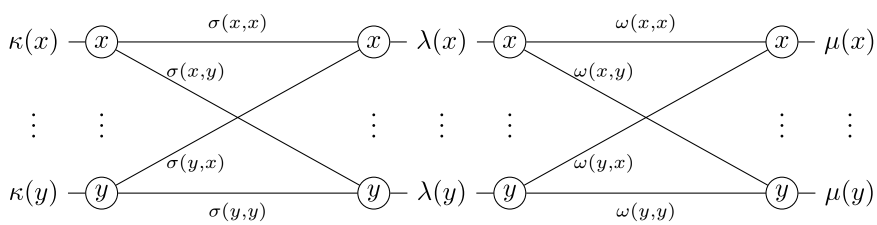
$$d_{TV}(\mu,\nu) = \max_{x \in X} |\mu(x) - \nu(x)|.$$

B Couplings

Lemma 1. *For all* κ, λ, μ, $\nu \in \mathcal{D}(X)$ *and* $\omega \in \Omega(\lambda,\mu)$, *there exist* $\pi \in \Omega(\kappa,\mu)$ *and* $\rho \in \Omega(\lambda,\nu)$ *such that* $d_{TV}(\omega,\pi) \leq d_{TV}(\kappa,\lambda)$ *and* $d_{TV}(\omega,\rho) \leq d_{TV}(\mu,\nu)$.

Proof. We only prove the existence of coupling π, as ρ can be dealt with similarly. Let κ, λ, $\mu \in \mathcal{D}(X)$ and $\omega \in \Omega(\lambda,\mu)$.

The proof is structured as follows. We first construct a coupling σ of κ and λ. Next, we compose this coupling σ with the coupling ω of λ and μ, obtaining a coupling π of κ and μ. Finally, we show that this coupling π satisfies the desired property.



We first construct $\sigma \in \Omega(\kappa,\lambda)$. Set $\sigma(x,x) = \min\{\kappa(x), \lambda(x)\}$ for all $x \in X$. It remains to consider κ', λ' with

$$\kappa'(x) = \begin{cases} 0 & \text{if } \lambda(x) \geq \kappa(x) \\ \kappa(x) - \lambda(x) & \text{otherwise} \end{cases}$$

and

$$\lambda'(x) = \begin{cases} 0 & \text{if } \lambda(x) \leq \kappa(x) \\ \lambda(x) - \kappa(x) & \text{otherwise.} \end{cases}$$

Note that $\lambda'(X) = \kappa'(X)$. By means of the North-West corner method, we construct a $\sigma' \in \Omega(\kappa',\lambda')$. Hence, for all $x \in X$, $\sigma'(x,X) = \kappa'(x)$ and $\sigma'(X,x) =$

$\lambda'(x)$. Since for all $x \in X$, $\kappa'(x) = 0$ or $\lambda'(x) = 0$, we can conclude that $\sigma'(x,x) = 0$. We complete σ by setting $\sigma(x,y) = \sigma'(x,y)$ for all x, $y \in X$ with $x \neq y$. It remains to show that $\sigma \in \Omega(\kappa, \lambda)$. For all $x \in X$,

$$\begin{aligned}
\sigma(x,X) &= \sigma(x,x) + \sigma(x, X \setminus \{x\}) \\
&= \min\{\kappa(x), \lambda(x)\} + \sigma'(x,X) \\
&= \min\{\kappa(x), \lambda(x)\} + \kappa'(x) \\
&= \kappa(x)
\end{aligned}$$

and

$$\begin{aligned}
\sigma(X,x) &= \sigma(x,x) + \sigma(X \setminus \{x\}, x) \\
&= \min\{\kappa(x), \lambda(x)\} + \sigma'(X,x) \\
&= \min\{\kappa(x), \lambda(x)\} + \lambda'(x) \\
&= \lambda(x).
\end{aligned}$$

Let $\omega \in \Omega(\lambda, \mu)$. We define

$$\pi(x,y) = \sum_{z \in X \wedge \lambda(z) \neq 0} \frac{\sigma(x,z)\,\omega(z,y)}{\lambda(z)}.$$

To conclude that $\pi \in \Omega(\kappa, \mu)$, we observe that for all $x \in X$,

$$\begin{aligned}
&\pi(x,X) \\
&= \sum_{y \in X} \sum_{z \in X \wedge \lambda(z) \neq 0} \frac{\sigma(x,z)\,\omega(z,y)}{\lambda(z)} \\
&= \sum_{z \in X \wedge \lambda(z) \neq 0} \frac{\sigma(x,z)}{\lambda(z)} \sum_{y \in X} \omega(z,y) \\
&= \sum_{z \in X \wedge \lambda(z) \neq 0} \frac{\sigma(x,z)}{\lambda(z)} \lambda(z) \qquad [\omega(z,X) = \lambda(z) \text{ since } \omega \in \Omega(\lambda,\mu)] \\
&= \sum_{z \in X \wedge \lambda(z) \neq 0} \sigma(x,z) \\
&= \sigma(x,X) \qquad [\text{if } \lambda(z) = 0 \text{ then } \sigma(x,z) \leq \sigma(X,z) = \lambda(z) = 0 \text{ since } \sigma \in \Omega(\kappa,\lambda)] \\
&= \kappa(x) \qquad [\sigma \in \Omega(\kappa,\lambda)]
\end{aligned}$$

and

$$
\begin{aligned}
&\pi(X, x) \\
&= \sum_{y \in X} \sum_{z \in X \wedge \lambda(z) \neq 0} \frac{\sigma(y, z)\, \omega(z, x)}{\lambda(z)} \\
&= \sum_{z \in X \wedge \lambda(z) \neq 0} \frac{\omega(z, x)}{\lambda(z)} \sum_{y \in X} \sigma(y, z) \\
&= \sum_{z \in X \wedge \lambda(z) \neq 0} \frac{\omega(z, x)}{\lambda(z)} \lambda(z) \qquad [\sigma(X, z) = \lambda(z) \text{ since } \sigma \in \Omega(\kappa, \lambda)] \\
&= \sum_{z \in X \wedge \lambda(z) \neq 0} \omega(z, x) \\
&= \omega(X, x) \qquad [\text{if } \lambda(z) = 0 \text{ then } \omega(z, x) \leq \omega(z, X) = \lambda(z) = 0 \text{ as } \omega \in \Omega(\lambda, \mu)] \\
&= \mu(x) \qquad [\omega \in \Omega(\lambda, \mu)]
\end{aligned}
$$

It remains to show that $d_{TV}(\omega, \pi) \leq d_{TV}(\kappa, \lambda)$. Let x, $y \in X$. It suffices to prove that $|\omega(x, y) - \pi(x, y)| \leq d_{TV}(\kappa, \lambda)$. We distinguish the following cases.

- Assume that $\lambda(x) = 0$. Then $\omega(x, y) \leq \omega(x, X) = \lambda(x) = 0$ since $\omega \in \Omega(\lambda, \mu)$ and, hence, $\omega(x, y) = 0$. Furthermore, $\pi(x, y) \leq \pi(x, X) = \kappa(x)$ since $\pi \in \Omega(\kappa, \mu)$. Hence, $|\omega(x, y) - \pi(x, y)| \leq \kappa(x) = \kappa(x) - \lambda(x) \leq d_{TV}(\kappa, \lambda)$.
- Assume that $\kappa(x) = 0$. Then $\pi(x, y) \leq \pi(x, X) = \kappa(x) = 0$ since $\pi \in \Omega(\kappa, \mu)$ and, hence, $\pi(x, y) = 0$. Furthermore, $\omega(x, y) \leq \omega(x, X) = \lambda(x)$ since $\omega \in \Omega(\lambda, \mu)$. Hence, $|\omega(x, y) - \pi(x, y)| \leq \lambda(x) = \lambda(x) - \kappa(x) \leq d_{TV}(\kappa, \lambda)$.
- Assume that $0 < \lambda(x) \leq \kappa(x)$. Then

$$
\begin{aligned}
\pi(x, y) &= \sum_{z \in X \wedge \lambda(z) \neq 0} \frac{\sigma(x, z)\, \omega(z, y)}{\lambda(z)} \\
&= \omega(x, y) + \sum_{z \in X \wedge \lambda(z) \neq 0 \wedge z \neq x} \frac{\sigma(x, z)\, \omega(z, y)}{\lambda(z)} \qquad [\sigma(x, x) = \lambda(x) > 0]
\end{aligned}
$$

and

$$
\begin{aligned}
&\sum_{z \in X \wedge \lambda(z) \neq 0 \wedge z \neq x} \frac{\sigma(x, z)\, \omega(z, y)}{\lambda(z)} \\
&\leq \sum_{z \in X \wedge \lambda(z) \neq 0 \wedge z \neq x} \sigma(x, z) \qquad [\omega(z, y) \leq \omega(z, X) = \lambda(z) \text{ since } \omega \in \Omega(\lambda, \mu)] \\
&= \sigma(x, X \setminus \{x\}) \\
&\qquad [\text{if } \lambda(z) = 0 \text{ then } \sigma(x, z) \leq \sigma(X, z) = \lambda(z) = 0 \text{ since } \sigma \in \Omega(\kappa, \lambda)] \\
&= \sigma'(x, X) \\
&= \kappa'(x) \qquad [\sigma' \in \Omega(\kappa', \lambda')] \\
&= \kappa(x) - \lambda(x) \\
&\leq d_{TV}(\kappa, \lambda).
\end{aligned}
$$

Hence, $|\omega(x,y) - \pi(x,y)| \leq d_{TV}(\kappa,\lambda)$.

– Assume that $0 < \kappa(x) \leq \lambda(x)$. Then

$$\begin{aligned}
&\pi(x,y)\\
&= \sum_{z \in X \wedge \lambda(z) \neq 0} \frac{\sigma(x,z)\,\omega(z,y)}{\lambda(z)}\\
&= \frac{\sigma(x,x)\,\omega(x,y)}{\lambda(x)} \qquad [\lambda(x) > 0 \text{ and } \sigma(x, X \setminus \{x\}) = 0 \text{ since } \kappa(x) \leq \lambda(x)]\\
&= \frac{\kappa(x)\,\omega(x,y)}{\lambda(x)} \qquad [\sigma(x,x) = \kappa(x) \text{ since } \kappa(x) \leq \lambda(x)]\\
&\leq \omega(x,y) \qquad [\kappa(x) \leq \lambda(x)]
\end{aligned}$$

Hence,

$$\begin{aligned}
&\omega(x,y) - \pi(x,y)\\
&= \left(1 - \frac{\kappa(x)}{\lambda(x)}\right) \omega(x,y)\\
&= (\lambda(x) - \kappa(x))\,\frac{\omega(x,y)}{\lambda(x)}\\
&\leq \lambda(x) - \kappa(x) \qquad [\omega(x,y) \leq \omega(x,X) = \lambda(x) \text{ since } \omega \in \Omega(\lambda,\mu)]\\
&\leq d_{TV}(\kappa,\lambda).
\end{aligned}$$

Because $\pi(x,y) \leq \omega(x,y)$ and $\omega(x,y) - \pi(x,y) \leq d_{TV}(\kappa,\lambda)$, we have that $|\omega(x,y) - \pi(x,y)| \leq d_{TV}(\kappa,\lambda)$.

□

Corollary 1. *For all κ, λ, μ, $\nu \in \mathcal{D}(X)$ and $\omega \in \Omega(\lambda,\mu)$, there exist $\pi \in \Omega(\kappa,\nu)$ such that $d_{TV}(\omega,\pi) \leq d_{TV}(\kappa,\lambda) + d_{TV}(\mu,\nu)$.*

Proof. Let κ, λ, μ, $\nu \in \mathcal{D}(X)$ and $\omega \in \Omega(\lambda,\mu)$. By Lemma 1, there exists $\rho \in \Omega(\lambda,\nu)$ such that $d_{TV}(\omega,\rho) \leq d_{TV}(\mu,\nu)$ and, again using Lemma 1, there exists $\pi \in \Omega(\kappa,\nu)$ such that $d_{TV}(\rho,\pi) \leq d_{TV}(\kappa,\lambda)$. Therefore,

$$\begin{aligned}
d_{TV}(\omega,\pi) &\leq d_{TV}(\omega,\rho) + d_{TV}(\rho,\pi) \qquad [\text{triangle inequality}]\\
&\leq d_{TV}(\kappa,\lambda) + d_{TV}(\mu,\nu).
\end{aligned}$$

□

Proposition 9. *For all $n \in \mathbb{N}$, let μ_n, $\nu_n \in \mathcal{D}(S)$ and $\omega_n \in \Omega(\mu_n,\nu_n)$. If $(\mu_n)_n$ and $(\nu_n)_n$ converge to μ and ν then $\liminf_n \omega_n \in \Omega(\mu,\nu)$ and $\limsup_n \omega_n \in \Omega(\mu,\nu)$.*

Proof. Let μ_n, $\nu_n \in \mathcal{D}(S)$ and $\omega_n \in \Omega(\mu_n,\nu_n)$ for all $n \in \mathbb{N}$. Assume that $(\mu_n)_n$ and $(\nu_n)_n$ converge to μ and ν. Let $s \in S$. Then

$$\begin{aligned}
\liminf_n \omega_n(s,S) &= \liminf_n \mu_n(s) \qquad [\omega_n \in \Omega(\mu_n,\nu_n)]\\
&= \lim_n \mu_n(s) \qquad [(\mu_n)_n \text{ and, hence, } (\mu_n(s))_n \text{ is converging}]\\
&= \mu(s)
\end{aligned}$$

and

$$\begin{aligned}
\liminf_n \omega_n(S, s) &= \liminf_n \nu_n(s) && [\omega_n \in \Omega(\mu_n, \nu_n)] \\
&= \lim_n \nu_n(s) && [(\nu_n)_n \text{ and, hence, } (\nu_n(s))_n \text{ is converging}] \\
&= \nu(s).
\end{aligned}$$

We can prove $\limsup_n \omega_n \in \Omega(\mu, \nu)$ similarly. □

C Policies

Proposition 10. *For all σ, $\tau : S \to \mathcal{D}(S)$, and $P \in \mathcal{P}_\sigma$, there exists $Q \in \mathcal{P}_\tau$ such that $d_F(P, Q) \leq 2\, d_F(\sigma, \tau)$.*

Proof. Let σ, $\tau : S \to \mathcal{D}(S)$, and $P \in \mathcal{P}_\sigma$. For each $(s, t) \in S^2_{0?}$, $P(s, t) \in \Omega(\sigma(s), \sigma(t))$ and by Corollary 1 there exists $\omega_{st} \in \Omega(\tau(s), \tau(t))$ such that

$$d_{TV}(P(s, t), \omega_{st}) \leq d_{TV}(\sigma(s), \tau(s)) + d_{TV}(\sigma(t), \tau(t)) \leq 2\, d_F(\sigma, \tau). \tag{1}$$

We define Q by

$$Q(s, t) = \begin{cases} P(s, t) & \text{if } (s, t) \in S^2_\Delta \cup S^2_1 \\ \omega_{st} & \text{otherwise.} \end{cases}$$

We leave it to the reader to verify that $Q \in \mathcal{P}_\tau$. It suffices to show that for all s, $t \in S$, $d_{TV}(P(s, t), Q(s, t)) \leq 2\, d_F(\sigma, \tau)$. If $(s, t) \in S^2_\Delta \cup S^2_1$ then this is vacuously true. Otherwise, it follows from (1). □

Proposition 11. *For all $n \in \mathbb{N}$, let $\tau_n : S \to \mathcal{D}(S)$ and $P_n \in \mathcal{P}_{\tau_n}$. If $(\tau_n)_n$ converges to τ then $\liminf_n P_n \in \mathcal{P}_\tau$ and $\limsup_n P_n \in \mathcal{P}_\tau$.*

Proof. Let $\tau_n : S \to \mathcal{D}(S)$ and $P_n \in \mathcal{P}_{\tau_n}$ for all $n \in \mathbb{N}$. Assume that $(\tau_n)_n$ converges to τ. Let s, $t \in S$. We distinguish two cases.

- Assume that $(s, t) \in S^2_{0?}$. Since $(\tau_n(s))_n$ and $(\tau_n(t))_n$ converge to $\tau(s)$ and $\tau(t)$, and $P_n(s, t) \in \Omega(\tau_n(s), \tau_n(t))$ for all $n \in \mathbb{N}$, we can conclude from Proposition 9 that $\liminf_n P_n(s, t) \in \Omega(\tau(s), \tau(t))$.
- Assume that $(s, t) \in S^2_\Delta \cup S^2_1$. Since for all $n \in \mathbb{N}$, $\text{support}(P_n(s, t)) = \{(s, t)\}$, we can conclude that $\text{support}(\liminf_n P_n(s, t)) = \{(s, t)\}$.

We can prove $\limsup_n P_n \in \mathcal{P}_\tau$ similarly. □

D Value Function

Definition 10. *The function $\Gamma : (S \times S \to \mathcal{D}(S \times S)) \to (S \times S \to [0, 1]) \to (S \times S \to [0, 1])$ is defined by*

$$\Gamma_P(d)(s, t) = \begin{cases} 0 & \text{if } (s, t) \in S^2_\Delta \\ 1 & \text{if } (s, t) \in S^2_1 \\ P(s, t) \cdot d & \text{otherwise,} \end{cases}$$

where $P(s, t) \cdot d = \sum_{u, v \in S} P(s, t)(u, v)\; d(u, v)$.

For each $P : S \times S \to \mathcal{D}(S \times S)$, Γ_P is a monotone function from the complete lattice $S \times S \to [0,1]$ to itself (see, for example, [40, Proposition 6.1.3]). According to the Knaster-Tarski fixed point theorem, Γ_P has a least fixed point, which we denote by γ_P. Note that $\langle S \times S, P \rangle$ is a Markov chain.

Recall that $\mathcal{P}_\tau$ is the set of policies for τ and that the subscript τ is omitted when clear from the context.

Theorem 4 ([2, Theorem 10.15]). *For all $P \in \mathcal{P}$ and s, $t \in S$, $\gamma_P(s,t)$ is the probability of reaching S_1^2 from (s,t) in $\langle S \times S, P \rangle$.*

Theorem 5 ([8, Theorem 8]). $\delta_\tau = \min\limits_{P \in \mathcal{P}} \gamma_P$.

The above theorem is proved by showing that $\delta_\tau \sqsubseteq \gamma_P$ for all $P \in \mathcal{P}$ and that there exists $P \in \mathcal{P}$ such that $\delta_\tau = \gamma_P$.

Proof (of Proposition 2). Follows immediately from Theorems 4 and 5. □

Definition 11. *Let s, $t \in S$.*

- *A policy $P \in \mathcal{P}$ is* optimal for (s,t) *if $\gamma_P(s,t) = \delta_\tau(s,t)$.*
- *A policy $P \in \mathcal{P}$ is* optimal *if for all s, $t \in S$, P is optimal for (s,t).*

Note that from Theorem 5 we can conclude that optimal policies exist.

Proposition 12. *For all $P \in \mathcal{P}$, the following are equivalent.*

1. *P is optimal*
2. *$\Gamma_P(\delta_\tau) = \delta_\tau$*
3. *$\Gamma_P(\delta_\tau) \sqsubseteq \delta_\tau$*

Proof. Let $P \in \mathcal{P}$. We prove three implications.

1. ⇒ 2. Assume that P is optimal. Then $\gamma_P = \delta_\tau$. Therefore,

$$\Gamma_P(\delta_\tau) = \Gamma_P(\gamma_P) = \gamma_P = \delta_\tau.$$

2. ⇒ 3. Immediate.

3. ⇒ 1. Assume that $\Gamma_P(\delta_\tau) \sqsubseteq \delta_\tau$, that is, δ_τ is a pre-fixed point of Γ_P. By the Knaster-Tarski fixed point theorem (see, for example, [11, Theorem 2.35]), γ_P is the least pre-fixed point of Γ_P. Hence, $\gamma_P \sqsubseteq \delta_\tau$. By Theorem 5, $\delta_\tau \sqsubseteq \gamma_P$. Therefore, P is optimal.

□

Proposition 13. *Let $P \in \mathcal{P}$. If C is a closed communication class of $\langle S \times S, P \rangle$ then*

1. *$C = \{(s,t)\}$ for some $(s,t) \in S_\Delta^2 \cup S_1^2$, or*
2. *$C \subseteq S_{0,\tau}^2$.*

Proof. Let $P \in \mathcal{P}$. From the definition of $\mathcal{P}$ it immediately follows that the sets defined in 1. are closed communication classes of $\langle S \times S, P \rangle$.

Towards a contradiction, assume that C is a closed communication class of $\langle S \times S, P \rangle$ not satisfying 1. and 2. Then there exists $(s,t) \in C$ with $s \not\sim t$ and $\ell(s) = \ell(t)$. Hence, $\delta_\tau(s,t) > 0$ by Theorem 1. By Theorem 5, $\gamma_P(s,t) > 0$. Hence, there exists a path in $\langle S \times S, P \rangle$ from (s,t) to some (u,v) with $\ell(u) \neq \ell(v)$. Since $\ell(s) = \ell(t)$, we can conclude that the path is nonempty. Therefore, there exists a nonempty path from (s,t) to the closed communication class $\{(u,v)\}$. This contradicts that the communication class C is closed. $\square$

E Linear Algebra

We denote the infinity norm by $\| \cdot \|$. Recall that for an n-vector x, we have that $\|x\| = \max_{0 \leq i < n} |x_i|$ and for an $m \times n$-matrix A, we have that $\|A\| = \max_{0 \leq i < m} \sum_{0 \leq j < n} |A_{ij}|$. Given n-vectors x and y, we write $x \lneqq y$ if $x_i \leq y_i$ for all $0 \leq i < n$ and $x_j < y_j$ for some $0 \leq j < n$. We denote constant vectors and matrices simply by their value. A matrix A is *strictly substochastic* if $A1 \lneqq 1$.

The definition of an irreducible matrix from [4] is the following, however, we will rely only on the characterisation of irreducibility in Theorem 6. An $n \times n$ matrix A is *cogredient* to a matrix E if for some permutation matrix P, $PAP^t = E$. A is *reducible* if it is cogredient to $E = \left[\begin{smallmatrix} B & 0 \\ C & D \end{smallmatrix}\right]$, where B and D are square matrices, or if $n = 1$ and $A = 0$. Otherwise, A is *irreducible*.

Theorem 6 ([4, Theorem 2.2.1]). *A nonnegative $n \times n$-matrix A is irreducible if and only if for every $0 \leq i, j < n$ there exists $m > 0$ such that $A^m_{ij} > 0$.*

Proposition 14. *Let A be an irreducible and strictly substochastic $n \times n$-matrix. Then $I - A$ is invertible.*

Proof. Let A be an irreducible and strictly substochastic $n \times n$-matrix. Then A is an irreducible nonnegative square matrix. Since A strictly substochastic, $A1 \lneqq 1$. [4, Theorem 2.1.11] states that for an irreducible nonnegative square matrix A, if $Ax \lneqq x$ for some $x \gneqq 0$ then $\rho(A) < 1$, where $\rho(A)$ is the spectral radius of A. Since $A1 \lneqq 1$, we have thus that $\rho(A) < 1$. Towards a contradiction, assume that $I - A$ is not invertible. Then there exists $x \neq 0$ with $(I - A)x = 0$. That is, $Ax = x$. Thus, one is an eigenvalue of A, and so $\rho(A) \geq 1$. $\square$

F Probabilistic Bisimilarity Distances

The variable *tails* evaluates to one in the red states and zero in the blue states, that is, $\mathit{tails} = \{h_2 \mapsto 0, h_3 \mapsto 0, t_4 \mapsto 1, t_5 \mapsto 1\}$. Let $\varphi_1 = \mu V.\, \mathrm{next}(\mathit{tails} \vee V) \ominus \frac{1}{2}$, then the computation of the quantitative μ-calculus formula of Example 3 is as follows,

$$\begin{aligned}
[\![\varphi_1]\!] &= [\![\mu V.\, \mathrm{next}(\mathit{tails} \vee V) \ominus \tfrac{1}{2}]\!] \\
&= \inf\{f \in \mathcal{F} \mid f = [\![\mathrm{next}(\mathit{tails} \vee V) \ominus \tfrac{1}{2}]\!]\} \\
&= \inf\{f \in \mathcal{F} \mid f = [\![\mathrm{next}(\mathit{tails} \vee V)]\!] \ominus \tfrac{1}{2}\} \\
&= \inf\{f \in \mathcal{F} \mid f = \mathrm{Next}([\![\mathit{tails} \vee V]\!]) \ominus \tfrac{1}{2}\} \\
&= \inf\{f \in \mathcal{F} \mid f = \mathrm{Next}([\![\mathit{tails}]\!] \sqcup [\![V]\!]) \ominus \tfrac{1}{2}\} \\
&= \inf\{f \in \mathcal{F} \mid f = \mathrm{Next}([\mathit{tails}] \sqcup f) \ominus \tfrac{1}{2}\}
\end{aligned}$$

$$\begin{aligned}
[\![\varphi_1]\!](h_0) &= \mathrm{Next}([\mathit{tails}] \sqcup [\![\varphi_1]\!])(h_0) \ominus \tfrac{1}{2} \\
&= \tfrac{1}{2}([\mathit{tails}] \sqcup [\![\varphi_1]\!])(h_0) + \tfrac{1}{2}([\mathit{tails}] \sqcup [\![\varphi_1]\!])(t) \ominus \tfrac{1}{2} \\
&= \tfrac{1}{2}([\![\varphi_1]\!](h_0)) + \tfrac{1}{2}([\mathit{tails}](t)) \ominus \tfrac{1}{2} \\
&= \tfrac{1}{2}([\![\varphi_1]\!](h_0)) \\
&= 0
\end{aligned}$$

$$\begin{aligned}
[\![\varphi_1]\!](h_1) &= \mathrm{Next}([\mathit{tails}] \sqcup [\![\varphi_1]\!])(h_1) \ominus \tfrac{1}{2} \\
&= (\tfrac{1}{2} - \varepsilon)([\mathit{tails}] \sqcup [\![\varphi_1]\!])(h_1) + (\tfrac{1}{2} + \varepsilon)([\mathit{tails}] \sqcup [\![\varphi_1]\!])(t) \ominus \tfrac{1}{2} \\
&= (\tfrac{1}{2} - \varepsilon)([\![\varphi_1]\!](h_1)) + (\tfrac{1}{2} + \varepsilon)([\mathit{tails}](t)) \ominus \tfrac{1}{2} \\
&= (\tfrac{1}{2} - \varepsilon)([\![\varphi_1]\!](h_1)) + \varepsilon \\
&= \tfrac{\varepsilon}{0.5+\varepsilon}
\end{aligned}$$

Let $\varphi_2 = \mu V.\, \mathrm{next}(\mathit{tails} \vee V)$, then the computation of the quantitative μ-calculus formula of Example 4 is as follows,

$$\begin{aligned}
[\![\varphi_2]\!] &= [\![\mu V.\, \mathrm{next}(\mathit{tails} \vee V)]\!] \\
&= \inf\{f \in \mathcal{F} \mid f = [\![\mathrm{next}(\mathit{tails} \vee V)]\!]\} \\
&= \inf\{f \in \mathcal{F} \mid f = \mathrm{Next}([\![\mathit{tails} \vee V]\!])\} \\
&= \inf\{f \in \mathcal{F} \mid f = \mathrm{Next}([\![\mathit{tails}]\!] \sqcup [\![V]\!])\} \\
&= \inf\{f \in \mathcal{F} \mid f = \mathrm{Next}([\mathit{tails}] \sqcup f)\}
\end{aligned}$$

$$\begin{aligned}
[\![\varphi_2]\!](h_2) &= \mathrm{Next}([\mathit{tails}] \sqcup [\![\varphi_2]\!])(h_2) \\
&= 1([\mathit{tails}] \sqcup [\![\varphi_2]\!])(h_2) \\
&= [\![\varphi_2]\!](h_2) \\
&= 0
\end{aligned}$$

$$\begin{aligned}
[\![\varphi_2]\!](h_3) &= \mathrm{Next}([\mathit{tails}] \sqcup [\![\varphi_2]\!])(h_3) \\
&= (1 - \varepsilon)([\mathit{tails}] \sqcup [\![\varphi_2]\!])(h_3) + \varepsilon([\mathit{tails}] \sqcup [\![\varphi_2]\!])(t_5) \\
&= (1 - \varepsilon)([\![\varphi_2]\!](h_3)) + \varepsilon([\mathit{tails}](t_5)) \\
&= (1 - \varepsilon)([\![\varphi_2]\!](h_3)) + \varepsilon \\
&= 1
\end{aligned}$$

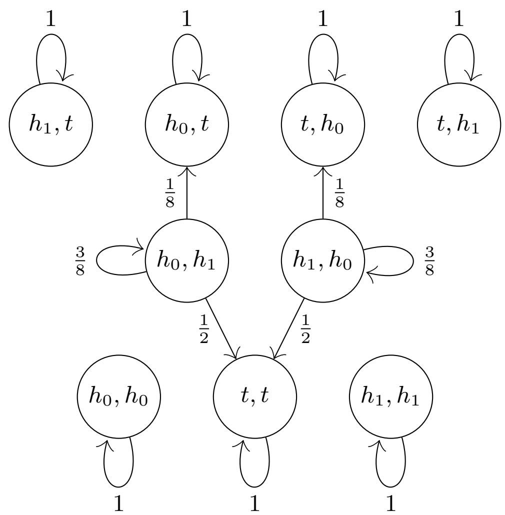


Fig. 4: The Markov chain $\langle S \times S, P\rangle$ induced by the policy P such that (h_0, h_1) reaches S_1^2 with probability $\frac{1}{5}$.

G Continuity

Proof (of Proposition 1). Let $s, t \in S$ and $\tau : S \to \mathcal{D}(S)$. It suffices to show that for each sequence $(\tau_n)_n$ converging to τ, $\liminf_n \delta_{\tau_n}(s,t) \geq \delta_\tau(s,t)$.

Let $(\tau_n)_n$ be a sequence converging to τ. Below, we prove that $\liminf_n \delta_{\tau_n}$ is a pre-fixed point of Δ_τ. Since δ_τ is the least pre-fixed point of Δ_τ by the Knaster-Tarski fixed point theorem, we can conclude that $\delta_\tau \sqsubseteq \liminf_n \delta_{\tau_n}$ and, hence, $\delta_\tau(s,t) \leq \liminf_n \delta_{\tau_n}(s,t)$.

To conclude that $\Delta_\tau(\liminf_n \delta_{\tau_n}) \sqsubseteq \liminf_n \delta_{\tau_n}$, it suffices to show that for all u, $v \in S$, $\Delta_\tau(\liminf_n \delta_{\tau_n})(u,v) \leq \liminf_n \delta_{\tau_n}(u,v)$. Let u, $v \in S$. We distinguish the following cases.

- If $(u,v) \in S_\Delta^2$ then
$$\Delta_\tau(\liminf_n \delta_{\tau_n})(u,v) = 0 = \liminf_n \Delta_{\tau_n}(\delta_{\tau_n})(u,v) = \liminf_n \delta_{\tau_n}(u,v).$$
- If $(u,v) \in S_1^2$ then
$$\Delta_\tau(\liminf_n \delta_{\tau_n})(u,v) = 1 = \liminf_n \Delta_{\tau_n}(\delta_{\tau_n})(u,v) = \liminf_n \delta_{\tau_n}(u,v).$$
- Assume that $(u,v) \in S_{0?}^2$. By Theorem 5, for each τ_n there exists an optimal policy $P_n \in \mathcal{P}_{\tau_n}$. By Proposition 11, we have that $\liminf_n P_n \in \mathcal{P}_\tau$. Hence,

$(\liminf_n P_n)(u, v) \in \Omega(\tau(u), \tau(v))$. Therefore,

$$
\begin{aligned}
&\Delta_\tau(\liminf_n \delta_{\tau_n})(u, v)\\
&= \inf_{\omega \in \Omega(\tau(u), \tau(v))} \omega \cdot \liminf_n \delta_{\tau_n}\\
&\le (\liminf_n P_n)(u, v) \cdot \liminf_n \delta_{\tau_n} \qquad [(\liminf_n P_n)(u, v) \in \Omega(\tau(u), \tau(v))]\\
&\le \liminf_n P_n(u, v) \cdot \delta_{\tau_n}\\
&= \liminf_n \Gamma_{P_n}(\delta_{\tau_n})(u, v)\\
&= \liminf_n \delta_{\tau_n}(u, v) \qquad [P_n \text{ is optimal, Proposition 12}].
\end{aligned}
$$

□

Lemma 2. *Let s, $t \in S$. If there exists a policy $P \in \mathcal{P}$ such that*

$$(s, t) \text{ reaches } S^2_\Delta \text{ with probability } 1 \text{ in } \langle S \times S, P\rangle \tag{2}$$

then the function $\delta_\cdot(s, t) : (S \to \mathcal{D}(S)) \to [0, 1]$ is upper semi-continuous at τ.

Proof. Let $\tau : S \to \mathcal{D}(S)$ and s, $t \in S$. Assume that $(\tau_n)_n$ is a sequence in $S \to \mathcal{D}(S)$ that converges to τ. It suffices to show that $\limsup_n \delta_{\tau_n}(s, t) \le \delta_\tau(s, t)$.

Assume that $P \in \mathcal{P}_\tau$ is a policy for (s, t) and (2). It follows from Theorem 4 that $\gamma_P(s, t) = 0$. Thus, by Theorem 5, $\delta_\tau(s, t) = 0$. Hence, P is optimal for (s, t) due to Definition 11. According to Proposition 10, for each $n \in \mathbb{N}$, there exists $P_n \in \mathcal{P}_{\tau_n}$ such that $d_F(P_n, P) \le 2\, d_F(\tau_n, \tau)$. Hence, $(P_n)_n$ converges to P.

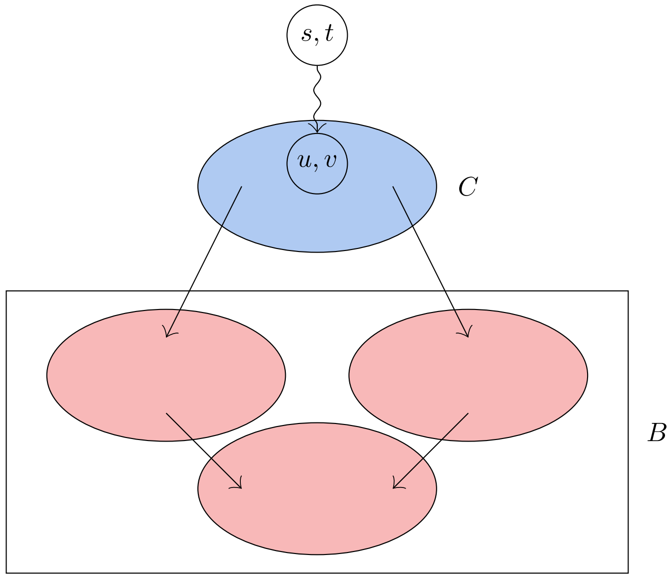
Consider the directed graph consisting of the communication classes of $\langle S \times S, P\rangle$ reachable from (s, t) as vertices. There is an edge from communication class C to communication class D if there exist $(u, v) \in C$ and $(w, x) \in D$ such that $P(u, v)(w, x) > 0$. This graph is acyclic. We first prove that for all communication classes C of $\langle S \times S, P\rangle$ that are reachable from (s, t) and for all $(u, v) \in C$,

$$\lim_n \gamma_{P_n}(u, v) = \gamma_P(u, v) \tag{3}$$

by induction on the length of a longest path from C in the communication classes graph.

In the base case, we consider the *closed* communication classes C, from which the length of a longest path in the communication classes graph is one. By (2) and Proposition 13, we only need to consider closed communication classes that are subsets of S^2_Δ. Let $C = \{(u, v)\}$ and $(u, v) \in S^2_\Delta$. According to Theorem 4, for all $n \in \mathbb{N}$, $\gamma_{P_n}(u, v) = 0$ and $\gamma_P(u, v) = 0$. Therefore, (3).

Next, we consider the inductive case. Let C be a communication class of $\langle S \times S, P\rangle$ reachable from (s, t). Let B be the set of state pairs of all communication classes that can be reached from C in the communication classes graph via a path of length greater than 1. By induction, for all $(u, v) \in B$, (3) holds. Let A be the set of all other state pairs, that is, $A = (S \times S) \setminus (B \cup C)$.



For $X \subseteq S \times S$ and $n \in \mathbb{N}$, consider the vectors

$$\gamma_{P_n}^X = (\gamma_{P_n}(u,v))_{(u,v) \in X}$$
$$\gamma_P^X = (\gamma_P(u,v))_{(u,v) \in X}$$

and the matrices

$$P_n^X = (P_n(u,v)(w,x))_{(u,v) \in C, (w,x) \in X}$$
$$P^X = (P(u,v)(w,x))_{(u,v) \in C, (w,x) \in X}$$

For all $n \in \mathbb{N}$, $\gamma_{P_n} = \Gamma_{P_n}(\gamma_{P_n})$ and, hence,

$$\gamma_{P_n}^C = P_n^C \gamma_{P_n}^C + P_n^B \gamma_{P_n}^B + P_n^A \gamma_{P_n}^A.$$

Since $\gamma_P = \Gamma_P(\gamma_P)$, we also have

$$\gamma_P^C = P^C \gamma_P^C + P^B \gamma_P^B + P^A \gamma_P^A.$$

From the communication classes graph we can infer that for all $(u,v) \in C$, support$(P(u,v)) \subseteq B \cup C$. Hence, $P^A = 0$. Since $\lim_n P_n = P$, we have that

$$\begin{aligned} \lim_n P_n^A &= P^A = 0 \\ \lim_n P_n^B &= P^B \\ \lim_n P_n^C &= P^C \end{aligned} \tag{4}$$

Next, we prove that the inverse $(I - P^C)^{-1}$ exists. We distinguish the following cases.

- If $P^C = 0$ then $I - P^C = I$, which has an inverse.
- Otherwise, $P^C \gneqq 0$. Because C is a communication class, for every (u, v), $(w, x) \in C$ there exists m such that $(P^C)^m_{(u,v),(w,x)} > 0$. By Theorem 6, P^C is irreducible. Since the communication class C is not closed, P^C is strictly substochastic. Hence, by Proposition 14, the inverse $(I - P^C)^{-1}$ exists.

Therefore,

$$
\begin{aligned}
&\gamma^C_{P_n} - \gamma^C_P \\
&= (P^C_n \gamma^C_{P_n} + P^B_n \gamma^B_{P_n} + P^A_n \gamma^A_{P_n}) - (P^C \gamma^C_P + P^B \gamma^B_P + P^A \gamma^A_P) \\
&= (P^C_n \gamma^C_{P_n} + P^B_n \gamma^B_{P_n} + P^A_n \gamma^A_{P_n}) - (P^C \gamma^C_P + P^B \gamma^B_P) \qquad [P^A = 0] \\
&= P^C(\gamma^C_{P_n} - \gamma^C_P) + (P^C_n - P^C)\gamma^C_{P_n} + P^B(\gamma^B_{P_n} - \gamma^B_P) + (P^B_n - P^B)\gamma^B_{P_n} + P^A_n \gamma^A_{P_n}
\end{aligned}
$$

Hence,

$$
(I - P^C)\,(\gamma^C_{P_n} - \gamma^C_P) = (P^C_n - P^C)\gamma^C_{P_n} + P^B(\gamma^B_{P_n} - \gamma^B_P) + (P^B_n - P^B)\gamma^B_{P_n} + P^A_n \gamma^A_{P_n}.
$$

As a consequence,

$$
\gamma^C_{P_n} - \gamma^C_P = (I - P^C)^{-1}\left((P^C_n - P^C)\gamma^C_{P_n} + P^B(\gamma^B_{P_n} - \gamma^B_P) + (P^B_n - P^B)\gamma^B_{P_n} + P^A_n \gamma^A_{P_n}\right)
$$

Hence,

$$
\begin{aligned}
&\|\gamma^C_{P_n} - \gamma^C_P\| \\
&= \|(I - P^C)^{-1}\left((P^C_n - P^C)\gamma^C_{P_n} + P^B(\gamma^B_{P_n} - \gamma^B_P) + (P^B_n - P^B)\gamma^B_{P_n} + P^A_n \gamma^A_{P_n}\right)\| \\
&\leq \|(I - P^C)^{-1}\|\,(\|P^C_n - P^C\|\,\|\gamma^C_{P_n}\| + \|P^B\|\,\|\gamma^B_{P_n} - \gamma^B_P\| + \|P^B_n - P^B\|\,\|\gamma^B_{P_n}\| \\
&\qquad + \|P^A_n\|\,\|\gamma^A_{P_n}\|) \\
&\leq \|(I - P^C)^{-1}\|\,(\|P^C_n - P^C\| + \|P^B\|\,\|\gamma^B_{P_n} - \gamma^B_P\| + \|P^B_n - P^B\| + \|P^A_n\|) \\
&\qquad [\|\gamma^X_{P_n}\| \leq 1] \\
&\leq \|(I - P^C)^{-1}\|\,(\|P^C_n - P^C\| + |S|^2\,\|\gamma^B_{P_n} - \gamma^B_P\| + \|P^B_n - P^B\| + \|P^A_n\|) \\
&\qquad [\|P^B\| \leq |S|^2]
\end{aligned}
$$

We need to prove that $\lim_n \gamma^C_{P_n} = \gamma^C_P$ and that this limit exists. It is sufficient to show that $\limsup_n \|\gamma^C_{P_n} - \gamma^C_P\| = 0$. From the above we can conclude that this holds, as

$$
\begin{aligned}
&\limsup_n \|\gamma^C_{P_n} - \gamma^C_P\| \\
&\leq \limsup_n \|(I - P^C)^{-1}\|\,(\|P^C_n - P^C\| + |S|^2\,\|\gamma^B_{P_n} - \gamma^B_P\| + \|P^B_n - P^B\| \\
&\qquad + \|P^A_n\|) \\
&= \|(I - P^C)^{-1}\|\,(\limsup_n \|P^C_n - P^C\| + |S|^2 \limsup_n \|\gamma^B_{P_n} - \gamma^B_P\| \\
&\qquad + \limsup_n \|P^B_n - P^B\| + \limsup_n \|P^A_n\|) \\
&= \|(I - P^C)^{-1}\|\,(|S|^2 \limsup_n \|\gamma^B_{P_n} - \gamma^B_P\|) \qquad [(4)] \\
&= 0 \qquad [\limsup_n \|\gamma^B_{P_n} - \gamma^B_P\| = 0 \text{ by induction}]
\end{aligned}
$$

This proves (3).

Assume that (s,t) belongs to communication class C. Then

$$\begin{aligned}
\limsup_n \delta_{\tau_n}(s,t) &\leq \limsup_n \gamma_{P_n}(s,t) && [\delta_{\tau_n} \sqsubseteq \gamma_{P_n} \text{ by Theorem 5}] \\
&= \limsup_n \gamma^C_{P_n}(s,t) && [(s,t) \in C] \\
&= \gamma^C_P(s,t) && [(3)] \\
&= \gamma_P(s,t) && [(s,t) \in C] \\
&= \delta_\tau(s,t) && [P \in \mathcal{P}_\tau \text{ is optimal for } (s,t)]
\end{aligned}$$

Hence, the function $\delta_{\cdot}(s,t)$ is upper semi-continuous at τ. □

Proof (of Theorem 2). Follows directly from Proposition 1 and Lemma 2. □

H Robust Probabilistic Bisimilarity

Theorem 7 ([40, Theorem 2.1.30]). *For all $\tau : S \to \mathcal{D}(S)$, δ_τ is a pseudometric.*

Proposition 15. *For all $\tau : S \to \mathcal{D}(S)$, $\sim$ is an equivalence relation.*

Proof. Let $\tau : S \to \mathcal{D}(S)$ and $s \in S$. By Theorem 7, $\delta_\tau(s,s) = 0$ and, hence, $s \sim s$ by Theorem 1. Let s, $t \in S$. Then

$$\begin{aligned}
s \sim t \text{ iff } & \delta_\tau(s,t) = 0 && [\text{Theorem 1}] \\
\text{iff } & \delta_\tau(t,s) = 0 && [\delta_\tau(s,t) = \delta_\tau(t,s) \text{ by Theorem 7}] \\
\text{iff } & t \sim s && [\text{Theorem 1}]
\end{aligned}$$

Let s, t, $u \in S$. Then

$$\begin{aligned}
s \sim t \text{ and } t \sim u \text{ iff } & \delta_\tau(s,t) = 0 \text{ and } \delta_\tau(t,u) = 0 && [\text{Theorem 1}] \\
\text{iff } & \delta_\tau(s,u) = 0 && [\delta_\tau(s,u) \leq \delta_\tau(s,t) + \delta_\tau(t,u) \text{ by Theorem 7}] \\
\text{iff } & s \sim u && [\text{Theorem 1}]
\end{aligned}$$

Therefore, $\sim$ is an equivalence relation.

Proposition 16. *Robust bisimilarity, $\simeq$, is a bisimulation.*

Proof. Let $\tau : S \to \mathcal{D}(S)$ and $s, t \in S$ such that $s \simeq t$. By Definition 1 we need to show that $\ell(s) = \ell(t)$, there exists $\omega \in \Omega(\tau(s), \tau(t))$ such that $\text{support}(\omega) \subseteq \simeq$, and that $\simeq$ is an equivalence relation.

Let $P_{st} \in \mathcal{P}$ be the policy such that (s,t) reaches S^2_Δ with probability 1 in $\langle S \times S, P_{st} \rangle$. Since, according to Proposition 13, all pairs of states in S^2_1 are closed communication classes, we know that $(s,t) \notin S^2_1$ and $\ell(s) = \ell(t)$.

Let $\omega = P_{st}(s,t)$, u, $v \in S$ and $(u,v) \in \text{support}(\omega)$. Hence, $\omega(u,v) > 0$ and (u,v) is reachable from (s,t). Therefore, (u,v) must reach S^2_Δ with probability 1 in $\langle S \times S, P_{st} \rangle$. Consequently, $u \simeq v$. As a result, $\text{support}(\omega) \subseteq \simeq$.

It remains to prove that $\simeq$ is an equivalence relation. Clearly $S^2_\Delta \subseteq \simeq$, thus, $\simeq$ is reflexive. We can construct P_{ts} such that for all w, x, y, $z \in S$, $P_{ts}(x,w)(z,y) = P_{st}(w,x)(y,z)$. Since (t,s) reaches S^2_Δ with probability 1 in $\langle S \times S, P_{ts}\rangle$, we have $t \simeq s$. Thus, $\simeq$ is symmetric.

Let $u \in S$ such that $t \simeq u$. Then there exists a policy $P_{tu} \in \mathcal{P}$ such that (t,u) reaches S^2_Δ with probability 1 in $\langle S \times S, P_{tu}\rangle$. To show that $\simeq$ is transitive, it suffices to show that $s \simeq u$. We define the following sets,

$$\begin{aligned}
R_{st} &= \{\, (a,b) \in S \times S \mid (a,b) \text{ is reachable from } (s,t) \text{ in } \langle S \times S, P_{st}\rangle \,\}\\
R_{tu} &= \{\, (a,b) \in S \times S \mid (a,b) \text{ is reachable from } (t,u) \text{ in } \langle S \times S, P_{tu}\rangle \,\}\\
R &= \{\, (a,c) \in S \times S \mid b_{(a,c)} \neq \varnothing \,\}, \text{ where}\\
b_{(a,c)} &= \{\, b \in S \mid (a,b) \in R_{st} \text{ and } (b,c) \in R_{tu} \,\}.
\end{aligned}$$

Let T, $U \in S \times S$. We define $T \bowtie U$ as the set $\{\, (s_1,s_3) \in S \times S \mid \exists s_2 \in S$ such that $(s_1,s_2) \in T$ and $(s_2,s_3) \in U \,\}$. With this notation, $R = R_{st} \bowtie R_{tu}$.

Let $(a,c) \in R$, we construct $\omega_{(a,c)}$ as follows. For all $(x,z) \in S \times S$, let

$$\omega_{(a,c)}(x,z) = \frac{\sum_{b \in b_{(a,c)}} \sum_{y \in S} \frac{P_{st}(a,b)(x,y) P_{tu}(b,c)(y,z)}{\tau(b)(y)}}{|b_{(a,c)}|}.$$

Then, for all $x \in S$ we have

$$\begin{aligned}
\omega_{(a,c)}(x,S) &= \sum_{z \in S} \frac{\sum_{b \in b_{(a,c)}} \sum_{y \in S} \frac{P_{st}(a,b)(x,y) P_{tu}(b,c)(y,z)}{\tau(b)(y)}}{|b_{(a,c)}|}\\
&= \frac{\sum_{b \in b_{(a,c)}} \sum_{y \in S} \frac{P_{st}(a,b)(x,y)\tau(b)(y)}{\tau(b)(y)}}{|b_{(a,c)}|} \qquad [P_{tu}(b,c) \in \Omega(\tau(b),\tau(c))]\\
&= \frac{\sum_{b \in b_{(a,c)}} \tau(a)(x)}{|b_{(a,c)}|} \qquad [P_{st}(a,b) \in \Omega(\tau(a),\tau(b))]\\
&= \tau(a)(x).
\end{aligned}$$

One can show similarly that $\omega_{(a,c)}(S,z) = \tau(c)(z)$ holds for all $z \in S$. Hence $\omega_{(a,c)} \in \Omega(\tau(a),\tau(c))$.

Claim 1. For all $(a,c) \in R$, we have

$$\text{support}(\omega_{(a,c)}) = \bigcup_{b \in b_{(a,c)}} \big(\text{support}(P_{st}(a,b)) \bowtie \text{support}(P_{tu}(b,c))\big) \subseteq R.$$

Proof (of Claim 1). Assume that $(a,c) \in R$. First, we show support$(\omega_{(a,c)}) \supseteq \bigcup_{b \in b_{(a,c)}}$ support$(P_{st}(a,b)) \bowtie$ support$(P_{tu}(b,c))$. Let $(x,z) \in$ support$(P_{st}(a,b))$ $\bowtie$ support$(P_{tu}(b,c))$ for some $b \in b_{(a,c)}$. Then there exists $y \in S$ such that $(x,y) \in$ support$(P_{st}(a,b))$ and $(y,z) \in$ support$(P_{tu}(b,c))$. By the definition of $\omega_{(a,c)}$, we have $(x,z) \in$ support$(\omega_{(a,c)})$. To show the other inclusions, let $(x,z) \in$ support$(\omega_{(a,c)})$. Then there exist $b \in b_{(a,c)}$ and $y \in S$ such that

$(x, y) \in \text{support}(P_{st}(a, b))$ and $(y, z) \in \text{support}(P_{tu}(b, c))$. Therefore, $(x, z) \in \text{support}(P_{st}(a, b)) \bowtie \text{support}(P_{tu}(b, c))$. Moreover, since $b \in b_{(a,c)}$, we have $(a, b) \in R_{st}$ and $(b, c) \in R_{tu}$. It follows that $(x, y) \in R_{st}$ and $(y, z) \in R_{tu}$. Hence, $(x, z) \in R$. Thus, $\text{support}(\omega_{(a,c)}) \subseteq R$. This proves Claim 1.

Let $P \in \mathcal{P}$ be a policy with

$$P(a, c)(x, z) = \begin{cases} P_{st}(a, c)(x, z) & \text{if } (a, c) \in R_{st} \\ P_{tu}(a, c)(x, z) & \text{if } (a, c) \in R_{tu} \setminus R_{st} \\ \omega_{(a,c)}(x, z) & \text{if } (a, c) \in R \setminus (R_{st} \cup R_{tu} \cup S^2_{\Delta}) \end{cases}$$

Claim 2. For all $(a, c) \in R_{st} \cup R_{tu}$, we have $\text{support}(P(a, c)) \subseteq R_{st} \cup R_{tu}$ and (a, c) has a path to S^2_{Δ} in $\langle S \times S, P\rangle$.

Proof (of Claim 2). For all $(a, c) \in R_{st}$, we have $\text{support}(P(a, c)) \subseteq R_{st}$ and for all $(a, c) \in R_{tu}$, we have $\text{support}(P(a, c)) \subseteq R_{st} \cup R_{tu}$. Note that, from the definition of R_{st}, for all $(a, c) \in R_{st}$, it holds that (a, c) reaches S^2_{Δ} with probability 1 in $\langle S \times S, P\rangle$. It follows that for all $(a, c) \in R_{tu}$, (a, c) reaches S^2_{Δ} with probability 1 in $\langle S \times S, P\rangle$. This proves Claim 2.

We have $(s, u) \in R_{st} \bowtie R_{tu} = R$. Towards a contradiction, assume that $s \not\simeq u$. By Claims 1 and 2, for all (x, z) reachable from (s, u) in $\langle S \times S, P\rangle$, we have $(x, z) \in R \cup R_{st} \cup R_{tu}$. According to Proposition 13, (s, u) reaches some closed communication class $C \subseteq S^2_{0,\tau} \cup S^2_1$ in $\langle S \times S, P\rangle$. By Claim 2, we have $C \cap (R_{st} \cup R_{tu}) = \varnothing$. Let $(x_1, z_1) \in C$, then $(x_1, z_1) \in R \setminus (R_{st} \cup R_{tu})$ and there exists $y_1 \in b_{(x_1,z_1)}$. We consider the case when $(x_1, y_1) \in R_{st}$ has a path of shorter or equal length to S^2_{Δ} in $\langle S \times S, P\rangle$ than $(y_1, z_1) \in R_{tu}$ has. The other case is similar. Let $(x_1, y_1), \ldots, (x_n, y_n)$ be this path in $\langle S \times S, P\rangle$, where $x_n = y_n$. Since $y_1, \ldots, y_n$ is a path in the original Markov chain $\langle S, \tau\rangle$, there is also a path $z_1, \ldots, z_n$ in $\langle S, \tau\rangle$ such that $(y_1, z_1), \ldots, (y_n, z_n)$ is a path in $\langle S \times S, P\rangle$ and $(y_i, z_i) \in R_{tu}$ for all $1 \le i \le n$. Then, by Claim 1, we have $(x_i, z_i) \in \text{support}(P(x_{i-1}, z_{i-1}))$ for all $2 \le i \le n$. Hence, there exists a path $(x_1, z_1), \ldots, (x_n, z_n)$ in $\langle S \times S, P\rangle$. Thus, we have $(x_n, z_n) \in C$ and $(x_n, z_n) = (y_n, z_n) \in R_{tu}$, contradicting the fact that $C \cap R_{tu} = \varnothing$. Therefore, $s \simeq u$. $\square$

Proof (of Proposition 3). Let $\tau : S \to \mathcal{D}(S)$ and $s, t \in S$. Assume that $s \simeq t$. According to Proposition 16, $\simeq$ is a bisimulation. By Definition 6 we need to show that there exists a policy $P \in \mathcal{P}$ such that $\simeq$ supports a path from (s, t) to S^2_{Δ} in $\langle S \times S, P\rangle$.

Let $P \in \mathcal{P}$ be the policy such that (s, t) reaches S^2_{Δ} with probability 1 in $\langle S \times S, P\rangle$. Write $s_1 = s$ and $t_1 = t$. Since (s, t) reaches S^2_{Δ} with probability 1 in $\langle S \times S, P\rangle$, there is a path $(s_1, t_1), \ldots, (s_n, t_n)$ in $\langle S \times S, P\rangle$, with $s_n = t_n$. For all $2 \le i \le n$, (s_i, t_i) is reachable from (s, t). Therefore, (s_i, t_i) must reach S^2_{Δ} with probability 1 in $\langle S \times S, P\rangle$. Consequently, $s_i \simeq t_i$. Similarly, for each $(u, v) \in \text{support}(P(s_i, t_i))$, (u, v) must reach S^2_{Δ} with probability 1 in $\langle S \times S, P\rangle$. Hence, $u \simeq v$ and, as a result, $\text{support}(P(s_i, t_i)) \subseteq \simeq$. Thus, $\simeq$ supports a path from (s, t) to S^2_{Δ} in $\langle S \times S, P\rangle$. $\square$

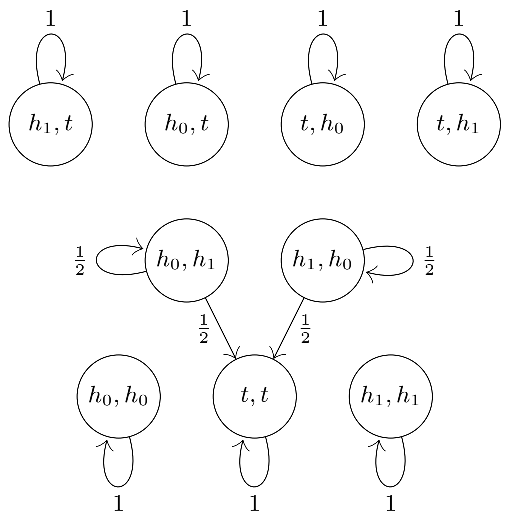


Fig. 5: The Markov chain $\langle S \times S, P\rangle$ induced by the policy P such that (h_0, h_1) reaches S^2_Δ with probability 1.

Proposition 17. *For all* μ, $\nu \in \mathcal{S}(X)$, *given an equivalence relation* $R \subseteq X \times X$ *such that for all* R*-equivalence classes* A, $\mu(A) = \nu(A)$, *one can compute in* $\mathcal{O}(|R|)$ *time a coupling* $\omega \in \Omega(\mu, \nu)$ *such that* $\mathrm{support}(\omega) \subseteq R$.

Proof. Let μ, $\nu \in \mathcal{S}(X)$ and $R \subseteq X \times X$ be an equivalence relation such that for all R-equivalence classes A, $\mu(A) = \nu(A)$. Let $A \subseteq X$ be an R-equivalence class and μ_A, $\nu_A \in \mathcal{S}(A)$ such that

$$\mu_A(x) = \begin{cases} \mu(x) & \text{if } x \in A \\ 0 & \text{otherwise} \end{cases}$$

$$\nu_A(x) = \begin{cases} \nu(x) & \text{if } x \in A \\ 0 & \text{otherwise.} \end{cases}$$

Since $\mu(A) = \nu(A)$ and, thus, $\mu_A(X) = \nu_A(X)$, we know that there exists a coupling of μ_A and ν_A [19, Lemma 1]. The North-West corner method [23] constructs a coupling $\omega_A \in \Omega(\mu_A, \nu_A)$ in $\mathcal{O}(|A|)$ time. Note that $\mathrm{support}(\omega_A) \subseteq A \times A \subseteq R$.

Let ω be the sum of ω_A over all R-equivalence classes A. Let $x \in X$ and $B \subseteq X$ be the R-equivalence class such that $x \in B$. Then for all $y \in X$, we have $\omega(x, y) = \omega_B(x, y)$. Therefore, $\omega \in \Omega(\mu, \nu)$ such that $\mathrm{support}(\omega) \subseteq R$. □

For μ, $\nu \in \mathcal{S}(S)$ and an equivalence relation $R \subseteq S \times S$ such that for all R-equivalence classes A, $\mu(A) = \nu(A)$, we say that $\omega \in \Omega(\mu, \nu)$ is a *maximal* R*-support coupling* if $\mathrm{support}(\omega) = (\mathrm{support}(\mu) \times \mathrm{support}(\nu)) \cap R$. Moreover, we say that $P \in \mathcal{P}$ is a *maximal* R*-support policy* if for all $(s, t) \in R \cap S^2_{0?}$, the coupling $P(s,t)$ is a maximal R-support coupling, that is, $\text{support}(P(s,t)) = \text{Post}((s,t)) \cap R$.

Proposition 18. *For all μ, $\nu \in \mathcal{S}(X)$, given an equivalence relation $R \subseteq X \times X$ such that for all R-equivalence classes A, $\mu(A) = \nu(A)$, there exists a maximal R-support coupling $\omega \in \Omega(\mu, \nu)$.*

Proof. Let μ, $\nu \in \mathcal{S}(X)$ and $R \subseteq X \times X$ be an equivalence relation such that for all R-equivalence classes A, $\mu(A) = \nu(A)$. For each $x \in X$, let $L_x^{(1)}$ be the set $\{s \in X \mid (x,s) \in (\text{support}(\mu) \times \text{support}(\nu)) \cap R\}$ and $L_x^{(2)}$ be the set $\{s \in X \mid (s,x) \in (\text{support}(\mu) \times \text{support}(\nu)) \cap R\}$. We assign ω_1, μ' and ν' as follows:

1 $\mu' \leftarrow \mu$
2 $\nu' \leftarrow \nu$
3 **for each** $(u,v) \in (\text{support}(\mu) \times \text{support}(\nu)) \cap R$
4 $\quad p \leftarrow \min\left(\dfrac{\mu(u)}{|L_u^{(1)}|}, \dfrac{\nu(v)}{|L_v^{(2)}|}\right)$
5 $\quad \omega_1(u,v) \leftarrow p$
6 $\quad \mu'(u) \leftarrow \mu'(u) - p$
7 $\quad \nu'(v) \leftarrow \nu'(v) - p$

Initially, for all R-equivalence classes A, $\mu'(A) = \nu'(A)$. In each iteration of the loop above, $(u,v) \in R$, and therefore lines 6 and 7 preserve this property. At the end we have $\text{support}(\omega_1) = (\text{support}(\mu) \times \text{support}(\nu)) \cap R$. We can then construct a coupling $\omega_2 \in \Omega(\mu', \nu')$ with $\text{support}(\omega_2) \subseteq R$ as described in Proposition 17. Define $\omega = \omega_1 + \omega_2$. Observe that $\omega \in \Omega(\mu, \nu)$ and $\text{support}(\omega) = (\text{support}(\mu) \times \text{support}(\nu)) \cap R$. ◻

Proposition 19. *For any bisimulation $R \subseteq S \times S$, a maximal R-support policy $P \in \mathcal{P}$ exists.*

Proof. Let $R \subseteq S \times S$ be a bisimulation. Define $P \in \mathcal{P}$ to be a policy such that for all $(s,t) \in R \cap S^2_{0?}$, $P(s,t) \in \Omega(\tau(s), \tau(t))$ is a maximal R-support coupling. Such a policy P exists by Proposition 18. It is a maximal R-support policy. ◻

Proof (of Proposition 4). Let $R \subseteq S \times S$ be a robust bisimulation and $P \in \mathcal{P}$ be a maximal R-support policy. Let $(s,t) \in R$. Then there exists a policy $P_{st} \in \mathcal{P}$ such that R supports a path from (s,t) to S^2_Δ in $\langle S \times S, P_{st} \rangle$. Thus, R supports the same path in $\langle S \times S, P \rangle$ as well. Therefore, for every $(s,t) \in R$, R supports a path from (s,t) to S^2_Δ in $\langle S \times S, P \rangle$.

Let s, $t \in S$. Assume that $(s,t) \in R$. According to, for example, [2, Theorem 10.27], (s,t) almost surely reaches a closed communication class in $\langle S \times S, P \rangle$. By Proposition 13, a closed communication class, say C, is a subset of S^2_1, $S^2_{0,\tau}$, or S^2_Δ. Since $\text{support}(P(u,v)) \subseteq R$, for all $(u,v) \in R$, (s,t) cannot leave R. By the definition of robust bisimilarity, we know that $S^2_1 \cap \simeq = \varnothing$, thus, $S^2_1 \cap R = \varnothing$. Observe that for all closed communication classes C of $\langle S \times S, P \rangle$ with $C \subseteq S^2_{0,\tau}$, we have $C \cap R = \varnothing$, as each $(s,t) \in R$ reaches S^2_Δ in $\langle S \times S, P \rangle$. Therefore, we can conclude that (s,t) reaches S^2_Δ with probability 1 in $\langle S \times S, P \rangle$. ◻

I Algorithm

Proof (of Proposition 5). Clearly, Bisim is monotone with respect to $\subseteq$.

Let A, $B \subseteq S \times S$, with $A \subseteq B$. Then for all $(s,t) \in \text{Filter}(A)$ there exists a policy $P \in \mathcal{P}$ such that A supports a path from (s,t) to S^2_Δ in $\langle S \times S, P\rangle$. Since $A \subseteq B$, B supports the same path from (s,t) to S^2_Δ in $\langle S \times S, P\rangle$. Thus, $(s,t) \in \text{Filter}(B)$.

Let $s, t, u \in S$. Let $A = \{(s,s), (t,t), (u,u), (s,t), (t,s)\}$ and $B = \{(s,s), (t,t), (u,u), (s,t), (t,s), (t,u), (u,t)\}$. A and B are symmetric and reflexive and, thus, can be visualized as an undirected graph as shown in Figure 2. Observe that $A \subseteq B$, however, $\text{Prune}(A) = \{(s,s), (t,t), (u,u), (s,t), (t,s)\} \not\subseteq \text{Prune}(B) = \{(s,s), (t,t), (u,u)\}$. Thus, Prune is not monotone. □

Proof (of Proposition 6). Let $R \subseteq S \times S$. We prove the two implications.

Assume that R is a robust bisimulation. It is sufficient to show that R is a fixed point of Filter, Prune and Bisim. By the definitions of the functions, $\text{Filter}(R) \subseteq R$, $\text{Prune}(R) \subseteq R$ and $\text{Bisim}(R) \subseteq R$. Since R is a robust bisimulation, $R \subseteq \text{Filter}(R)$, $R \subseteq \text{Prune}(R)$ and $R \subseteq \text{Bisim}(R)$.

Assume that R is a fixed point of Refine, thus $R = \text{Refine}(R)$. It follows that $R = \text{Bisim}(R)$ and $R = \text{Filter}(R)$. Then, R is a bisimulation and for every $(s,t) \in R$ there exists a policy $P \in \mathcal{P}$ such that R supports a path from (s,t) to S^2_Δ in $\langle S \times S, P\rangle$. Therefore, R is a robust bisimulation. □

Proposition 20. *For all $R \in [S^2_\Delta, \sim]_\mathcal{B}$ and for all maximal R-support policies $P \in \mathcal{P}$, we have* $\text{Filter}(R) = \{\, (s,t) \in R \mid \exists$ *path from* (s,t) *to* S^2_Δ *in* $\langle S \times S, P\rangle \,\}$.

Proof. Let $R \in [S^2_\Delta, \sim]_\mathcal{B}$ and $(s_1,t_1) \in R$. Let $P \in \mathcal{P}$ be a maximal R-support policy. We prove the two inclusions.

Assume that $(s_1,t_1) \in \text{Filter}(R)$. Then there exists a policy $P_1 \in \mathcal{P}$ such that R supports a path $(s_1,t_1), \ldots, (s_n,t_n)$ in $\langle S \times S, P_1\rangle$, where $s_n = t_n$. Hence, for all $1 \le i \le n$ we have $(s_i,t_i) \in R$. Thus, the same path $(s_1,t_1), \ldots, (s_n,t_n)$ exists in $\langle S \times S, P\rangle$.

Assume that there exists a path from (s_1,t_1) to S^2_Δ in $\langle S \times S, P\rangle$. Below we prove, by induction, that for all $1 \le i \le n$ we have $(s_i,t_i) \in R$ and $\text{support}(P(s_i,t_i)) \subseteq R$. Therefore, R supports the same path $(s_1,t_1), \ldots, (s_n,t_n)$ in $\langle S \times S, P\rangle$. Thus, $(s_1,t_1) \in \text{Filter}(R)$.

In the base case, we know that $(s_1,t_1) \in R$ and $\text{support}(P(s_1,t_1)) \subseteq R$, as P is a maximal R-support policy. In the inductive case, assume that $(s_i,t_i) \in R$ and $\text{support}(P(s_i,t_i)) \subseteq R$. Since $(s_{i+1},t_{i+1}) \in \text{support}(P(s_i,t_i))$, we have $(s_{i+1},t_{i+1}) \in R$. Hence, $\text{support}(P(s_{i+1},t_{i+1})) \subseteq R$. □

In the following we use implicitly the characterization of Filter from Proposition 20.

Proposition 21. *For all $R \in [S^2_\Delta, \sim]_\mathcal{B}$,* $\text{Filter}(R)$ *is symmetric and reflexive.*

Proof. Let $R \in [S^2_\Delta, \sim]_\mathcal{B}$. By the definition of the function, $S^2_\Delta \subseteq \mathrm{Filter}(R)$, hence $\mathrm{Filter}(R)$ is reflexive. Let s_1, $t_1 \in S$ and $P \in \mathcal{P}$ be a maximal R-support policy. Assume that $(s_1, t_1) \in \mathrm{Filter}(R)$. Then there exists a path $(s_1, t_1), \ldots, (s_n, t_n)$ in $\langle S \times S, P \rangle$, where $s_n = t_n$. By the definition of a maximal R-support policy, for all $1 \leq i \leq n$ we have $(s_i, t_i) \in R$. Since R is an equivalence relation, for all $1 \leq i \leq n$ we have $(t_i, s_i) \in R$. Thus, there exists a path $(t_1, s_1), \ldots, (t_n, s_n)$ in $\langle S \times S, P \rangle$, where $t_n = s_n$. Since P is a maximal R-support policy, it follows from Proposition 20 that $(t, s) \in \mathrm{Filter}(R)$. Hence, $\mathrm{Filter}(R)$ is symmetric. □

Proposition 22. *For all $R \in [S^2_\Delta, \sim]$ such that R is symmetric and reflexive,* $\mathrm{Prune}(R)$ *is an equivalence relation.*

Proof. Let $R \in [S^2_\Delta, \sim]$ be symmetric and reflexive. It is sufficient to show that $\mathrm{Prune}(R)$ is reflexive, symmetric and transitive.

By the definition of the function, $S^2_\Delta \subseteq \mathrm{Prune}(R)$, hence $\mathrm{Prune}(R)$ is reflexive.

Let s, t, $u \in S$. Assume that $(s, t) \in \mathrm{Prune}(R)$, then $\forall (t, u) \in R : (s, u) \in R$ and $\forall (u, s) \in R : (u, t) \in R$. Since R is symmetric, $\forall (u, t) \in R : (u, s) \in R$ and $\forall (s, u) \in R : (t, u) \in R$. Thus, $(t, s) \in \mathrm{Prune}(R)$ and $\mathrm{Prune}(R)$ is symmetric.

Lastly, assume that (s, t), $(t, u) \in \mathrm{Prune}(R)$. Then $(s, t) \in R$, $(t, u) \in R$ and we have that $\forall (t, x) \in R : (s, x) \in R$, $\forall (x, s) \in R : (x, t) \in R$, $\forall (u, x) \in R : (t, x) \in R$, and $\forall (x, t) \in R : (x, u) \in R$. Therefore, $(s, u) \in R$, $\forall (u, x) \in R : (s, x) \in R$, and $\forall (x, s) \in R : (x, u) \in R$. Thus, $(s, u) \in \mathrm{Prune}(R)$. Hence, $\mathrm{Prune}(R)$ is transitive. □

Proposition 23. *Given an equivalence relation $E \in [S^2_\Delta, \sim]$,* $\mathrm{Bisim}(E)$ *can be computed in polynomial time.*

Proof. Let $E \in [S^2_\Delta, \sim]$ be an equivalence relation. The largest bisimulation $E' \subseteq E$ exists, since the transitive closure of the union of all bisimulations $E' \subseteq E$, is also $\subseteq E$. $\mathrm{Bisim}(E)$ can be computed in polynomial time, for example, by using the partition refinement algorithm by Derisavi et al. [13]. The algorithm takes as input an equivalence relation E and returns the largest equivalence relation $E' \subseteq E$ such that for all $(s, t) \in E'$ and for all E'-equivalence classes C, we have $\tau(s)(C) = \tau(t)(C)$. This is done by selecting a single equivalence class X from the current partition at each iteration and then refining the partition by comparing $\tau(s)(X)$ for each $s \in S$, until a fixed point is reached. Since for all $(s, t) \in E$, $\ell(s) = \ell(t)$, it follows that E' is the largest bisimulation $\subseteq E$. □

Proposition 24. *Algorithm 3 computes* Filter*.*

Proof. Let $R \in [S^2_\Delta, \sim]_\mathcal{B}$ and $P \in \mathcal{P}$ be a maximal R-support policy. We show the following loop invariant of Algorithm 3: $Q = \{\, (s, t) \in R \mid \exists \text{ path of length} \leq n \text{ from } (s, t) \text{ to } S^2_\Delta \text{ in } \langle S \times S, P \rangle \,\}$.

Q is initialized to S^2_Δ. For all $(s, t) \in S^2_\Delta$, (s, t) can reach S^2_Δ in 0 steps in $\langle S \times S, P \rangle$. Hence, the loop invariant holds before the loop.

Assume that the loop invariant holds before an iteration of the loop, that is, $Q = Q_{\text{old}} = \{ (s,t) \in R \mid \exists$ path of length $\leq n$ from (s,t) to S^2_Δ in $\langle S \times S, P \rangle \}$. Let s, $t \in S$ and $(s,t) \in R$. We need to show that (s,t) is added to Q on line 6 if and only if there exists a shortest path of length $n+1$ from (s,t) to S^2_Δ in $\langle S \times S, P \rangle$. We prove the two implications.

Assume that (s,t) is added to Q. Then $(s,t) \in (R \cap \text{Pre}(Q_{\text{old}})) \setminus Q_{\text{old}}$. Thus, there is $(u,v) \in \text{Post}((s,t)) \cap Q_{\text{old}}$. Since P is a maximal R-support policy and $Q_{\text{old}} \subseteq R$, we have $(u,v) \in \text{support}(P(s,t)) \cap Q_{\text{old}}$. By the induction hypothesis and since $(s,t) \notin Q_{\text{old}}$, there exists a shortest path of length n from (u,v) to S^2_Δ in $\langle S \times S, P \rangle$. Therefore, there exists a shortest path of length $n+1$ from (s,t) to S^2_Δ in $\langle S \times S, P \rangle$.

To prove the other implication, assume that there exists a shortest path of length $n+1$ from (s,t) to S^2_Δ in $\langle S \times S, P \rangle$. Let u, $v \in S$ and $(u,v) \in R$ such that the path is $(s,t), (u,v), \ldots, S^2_\Delta$. Then there exists a shortest path of length n from (u,v) to S^2_Δ in $\langle S \times S, P \rangle$. Hence, by the induction hypothesis, $(u,v) \in Q_{\text{old}}$ and $(s,t) \in (R \cap \text{Pre}(Q_{\text{old}})) \setminus Q_{\text{old}}$. Therefore, (s,t) is added to Q.

Hence, $Q = \{ (s,t) \in R \mid \exists$ path of length $\leq n+1$ from (s,t) to S^2_Δ in $\langle S \times S, P \rangle \}$ and, thus, the loop invariant is maintained in each iteration of the loop.

The loop terminates when a fixed point is reached, therefore, by the loop invariant we know that there are no pairs of states $(s,t) \in R \setminus Q$ such that (s,t) can reach S^2_Δ in $\langle S \times S, P \rangle$ with a shortest path of length n. It follows that there are no pairs of states $(s,t) \in R \setminus Q$ such that (s,t) can reach S^2_Δ in $\langle S \times S, P \rangle$ with a shortest path of length $\geq n$. Hence, $Q = \{ (s,t) \in R \mid \exists$ path from (s,t) to S^2_Δ in $\langle S \times S, P \rangle \}$. ☐

Proposition 25. *Algorithm 4 computes* Prune.

Proof. Let $Q \in [S^2_\Delta, \sim]$. Let s, $t \in S$ and $(s,t) \in Q$. Initially, $E = Q$. Observe that (s,t) is removed from E on line 5 if and only if there exists $u \in S$ such that $(t,u) \in Q \wedge (s,u) \notin Q$ or $(u,s) \in Q \wedge (u,t) \notin Q$. Therefore, $E = \text{Prune}(Q)$. ☐

Proof (of Proposition 7). Let $R \in [\simeq, \sim]_{\mathcal{B}}$ and s, t, $u \in S$ with (s,t), $(t,u) \in \text{Filter}(R)$. We show that if $t \simeq u$ then $(s,u) \in \text{Filter}(R)$. The case $s \simeq t$ is similar.

Since $\text{Filter}(R) \subseteq R$, we have (s,t), $(t,u) \in R$. Since R is an equivalence relation, $(s,u) \in R$. Write $s_1 = s$ and $t_1 = t$ and $u_1 = u$. Let $P \in \mathcal{P}$ be a maximal R-support policy. Since $(s,t) \in \text{Filter}(R)$, there exists a path $(s_1,t_1), \ldots, (s_n,t_n)$ in $\langle S \times S, P \rangle$, where $s_n = t_n$.

Assume that $(t,u) \in\ \simeq$. By Proposition 3, $\simeq$ is a bisimulation. Since $t_1, \ldots, t_n$ is a path in the original Markov chain $\langle S, \tau \rangle$, there is also a path $u_1, \ldots, u_n$ in $\langle S, \tau \rangle$ such that $(t_i, u_i) \in\ \simeq$ for all $1 \leq i \leq n$. In particular, $(t_n, u_n) \in\ \simeq$. Since $\simeq\ \subseteq R$, there exists a path $(t_1,u_1), \ldots, (t_n,u_n)$ in $\langle S \times S, P \rangle$. Note that $(s_i,u_i) \in R$ for all $1 \leq i \leq n$. Hence, there exists a path $(s_1,u_1), \ldots, (s_n,u_n) = (t_n,u_n)$ in $\langle S \times S, P \rangle$. See Figure 3.

Since $(t_n,u_n) \in\ \simeq$, there exists a policy $P' \in \mathcal{P}$ such that (t_n,u_n) reaches S^2_Δ with probability 1. Therefore, there is a path $(t_n,u_n), \ldots, (t_m,u_m)$ in $\langle S \times S, P' \rangle$, with $t_m = u_m$ and $(t_i, u_i) \in \simeq$ for all $n \leq i \leq m$. Since $\simeq \subseteq R$, the same path $(t_n, u_n), \ldots, (t_m, u_m)$ exists in $\langle S \times S, P \rangle$, with $t_m = u_m$. Thus, there exists paths $(s_1, u_1), \ldots, (s_n, u_n)$ and $(t_n, u_n), \ldots, (t_m, u_m)$, with $s_n = t_n$ and $t_m = u_m$, in $\langle S \times S, P \rangle$. Hence, $(s, u) \in \text{Filter}(R)$. □

Proof (of Proposition 8). R is initialized to $\sim$, hence, by Proposition 15 and the definition of $\sim$, the loop invariant holds before the loop.

Assume that the loop invariant holds before an iteration of the loop, that is $R \in [\simeq, \sim]_{\mathcal{B}}$. Since $\simeq \subseteq R$ and, by Proposition 5, Filter is monotone, we have $\text{Filter}(\simeq) \subseteq \text{Filter}(R)$. According to Proposition 6, $\simeq$ is a fixed point of Refine. It follows that $\simeq = \text{Filter}(\simeq) \subseteq \text{Filter}(R)$. Next we show that $\simeq \subseteq \text{Prune}(\text{Filter}(R))$. Let $s, t \in S$ and $s \simeq t$. Thus, $(s, t) \in \text{Filter}(R)$. Then, by Proposition 7, for all $(t, u) \in \text{Filter}(R)$ we have $(s, u) \in \text{Filter}(R)$ and for all $(u, s) \in \text{Filter}(R)$ we have $(u, t) \in \text{Filter}(R)$. Hence, $(s, t) \in \text{Prune}(\text{Filter}(R))$, and we have shown $\simeq \subseteq \text{Prune}(\text{Filter}(R))$. Since, by Proposition 5, Bisim is monotone, we have $\simeq = \text{Bisim}(\simeq) \subseteq \text{Bisim}(\text{Prune}(\text{Filter}(R))) = \text{Refine}(R)$. By the definition of Bisim, Refine(R) is a bisimulation, that is, $\text{Refine}(R) \in [\simeq, \sim]_{\mathcal{B}}$. Thus, the loop invariant is maintained in each iteration of the loop. □

Proof (of Theorem 3). The loop on lines 1-5 in Algorithm 1 can be rewritten as follows:

$R \leftarrow \sim$
while $\text{Refine}(R) \subsetneq R$
    $R \leftarrow \text{Refine}(R)$

It is immediate from the definitions of Bisim, Filter and Prune that $\text{Refine}(R) \subseteq R$ holds for all $R \subseteq S \times S$. Therefore, Algorithm 1 is a standard fixed-point iteration. By Proposition 8, $\simeq \subseteq R$, thus, it computes a fixed point of Refine greater than or equal to $\simeq$. Since $\simeq$ is the greatest fixed point of Refine, we can conclude that Algorithm 1 computes $\simeq$. □

Proposition 26. *Algorithm 1 runs in $\mathcal{O}(n^6)$ time, where $n = |S|$.*

Proof. Let $|S| = n$. Refine begins with $\sim$, containing at most n^2 pairs of states. Since at least one pair is removed in each iteration, Refine requires at most n^2 iterations.

Filter checks at most $|R| \leq n^2$ pairs of states per iteration and adds at least one pair of states to Q. Thus, there are at most n^2 iterations, with a total runtime of $\mathcal{O}(n^4)$.

Prune has an outer loop over $|Q| \leq n^2$ pairs of states and an inner loop over at most $|S| = n$ pairs of states. Hence, Prune runs in $\mathcal{O}(n^3)$ time.

Bisim runs in $\mathcal{O}(m \log n) = \mathcal{O}(n^2 \log n)$ time [13].

Therefore, the overall runtime of Refine is $\mathcal{O}(n^6)$. □

J Experiments

Below is a description of jpf-probabilistic's randomized algorithms utilized in our experiments.

- Erdös-Rényi Model: a model for generating a random (directed or undirected) graph. A graph with a given number of vertices v is constructed by placing an edge between each pair of vertices with a given probability p, independent from every other edge. We check the probability that the generated graph is connected (for every pair of nodes, there is a path). [17]
- Fair Baised Coin: makes a fair coin from a biased coin, where p denotes the probability by which the biased coin tosses heads. We check the probability that the coin toss results in heads. [35]
- Has Majority Element: a Monte Carlo algorithm that determines whether an integer array has a majority element (appears more than half of the time in the array). The parameter s denotes the size of the given array, t denotes the number of trials, and m denotes the amount of times that the majority element occurs in the array. We check the probability that the algorithm erroneously reports that the array does not have a majority element. [34]
- Pollards Integer Factorization: finds a factor of an integer i. We check the probability that the algorithm returns i, when i is not prime. [37]
- Queens: attempts to place a queen on each row of an $n \times n$ chess board such that no queen can attack another. We check the probability of success. [3]
- Set Isolation: finds a sample of the universe U that is disjoint from the subset S but not disjoint from the subset T. Let u denote the size of the universe and st denote the size of S and T. We check the probability that the randomly selected sample is good, that is, disjoint from S and intersects T. [26]

References

1. de Alfaro, L., Majumdar, R.: Quantitative solution of omega-regular games. In: Vitter, J.S., Spirakis, P.G., Yannakakis, M. (eds.) Proceedings of the 33rd Annual Symposium on Theory of Computing. pp. 675–683. ACM, Heraklion, Crete, Greece (Jul 2001)
2. Baier, C., Katoen, J.P.: Principles of model checking. The MIT Press, Cambridge, MA, USA (2008)
3. Barringer, H.: Randomized algorithms - a brief introduction (2010), lecture at the University of Manchester
4. Berman, A., Plemmons, R.: Nonnegative matrices in the mathematical sciences. SIAM (1994)
5. van Breugel, F., Worrell, J.: Towards quantitative verification of probabilistic transition systems. In: Orejas, F., Spirakis, P.G., van Leeuwen, J. (eds.) Proceedings of the 28th International Colloquium on Automata, Languages and Programming. Lecture Notes in Computer Science, vol. 2076, pp. 421–432. Springer, Crete, Greece (Jul 2001)
6. Cai, X., Gu, Y.: Measuring anonymity. In: Bao, F., Li, H., Wang, G. (eds.) Proceedings of the 5th International Conference on Information Security Practice and Experience. Lecture Notes in Computer Science, vol. 5451, pp. 183–194. Springer, Xi'an, China (Apr 2009)
7. Chatterjee, K., de Alfaro, L., Majumdar, R., Raman, V.: Algorithms for game metrics (full version). Logical Methods in Computer Science **6**(3) (2010)
8. Chen, D., van Breugel, F., Worrell, J.: On the complexity of computing probabilistic bisimilarity. In: Birkedal, L. (ed.) Proceedings of the 15th International Conference on Foundations of Software Science and Computational Structures. Lecture Notes in Computer Science, vol. 7213, pp. 437–451. Springer-Verlag, Tallinn, Estonia (Mar/Apr 2012)
9. Çınlar, E.: Probability and stochastics, Graduate Texts in Mathematics, vol. 261. Springer-Verlag, New York, NY, US (2011)
10. Comanici, G., Precup, D.: Basis function discovery using spectral clustering and bisimulation metrics. In: Burgard, W., Roth, D. (eds.) Proceedings of the 25th AAAI Conference on Artificial Intelligence. pp. 325–330. AAAI Press, San Francisco, California, USA (Aug 2011)
11. Davey, B., Priestley, H.: Introduction to lattices and order. Cambridge University Press, Cambridge, United Kingdom (2002)
12. Derisavi, S.: Signature-based symbolic algorithm for optimal Markov chain lumping. In: Proceedings of the 4th International Conference on the Quantitative Evaluation of Systems. pp. 141–150. IEEE Computer Society, Edinburgh, Scotland, UK (Sep 2007)
13. Derisavi, S., Hermanns, H., Sanders, W.H.: Optimal state-space lumping in Markov chains. Information Processing Letters **87**(6), 309–315 (2003)
14. Desharnais, J., Gupta, V., Jagadeesan, R., Panangaden, P.: Metrics for labeled Markov systems. In: Baeten, J.C.M., Mauw, S. (eds.) Proceedings of the 10th International Conference on Concurrency Theory. Lecture Notes in Computer Science, vol. 1664, pp. 258–273. Springer-Verlag, Eindhoven, The Netherlands (Aug 1999)
15. Desharnais, J., Gupta, V., Jagadeesan, R., Panangaden, P.: Metrics for labelled Markov processes. Theoretical Computer Science **318**(3), 323–354 (Jun 2004)

16. Eastman, J.R., He, J.: A regression-based procedure for Markov transition probability estimation in land change modeling. Land **9**(11) (Oct 2020)
17. Erdös, P., Rényi, A.: On random graphs I. Publicationes Mathematicae **6**, 290–297 (1959)
18. Fatmi, S.Z., Chen, X., Dhamija, Y., Wildes, M., Tang, Q., van Breugel, F.: Probabilistic model checking of randomized Java code. In: Laarman, A., Sokolova, A. (eds.) Proceedings of the 27th International Symposium on Model Checking Software, SPIN. Lecture Notes in Computer Science, vol. 12864, pp. 157–174. Springer (Jul 2021)
19. Fulkerson, D.R.: Hitchcock transportation problem. Rand Corporation (1956)
20. Giacalone, A., Jou, C., Smolka, S.A.: Algebraic reasoning for probabilistic concurrent systems. In: Broy, M., Jones, C.B. (eds.) Proceedings of the Working Conference on Programming Concepts and Methods. pp. 443–458. North-Holland, Sea of Galilee, Israel (Apr 1990)
21. Hartmanns, A., Klauck, M., Parker, D., Quatmann, T., Ruijters, E.: The quantitative verification benchmark set. In: Vojnar, T., Zhang, L. (eds.) Proceedings of the 25th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Lecture Notes in Computer Science, vol. 11427, pp. 344–350. Springer, Prague, Czech Republic (Apr 2019)
22. Hensel, C., Junges, S., Katoen, J., Quatmann, T., Volk, M.: The probabilistic model checker storm. International Journal on Software Tools for Technology Transfer **24**(4), 589–610 (2022)
23. Hitchcock, F.: The distribution of a product from several sources to numerous localities. Studies in Applied Mathematics **20**(1/4), 224–230 (Apr 1941)
24. Jaeger, M., Mao, H., Larsen, K.G., Mardare, R.: Continuity properties of distances for Markov processes. In: Norman, G., Sanders, W.H. (eds.) Proceedings of the 11th International Conference on Quantitative Evaluation of Systems. Lecture Notes in Computer Science, vol. 8657, pp. 297–312. Springer-Verlag, Florence, Italy (Sep 2014)
25. Jonsson, B., Larsen, K.: Specification and refinement of probabilistic processes. In: Proceedings of the 6th Annual Symposium on Logic in Computer Science. pp. 266–277. IEEE, Amsterdam, The Netherlands (Jul 1991)
26. Karger, D.R., Motwani, R.: Derandomization through approximation: An NC algorithm for minimum cuts. In: Proceedings of the 26th Annual ACM Symposium on Theory of Computing. pp. 497–506. ACM, New York, NY, USA (May 1994)
27. Katoen, J., Kemna, T., Zapreev, I.S., Jansen, D.N.: Bisimulation minimisation mostly speeds up probabilistic model checking. In: Proceedings of the 13th International Conference on Tools and Algorithms for the Construction and Analysis of Systems. Lecture Notes in Computer Science, vol. 4424, pp. 87–101. Springer (2007)
28. Kemeny, J.G., Snell, J.L.: Finite Markov chains. Springer-Verlag, Heidelberg, Germany (1960)
29. Kozen, D.: A probabilistic PDL. In: Johnson, D.S., Fagin, R., Fredman, M.L., Harel, D., Karp, R.M., Lynch, N.A., Papadimitriou, C.H., Rivest, R.L., Ruzzo, W.L., Seiferas, J.I. (eds.) Proceedings of the 15th Annual Symposium on Theory of Computing. pp. 291–297. ACM, Boston, Massachusetts, USA (Apr 1983)
30. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: Verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) Proceedings of the 23rd International Conference on Computer Aided Verification. Lecture Notes in Computer Science, vol. 6806, pp. 585–591. Springer-Verlag, Snowbird, Utah, USA (Jul 2011)

31. Larsen, K., Skou, A.: Bisimulation through probabilistic testing. In: Proceedings of the 16th Annual ACM Symposium on Principles of Programming Languages. pp. 344–352. ACM, Austin, TX, USA (Jan 1989)
32. McIver, A., Morgan, C.: Abstraction, Refinement and Proof for Probabilistic Systems. Monographs in Computer Science, Springer (2004)
33. Mizutani, D., Lethanh, N., Adey, B.T., Kaito, K.: Improving the estimation of Markov transition probabilities using mechanistic-empirical models. Frontiers in Built Environment **3**, 58 (Oct 2017)
34. Motwani, R., Raghavan, P.: Randomized algorithms. Cambridge University Press, New York, NY, USA (1995)
35. von Neumann, J.: Various techniques used in connection with random digits. In: Householder, A.S., Forsythe, G.E., Germond, H.H. (eds.) Monte Carlo Method. National Bureau of Standards Applied Mathematics Series, vol. 12, pp. 36–38. US Government Printing Office, Washington, DC (1951)
36. Olariu, E., Cadwell, K.K., Hancock, E., Trueman, D., Chevrou-Severac, H.: Current recommendations on the estimation of transition probabilities in Markov cohort models for use in health care decision-making: a targeted literature review. ClinicoEconomics and Outcomes Research **9**, 537–546 (Sep 2017)
37. Pollard, J.M.: A Monte Carlo method for factorization. BIT Numerical Mathematics **15**(3), 331–334 (Sep 1975)
38. Spitzer, F.: Principles of random walk. Graduate Texts in Mathematics, Springer-Verlag, New York, NY, USA (1964)
39. Srivastava, T., Latimer, N.R., Tappenden, P.: Estimation of transition probabilities for state-transition models: A review of NICE appraisals. PharmacoEconomics **39**(8), 869–878 (Aug 2021)
40. Tang, Q.: Computing probabilistic bisimilarity distances. Ph.D. thesis, York University, Toronto, Canada (Aug 2018)
41. Tang, Q., van Breugel, F.: Algorithms to compute probabilistic bisimilarity distances for labelled Markov chains. In: Meyer, R., Nestmann, U. (eds.) Proceedings of the 28th International Conference on Concurrency Theory. LIPIcs, vol. 85, pp. 27:1–27:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, Berlin, Germany (Sep 2017)
42. Thorsley, D., Klavins, E.: Approximating stochastic biochemical processes with Wasserstein pseudometrics. IET Systems Biology **4**, 193–211 (2010)
43. Visser, W., Havelund, K., Brat, G., Park, S., Lerda, F.: Model checking programs. Automated Software Engineering **10**(2), 203–232 (Apr 2003)