

Approximate Bisimulation Minimisation

Stefan Kiefer  

Department of Computer Science, University of Oxford, UK

Qiyi Tang  

Department of Computer Science, University of Oxford, UK

Abstract

We propose polynomial-time algorithms to minimise labelled Markov chains whose transition probabilities are not known exactly, have been perturbed, or can only be obtained by sampling. Our algorithms are based on a new notion of an approximate bisimulation quotient, obtained by lumping together states that are exactly bisimilar in a slightly perturbed system. We present experiments that show that our algorithms are able to recover the structure of the bisimulation quotient of the unperturbed system.

2012 ACM Subject Classification Theory of computation → Program verification; Theory of computation → Models of computation; Mathematics of computing → Probability and statistics

Keywords and phrases Markov chains, Behavioural metrics, Bisimulation

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2021.48

Related Version *Full Version*: <https://arxiv.org/abs/2110.00326> [15]

Supplementary Material

Software (Source Code): <https://github.com/qiyitang71/approximate-quotienting>

archived at `swh:1:dir:42b8e694ce904b906e52bac502b425e424412461`

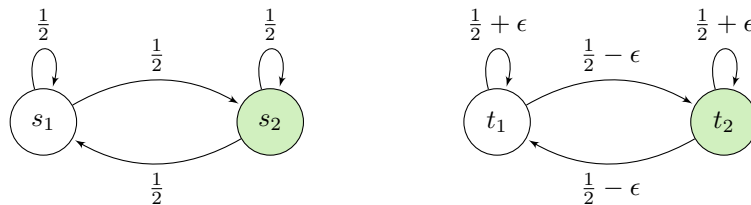
Dataset (Experimental Results): <https://bit.ly/3vcpb1Y>

Funding *Stefan Kiefer*: Supported by a Royal Society University Fellowship.

Acknowledgements We thank the anonymous reviewers of this paper for their constructive feedback.

1 Introduction

For the algorithmic analysis and verification of system models, computing the bisimulation quotient is a natural preprocessing step: it can make the system much smaller while preserving most properties of interest. This applies equally to probabilistic systems: probabilistic model checkers, e.g., Storm [12], speed up the verification process by “lumping” together states that are equivalent with respect to probabilistic bisimulation. While this is a safe approach, it may not be effective when the probabilities in the system are not known precisely. For



■ **Figure 1** Two intuitively similar labelled Markov chains.

example, in the labelled Markov chain shown in Figure 1 the states s_1, t_1 are intuitively “similar”, but they are not probabilistically bisimilar even though they carry the same label (here indicated with the colour white) and they both lead, with similar probabilities, to states s_2, t_2 , which are again intuitively “similar” but not probabilistically bisimilar.



© Stefan Kiefer and Qiyi Tang;

licensed under Creative Commons License CC-BY 4.0

41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021).

Editors: Mikołaj Bojańczyk and Chandra Chekuri; Article No. 48; pp. 48:1–48:16



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

For analysis and verification of a probabilistic system, tackling state space explosion is key. The more symmetry a system has (e.g., due to variables that do not influence the observable behaviour of the model), the greater are the benefits of computing a *quotient* system with respect to probabilistic bisimulation. However, when the transition probabilities in the Markov chain are perturbed or not known exactly, bisimulation minimisation may fail to capture the symmetries in the system and thus fail to achieve the objective of making the probabilistic system amenable to algorithmic analysis. The motivation of this paper is to counter this problem.

A principled approach towards this goal may consider notions of *distance* between probabilistic systems or between states in a probabilistic system. A *probabilistic bisimilarity pseudometric*, based on the Kantorovich metric, was introduced in [8, 9], which assigns to each pair of states s, t a number in the interval $[0, 1]$ measuring a distance between s, t : distance 0 means probabilistic bisimilar, and distance 1 means, in a sense, maximally non-bisimilar. This pseudometric can be computed in polynomial time [5], and, quoting [5], has “been studied in the context of systems biology, games, planning and security, among others”. The corresponding distances can be intuitively large though: the pseudometric yields a distance less than 1 only if the two states can reach, with the same label sequence, two states that are *exactly* bisimilar; see [18, Section 4]. As a consequence, for any $\epsilon > 0$, the states s_1, t_1 in Figure 1 have distance 1 in the probabilistic bisimilarity pseudometric of [9] (in the undiscounted version). From a slightly different point of view, a small perturbation of the transition probabilities in the model can change the distance between two states from 0 to 1.

Another pseudometric, ϵ -bisimulation \sim_ϵ , was defined in [10], which avoids this issue. It has natural characterisations in terms of games and can be computed in polynomial time using network-flow based algorithms [10]. The runtime of the algorithm from [10] is $O(|S|^7)$, where S is the set of states, thus not practical for large systems. A more fundamental problem lies in the fact that ϵ -bisimulation is not an equivalence: $s \sim_\epsilon t \sim_\epsilon u$ implies $s \sim_{2\epsilon} u$ (by the triangle inequality) but not necessarily $s \sim_\epsilon u$. Therefore, efficient minimisation algorithms via quotienting (such as partition refinement for exact probabilistic bisimilarity) are not available for ϵ -bisimulation.

In this paper we develop algorithms that, given a labelled Markov chain \mathcal{M} with possibly imprecise transition probabilities and a *slightly perturbed* version, say \mathcal{M}' , of \mathcal{M} , compute a compressed version, \mathcal{Q} , of \mathcal{M}' . By slightly perturbed we mean that for each state the successor distributions in \mathcal{M}' and \mathcal{M} have small (say, less than ϵ) L_1 -distance. We hope that \mathcal{Q} is not much bigger than the exact quotient of \mathcal{M} , and we design polynomial-time algorithms that fulfill this hope in practice, but we do not insist on computing the smallest possible \mathcal{Q} . Indeed, we show that, given an LMC, $\epsilon > 0$ and a positive integer k , it is NP-complete to decide whether there exists a perturbation of at most ϵ such that the (exact) bisimulation quotient of the perturbed system is of size k . See, e.g., [1] for an exact but non-polynomial approach, where the target number n of states is fixed, and a Markov chain with at most n states is sought that has minimal distance (with respect to the probabilistic bisimilarity pseudometric of [9]) to the given model.

It is not hard to prove (Proposition 2) that if an LMC can be made exactly bisimilar to another LMC by a perturbation of at most ϵ , then these two LMCs are also $\frac{\epsilon}{2}$ -bisimilar in the sense of [10]. If, in turn, two states are ϵ -bisimilar, one can show (see [3, Theorem 4]) that any linear-time property that depends only on the first k labels has similar probabilities in the two states, where similar means the difference in probabilities is at most $1 - (1 - \epsilon)^k$. Combining these two results, we obtain a continuity property that says if two LMCs can be

made bisimilar by a small perturbation, then any k -bounded linear-time property has similar probabilities in the two LMCs. In other words, our approximate minimisation approximately preserves the probability of bounded linear-time properties.

We apply our approximate minimisation algorithms in a setting of active learning. Here we assume we do not have access to the transition probabilities of the model; rather, for each state we only *sample* the successor distribution. Sampling gives us an approximation of the real Markov chain, and our approximate minimisation algorithms apply naturally. This allows us to lump states that are exactly bisimilar in the real model, but only approximately bisimilar in the sampled model. We give examples where in this way we *recover* the structure (not the precise transition probabilities) of the quotient of the exact model, knowing only the sampled model.

The rest of the paper is organised as follows. In Section 2 we introduce ϵ -quotient, a new notion of approximate bisimulation quotient. In Section 3 given an LMC, ϵ and $k > 0$, we show it is NP-complete to decide whether there exists an ϵ -quotient of size k . In Section 4 we present our approximate minimisation algorithms. We put them in a context of active learning in Section 5. In Section 6 we evaluate these algorithms on slightly perturbed versions of a number of LMCs taken from the probabilistic model checker PRISM [16]. We conclude in Section 7.

2 Preliminaries

We write \mathbb{N} for the set of nonnegative integers and \mathbb{Z}^+ for the set of positive integers. We write \mathbb{R} for the set of real numbers. Let S be a finite set. We denote by $\text{Distr}(S)$ the set of probability distributions on S . By default we view vectors, i.e., elements of \mathbb{R}^S , as row vectors. For a vector $\mu \in \mathbb{R}^S$ we write $\|\mu\|_1 := \sum_{s \in S} |\mu(s)|$ for its L_1 -norm. A vector $\mu \in [0, 1]^S$ is a distribution (resp. subdistribution) over S if $\|\mu\|_1 = 1$ (resp. $0 < \|\mu\|_1 \leq 1$). For a (sub)distribution μ we write $\text{support}(\mu) = \{s \in S : \mu(s) > 0\}$ for its support.

A partition of the states S is a set X consisting of pairwise disjoint subsets E of S with $\bigcup_{E \in X} E = S$. For an equivalence relation $\mathcal{R} \subseteq S \times S$, S/\mathcal{R} denotes its quotient set and $[s]_{\mathcal{R}}$ denotes the \mathcal{R} -equivalence class of $s \in S$.

A *labelled Markov chain* (LMC) is a quadruple $\langle S, L, \tau, \ell \rangle$ consisting of a nonempty finite set S of states, a nonempty finite set L of labels, a transition function $\tau : S \rightarrow \text{Distr}(S)$, and a labelling function $\ell : S \rightarrow L$.

We denote by $\tau(s)(t)$ the transition probability from s to t . Similarly, we denote by $\tau(s)(E) = \sum_{t \in E} \tau(s)(t)$ the transition probability from s to $E \subseteq S$. For the remainder of the paper, we fix an LMC $\mathcal{M} = \langle S, L, \tau, \ell \rangle$. Let $|\mathcal{M}|$ denote the number of states, $|S|$.

The direct sum $\mathcal{M}_1 \oplus \mathcal{M}_2$ of two LMCs $\mathcal{M}_1 = \langle S_1, L_1, \tau_1, \ell_1 \rangle$ and $\mathcal{M}_2 = \langle S_2, L_2, \tau_2, \ell_2 \rangle$ is the LMC formed by combining the state spaces of \mathcal{M}_1 and \mathcal{M}_2 .

An equivalence relation $\mathcal{R} \subseteq S \times S$ is a *probabilistic bisimulation* if for all $(s, t) \in \mathcal{R}$, $\ell(s) = \ell(t)$ and $\tau(s)(E) = \tau(t)(E)$ for each \mathcal{R} -equivalence class E . *Probabilistic bisimilarity*, denoted by $\sim_{\mathcal{M}}$ (or \sim when \mathcal{M} is clear), is the largest probabilistic bisimulation.

Any probabilistic bisimulation \mathcal{R} on \mathcal{M} induces a quotient LMC denoted by $\mathcal{M}/\mathcal{R} = \langle S/\mathcal{R}, L, \tau/\mathcal{R}, \ell/\mathcal{R} \rangle$ where the transition function $\tau/\mathcal{R}([s]_{\mathcal{R}})([t]_{\mathcal{R}}) = \tau(s)([t]_{\mathcal{R}})$ and the labelling function $\ell/\mathcal{R}([s]_{\mathcal{R}}) = \ell(s)$.

We define the notion of an *approximate quotient*. Let $\epsilon \geq 0$. An LMC \mathcal{Q} is an ϵ -quotient of \mathcal{M} if and only if there is transition function $\tau' : S \rightarrow \text{Distr}(S)$ such that for all $s \in S$ we have $\|\tau'(s) - \tau(s)\|_1 \leq \epsilon$ and \mathcal{Q} is the (exact) bisimulation quotient of the LMC $\mathcal{M}' = \langle S, L, \tau', \ell \rangle$, that is, $\mathcal{Q} = \mathcal{M}'/\sim_{\mathcal{M}'}$. Since the choice of τ' is not unique, there might be multiple ϵ -quotients

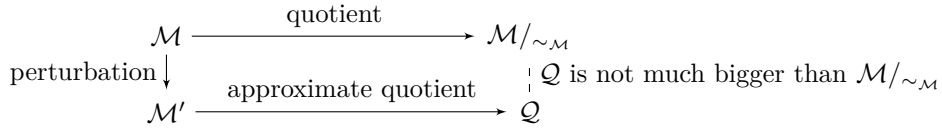
of \mathcal{M} . We are interested in the problem of obtaining an ϵ -quotient of \mathcal{M} with small state space. We retrieve the notion of (exact) quotient when $\epsilon = 0$. For s from \mathcal{M} , denote the state in \mathcal{Q} which corresponds to s by $[s]_{\mathcal{Q}}^{\epsilon}$ (or $[s]^{\epsilon}$ when \mathcal{Q} is clear).

The set $\Omega(\mu, \nu)$ of *couplings* of $\mu, \nu \in \text{Distr}(S)$ is defined as $\Omega(\mu, \nu) = \{ \omega \in \text{Distr}(S \times S) : \sum_{t \in S} \omega(s, t) = \mu(s) \wedge \sum_{s \in S} \omega(s, t) = \nu(t) \}$. Note that a coupling $\omega \in \Omega$ is a joint probability distribution with marginals μ and ν (see, e.g., [4, page 260-262]).

The ϵ -*lifting* of a relation $\mathcal{R} \subseteq S \times S$ proposed by Tracol et al. [19] is the relation $\mathcal{R}\uparrow_{\epsilon} \subseteq \text{Distr}(S) \times \text{Distr}(S)$ defined by $(\mu, \nu) \in \mathcal{R}\uparrow_{\epsilon}$ if there exists $\omega \in \Omega(\mu, \nu)$ such that $\sum_{(u,v) \in \mathcal{R}} \omega(u, v) \geq 1 - \epsilon$.

The ϵ -*bisimulation* (\sim_{ϵ}) by Desharnais et al. [10] is a relation $\mathcal{R} \subseteq S \times S$ in which for all $(s, t) \in \mathcal{R}$ we have $\ell(s) = \ell(t)$ and $(\tau(s), \tau(t)) \in \mathcal{R}\uparrow_{\epsilon}$. The ϵ -bisimulation is reflexive and symmetric, but in general not transitive; hence, it is not an equivalence relation.

3 Properties of Approximate Quotients

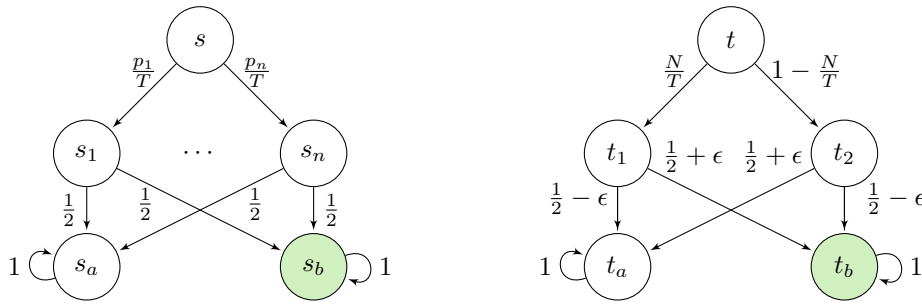


■ **Figure 2** Problem setup.

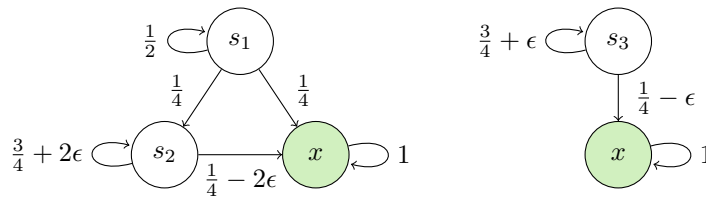
Recall from the introduction that we are given an LMC \mathcal{M}' , which is a slightly perturbed version of an (unknown) LMC \mathcal{M} . By slightly perturbed we mean that for each state the successor distributions in \mathcal{M}' and \mathcal{M} have small (say, less than ϵ) L_1 -distance. For example, with sampling we can obtain with high probability a perturbed system that has small distance with \mathcal{M} . Assume there are many symmetries, that is, lots of probabilistic bisimilar states in \mathcal{M} . The state space of \mathcal{M} can then be compressed a lot by (exact) quotienting. Since the transition probabilities are perturbed in \mathcal{M}' , the states that are probabilistic bisimilar in \mathcal{M} might become inequivalent in \mathcal{M}' ; as a result, the (exact) bisimulation quotient of \mathcal{M}' is much larger than that of \mathcal{M} . Given a small compression parameter $\epsilon_2 > 0$, we aim to compute an approximate quotient \mathcal{Q} , an ϵ' -quotient of \mathcal{M}' that satisfies two conditions: (1) ϵ' should be small, so that little precision is sacrificed; and (2) the state space of the quotient should be small, to speed up verification algorithms. Our contribution consists of approximate quotienting algorithms with (a) theoretical guarantees on goal (1) in Theorem 7 and Corollary 8, applying to both algorithms: ϵ' is bounded (and can be controlled) by a compression parameter ϵ_2 and the number of iterations i ; (b) empirical results on goal (2): the experiments show that our algorithms produce small quotients.

We first show that it is hard to find an ϵ_2 -quotient of \mathcal{M}' with minimum number of states: $\mathcal{Q}^* = \arg \min \{ |\mathcal{Q}| : \mathcal{Q} \text{ is an } \epsilon_2\text{-quotient of } \mathcal{M}' \}$. If there are several ϵ_2 -quotients of \mathcal{M}' of minimum size, \mathcal{Q}^* can be taken to be any one of them. Unfortunately, this problem is unlikely to have an efficient (polynomial-time) solution, as we will see from the next theorem that the decision version of this problem is NP-complete.

Given an LMC \mathcal{M}' , a compression parameter $\epsilon_2 > 0$ and a constant $k \in \mathbb{Z}^+$, it is NP-complete to decide whether there exists an ϵ_2 -quotient of \mathcal{M}' of size k . The hardness result is by reduction from the Subset Sum problem. Given a set $P = \{p_1, \dots, p_n\}$ and $N \in \mathbb{N}$, Subset Sum asks whether there exists a set $Q \subseteq P$ such that $\sum_{p_i \in Q} p_i = N$. Given an instance of Subset Sum $\langle P, N \rangle$ where $P = \{p_1, \dots, p_n\}$ and $N \in \mathbb{N}$, we construct an LMC;



■ **Figure 3** The LMC in the reduction for NP-hardness. All states have the same label a except s_b and t_b which have label b .



■ **Figure 4** An LMC in which $s_1 \sim_\epsilon s_3 \sim_\epsilon s_2$.

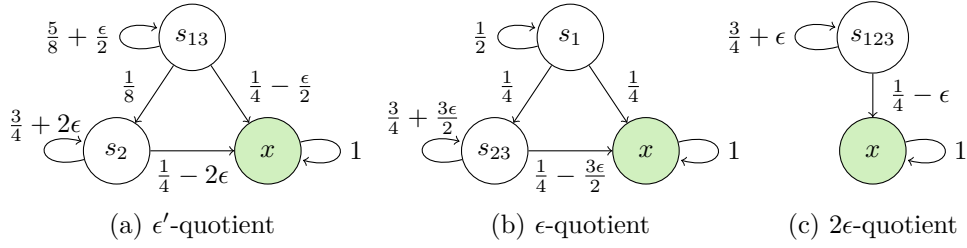
see Figure 3. Let $T = \sum_{p_i \in P} p_i$, $\epsilon = \frac{1}{2T}$ and $k = 5$. In the LMC, state s transitions to state s_i with probability p_i/T for all $1 \leq i \leq n$. Each state s_i transitions to s_a and s_b with equal probabilities. State t transitions to t_1 and t_2 with probability N/T and $1 - N/T$, respectively. State t_1 (resp. t_2) transitions to t_a (resp. t_b) and t_b (resp. t_a) with probability $\frac{1}{2} - \epsilon$ and $\frac{1}{2} + \epsilon$, respectively. All the remaining states transition to the successor state with probability one. States s_b and t_b have label b and all other states have label a . We can show that $\langle P, N \rangle \in \text{Subset Sum} \iff$ there exists an $\frac{1}{2T}$ -quotient of \mathcal{M}' of size 5.

► **Theorem 1.** *Given an LMC \mathcal{M}' , $\epsilon_2 \in (0, 1]$ and $k \in \mathbb{Z}^+$. The problem whether there exists an ϵ_2 -quotient of \mathcal{M}' of size k is NP-complete. It is NP-hard even for (fixed) $k = 5$.*

Due to the NP-hardness result, we hope to develop practical algorithms to compute approximate quotients of \mathcal{M}' that are small but not necessarily of minimum size. To do that, an intuitive idea is to merge “similar” states. As we have discussed in the introduction, merging states with small probabilistic bisimilarity distances might be insufficient. Consider the LMC shown in Figure 1. Assume $\epsilon > 0$. The states s_1 and t_1 (s_2 and t_2) have probabilistic bisimilarity distance one. Thus, to merge s_1, t_1 or s_2, t_2 , one needs to merge states with probabilistic bisimilarity distance one. Alternatively, we explore the relation of approximate quotient and ϵ -bisimulation. It is not hard to prove the following proposition:

► **Proposition 2.** *Let \mathcal{Q} be an ϵ_2 -quotient of \mathcal{M}' . Then in the LMC $\mathcal{M}' \oplus \mathcal{Q}$, we have $s \sim_{\frac{\epsilon_2}{2}} [s]_{\mathcal{Q}}^{\epsilon_2}$ for all s from \mathcal{M}' .*

Proposition 2 suggests that ϵ_2 -quotients and ϵ_2 -bisimulation are related. The runtime of the algorithm to compute the ϵ_2 -bisimulation in [10] is $O(|S|^7)$ which makes it not practical for large systems. Furthermore, the algorithms based on merging states that are ϵ_2 -bisimilar may produce an ϵ' -quotient where ϵ' is large, violating the first condition of a satisfying approximate quotient. Assume the positive number ϵ is much smaller than $\frac{1}{8}$. Let us choose the compression parameter ϵ_2 to be the same as ϵ . We compute the ϵ -bisimulation of the LMC shown in Figure 4 and get $s_1 \sim_\epsilon s_3 \sim_\epsilon s_2$. Since ϵ -bisimulation is not an equivalence



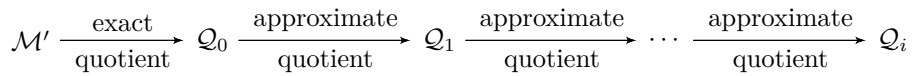
■ **Figure 5** (a) An ϵ' -quotient obtained by merging s_1 and s_3 where ϵ' is at least $\frac{1}{4} + \epsilon$; (b) An ϵ -quotient obtained by merging s_2 and s_3 ; (c) A 2ϵ -quotient obtained by merging s_1 , s_2 and s_3 .

relation, $s_1 \sim_\epsilon s_2$ does not necessarily follow. Indeed, in this LMC, we have $s_1 \sim_{2\epsilon} s_2$ but not $s_1 \sim_\epsilon s_2$. If s_2 and s_3 , related by \sim_ϵ , are chosen to be merged, the resulting LMC in Figure 5(b) is an ϵ -quotient. However, if s_1 and s_3 are (unfortunately) chosen to be merged, the resulting LMC, shown in Figure 5(a), is an ϵ' -quotient where ϵ' cannot be smaller than $\frac{1}{4} + \epsilon$. This ϵ' , much bigger than ϵ under the assumption that ϵ is much smaller than $\frac{1}{8}$, makes the resulting LMC undesirable. This example shows that arbitrarily merging states that are ϵ -bisimilar may not work. The LMC in Figure 5(c) is obtained by merging s_1 , s_2 and s_3 , the states that are related by the transitive closure of \sim_ϵ . We show in [15] that for any $n \in \mathbb{Z}^+$ there exists an LMC $\mathcal{M}(n)$ such that merging all states in $\mathcal{M}(n)$ that are related by the transitive closure of \sim_ϵ results in an ϵ' -quotient where ϵ' is at least $n\epsilon$.

Lemma 3, the additivity lemma, asserts an additivity property of approximate quotients. In Section 4, this lemma will be applied as the two minimisation algorithms successively compute a sequence of approximate quotients.

► **Lemma 3.** *Consider three LMCs \mathcal{M}_1 , \mathcal{M}_2 and \mathcal{M}_3 . Let $\epsilon_1 \geq 0$ and \mathcal{M}_2 be an ϵ_1 -quotient of \mathcal{M}_1 . Let $\epsilon_2 \geq 0$ and \mathcal{M}_3 be an ϵ_2 -quotient of \mathcal{M}_2 . Then \mathcal{M}_3 is an $(\epsilon_1 + \epsilon_2)$ -quotient of \mathcal{M}_1 .*

4

 Approximate Minimisation Algorithms


■ **Figure 6** Overview of the minimisation algorithms. Lemma 3 applies to \mathcal{M}' , \mathcal{Q}_0 , $\mathcal{Q}_1, \dots, \mathcal{Q}_i$.

In this section, we present two practical minimisation algorithms that compute approximate quotients of \mathcal{M}' . Given an LMC $\mathcal{M}' = \langle S, L, \tau_\epsilon, \ell \rangle$ with perturbed transition probabilities and a small compression parameter ϵ_2 . Both algorithms start by computing \mathcal{Q}_0 , the exact quotient of \mathcal{M}' . They proceed in iterations and compute a sequence of approximate quotients where the approximate quotient (\mathcal{Q}_i) computed at the end of the i th iteration is an ϵ_2 -quotient of the quotient (\mathcal{Q}_{i-1}) given at the beginning of that iteration. Using the additivity lemma, we can show that the (approximate) quotient \mathcal{Q}_i after the i th iteration is an $i\epsilon_2$ -quotient of \mathcal{M}' . See Figure 6 for an overview of this approach. Each iteration computes a partition of the state space, lumps the states that are together in the partition and concludes with taking the exact quotient.

4.1 Local Bisimilarity Distance

We define the notion of *local bisimilarity distance*, denoted by $d_{local}^{\mathcal{M}}$ (or d_{local} when \mathcal{M} is clear). Intuitively, two states s and t are at small local bisimilarity distance if they are probabilistic bisimilar in an LMC which is slightly perturbed only at the successor distributions of s and t . We provide a polynomial-time algorithm to compute the local bisimilarity distance. Given an LMC $\mathcal{M}' = \langle S, L, \tau_\epsilon, \ell \rangle$ (with perturbed transition probabilities) and a small compression parameter ϵ_2 , we propose an iterative minimisation algorithm to compute approximate quotients of \mathcal{M}' by merging state pairs with small local bisimilarity distances. In each iteration of the algorithm, we select the state pair with the same label and the minimum local bisimilarity distance if such distance is at most ϵ_2 . We compute a partition in which this state pair are together and lump together the states that are together in the partition. The algorithm terminates when no pairs can be lumped, that is, all state pairs have local bisimilarity distances greater than ϵ_2 .

Computing Local Bisimilarity Distances

Given two different states $s, t \in S$ with the same label. We want to compute a new transition function τ'_ϵ by only changing the successor distributions of s and t ($\tau_\epsilon(s)$ and $\tau_\epsilon(t)$, respectively) such that $\{s, t\}$ belongs to an \mathcal{R} -induced partition where \mathcal{R} is a probabilistic bisimulation of the LMC $\mathcal{M}'' = \langle S, L, \tau'_\epsilon, \ell \rangle$. Let T be the set of the all transition functions that satisfy this condition, more precisely, we define $T = \{\tau'_\epsilon : \tau'_\epsilon(x) = \tau_\epsilon(x) \forall x \notin \{s, t\} \wedge \{s, t\} \in S/\mathcal{R} \text{ where } \mathcal{R} \text{ is a probabilistic bisimulation of the LMC } \mathcal{M}'' = \langle S, L, \tau'_\epsilon, \ell \rangle\}$. The local bisimilarity distance is defined as $d_{local}^{\mathcal{M}'}(s, t) = \inf_{\tau' \in T} \max\{\|\tau'(s) - \tau_\epsilon(s)\|_1, \|\tau'(t) - \tau_\epsilon(t)\|_1\}$. It is not immediately clear how to compute it.

By the definition of T , the probabilistic bisimulation \mathcal{R} is the same for any LMC $\langle S, L, \tau'_\epsilon, \ell \rangle$ with $\tau'_\epsilon \in T$. Let us define the partition $X = S/\mathcal{R}$ where \mathcal{R} is the common probabilistic bisimulation. The local bisimilarity distance can be computed by using X :

► **Proposition 4.** *We have $d_{local}^{\mathcal{M}'}(s, t) = \frac{1}{2} \|(\tau_\epsilon(s)(E))_{E \in X} - (\tau_\epsilon(t)(E))_{E \in X}\|_1$.*

It turns out that X can simply be computed by Algorithm 1. As this algorithm is basically taking the (exact) quotient of the LMC constructed on line 1, it runs in polynomial time. It follows from Proposition 4 that the local bisimilarity distance can be computed in polynomial time.

■ **Algorithm 1** Compute Partition for Local Bisimilarity Distances.

Input: An LMC $\mathcal{M}' = \langle S, L, \tau_\epsilon, \ell \rangle$, a state pair $(s, t) \in S \times S$

Output: A partition X over S containing $\{s, t\}$

- 1 Construct a new LMC \mathcal{M}'' from \mathcal{M}' by introducing a new label, labelling both s and t with the new label and making both s and t absorbing¹
 - 2 $X := S/\sim_{\mathcal{M}''}$
-

► **Example 5.** Assume $\epsilon < \frac{1}{2}$. Consider the LMC shown in Figure 1. Let τ_ϵ denote its transition function. To compute the local bisimilarity distance of s_1 and t_1 , we first compute the partition containing $\{s_1, t_1\}$: $X = \{\{s_1, t_1\}, \{s_2\}, \{t_2\}\}$. We have

¹ An absorbing state is a state that, once entered, cannot be left; that is, a state with self-loop.

$(\tau_\epsilon(s_1)(E))_{E \in X} = (\frac{1}{2}, \frac{1}{2}, 0)$ and $(\tau_\epsilon(t_1)(E))_{E \in X} = (\frac{1}{2} + \epsilon, 0, \frac{1}{2} - \epsilon)$. By Proposition 4, the local bisimilarity distance is $d_{local}(s_1, t_1) = \frac{1}{2} \|(\tau_\epsilon(s_1)(E))_{E \in X} - (\tau_\epsilon(t_1)(E))_{E \in X}\|_1 = \frac{1}{2}$. Similarly, we have $d_{local}(s_2, t_2) = \frac{1}{2}$.

■ **Algorithm 2** LMC Minimisation Using Local Bisimilarity Distances.

Input: An LMC $\mathcal{M}' = \langle S, L, \tau_\epsilon, \ell \rangle$, a compression parameter ϵ_2
Output: An LMC \mathcal{Q}_i

- 1 $i := 0$
- 2 $\mathcal{Q}_i := \mathcal{M}' / \sim_{\mathcal{M}'}$ and $\mathcal{Q}_i = \langle S^{\mathcal{Q}_i}, L, \tau^{\mathcal{Q}_i}, \ell^{\mathcal{Q}_i} \rangle$
- 3 **while** $\exists u, v \in S^{\mathcal{Q}_i}$ such that $u \neq v$ and $\ell^{\mathcal{Q}_i}(u) = \ell^{\mathcal{Q}_i}(v)$ and $d_{local}^{\mathcal{Q}_i}(u, v) \leq \epsilon_2$ **do**
- 4 $(s, t) = \arg \min \{d_{local}^{\mathcal{Q}_i}(u, v) : (u, v) \in S^{\mathcal{Q}_i} \times S^{\mathcal{Q}_i} \wedge u \neq v \wedge \ell^{\mathcal{Q}_i}(u) = \ell^{\mathcal{Q}_i}(v)\}$
- 5 Compute X_i by running Algorithm 1 with input \mathcal{Q}_i and (s, t)
- 6 Construct an LMC $\mathcal{M}_{i+1} := \langle X_i, L, \tau^{\mathcal{M}_{i+1}}, \ell^{\mathcal{M}_{i+1}} \rangle$ from \mathcal{Q}_i where

$$\tau^{\mathcal{M}_{i+1}}(E) := \begin{cases} (\tau^{\mathcal{Q}_i}(u)(E'))_{E' \in X_i} \text{ for any } u \in E \text{ if } E \in X_i \text{ and } E \neq \{s, t\} \\ \frac{(\tau^{\mathcal{Q}_i}(s)(E'))_{E' \in X_i} + (\tau^{\mathcal{Q}_i}(t)(E'))_{E' \in X_i}}{2} \text{ if } E = \{s, t\} \end{cases}$$
 and $\ell^{\mathcal{M}_{i+1}}(E) := \ell^{\mathcal{Q}_i}(u)$ for $E \in X_i$ and any $u \in E$
- 7 $\mathcal{Q}_{i+1} := \mathcal{M}_{i+1} / \sim_{\mathcal{M}_{i+1}}$
- 8 $i := i + 1$
- 9 **end**

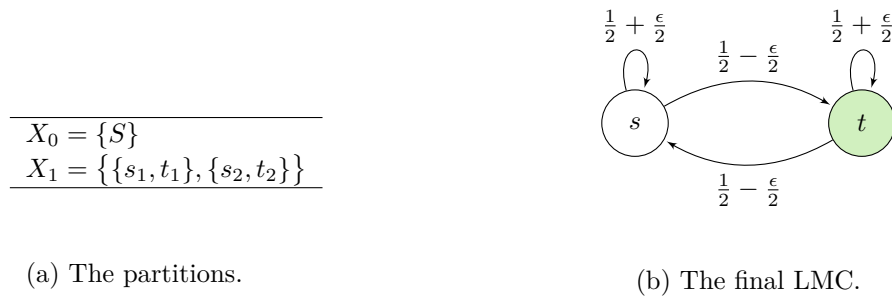
Minimisation Algorithm Using Local Bisimilarity Distances

Algorithm 2 shows the minimisation algorithm using local bisimilarity distances. The input is an LMC \mathcal{M}' and a compression parameter ϵ_2 . We start by initializing an index i to 0 and building the quotient LMC $\mathcal{Q}_0 = \mathcal{M}' / \sim_{\mathcal{M}'}$. If there are no states in \mathcal{Q}_i with local bisimilarity distance less than ϵ_2 , the algorithm terminates. Otherwise, it steps into the i 'th iteration of the loop and computes the local bisimilarity distances for all pairs of states in \mathcal{Q}_i with the same label. It selects the state pair (s, t) which has the smallest local bisimilarity distance on line 4. It then computes the new approximate quotient by merging states s and t on line 5-7. This computation is in three steps where the first step is to compute the partition X_i (line 5) by running Algorithm 1 with input \mathcal{Q}_i and the state pair (s, t) . The second step is to construct a new LMC \mathcal{M}_{i+1} by setting X_i as its state space (line 6). The final step is to compute a new approximate quotient \mathcal{Q}_{i+1} by taking the exact quotient of the LMC \mathcal{M}_{i+1} obtained from the previous step. We increment i at the end of the iteration and continue with another iteration if there are states in \mathcal{Q}_{i+1} with local bisimilarity distance at most ϵ_2 . Since there are finitely many states and it is polynomial time to compute the local bisimilarity distances, the algorithm always terminates and runs in polynomial time.

4.2 Minimisation by Approximate Partition Refinement

Consider the LMC in Figure 1. Assume $\epsilon < \frac{1}{2}$ and $\epsilon_2 < \frac{1}{2}$. The minimisation algorithm using local bisimilarity distance (Algorithm 2) cannot merge states s_1, t_1 (or s_2, t_2) as $d_{local}(s_1, t_1) = d_{local}(s_2, t_2) = \frac{1}{2} > \epsilon_2$ as shown by Example 5.

We introduce an approximate partition refinement, a polynomial algorithm similar to the exact partition refinement, which can fix this problem. In the exact partition refinement algorithm, the states will only remain in the same set in an iteration if they have the



■ **Figure 7** Example of running the minimisation algorithm using approximate partition refinement (Algorithm 3) on the LMC in Figure 1.

same label and their probability distributions over the previous partition are the same. Similarly, we design the approximate partition refinement such that states only remain in the same set in an iteration if they have the same label and the L_1 -distance between the probability distributions over the previous partition is small, say, at most ϵ_2 . Given an LMC $\mathcal{M}' = \langle S, L, \tau_\epsilon, \ell \rangle$ with perturbed transition probabilities, the minimisation algorithm using the approximate partition refinement also proceeds in iterations. In each iteration, the approximate partition refinement computes a partition X and then the states which are together in X are lumped to form a new LMC. To make sure the new LMC is a quotient, we take the (exact) quotient of this LMC as our new approximate quotient. The algorithm continues when there are states that could be lumped, and it terminates when all sets in the partition computed by the approximate partition refinement are singletons, that is, no states can be lumped.

► **Example 6.** Consider again the LMC in Figure 1. Assume $\epsilon < \frac{1}{2}$ and the compression parameter $\epsilon_2 \geq 2\epsilon$. We run the above-mentioned minimisation algorithm using the approximate partition refinement. It will only run for one iteration of approximate partition refinement, as we will see in the following. Figure 7(a) shows the partitions of this iteration. At the beginning of the approximate partition refinement, we have partition X_0 as all states are in the same set. The states are then split by the labels and we get partition X_1 . There is no further split since the L_1 -distance between the probability distributions over X_1 from s_1 and t_1 (resp. s_2 and t_2) is 2ϵ which is bounded by the compression parameter ϵ_2 , that is, $\|(\tau(s_1)(E))_{E \in X_1} - (\tau(t_1)(E))_{E \in X_1}\|_1 = \|(\tau(s_2)(E))_{E \in X_1} - (\tau(t_2)(E))_{E \in X_1}\|_1 = 2\epsilon \leq \epsilon_2$. The states together in X_1 are then lumped to form the new LMC shown in Figure 7(b). The algorithm terminates as no states in the new LMC can be lumped.

Approximate Partition Refinement

Given a compression parameter ϵ_2 , the approximate partition refinement is shown in Algorithm 3. At the beginning, an index i is initialized to zero and we have $X_0 = \{S\}$, that is, all states are in the same set. In a refinement step, we increment i and split each set $E \in X_{i-1}$ into one or more sets. We iterate through all $E \in X_{i-1}$ and for each E we construct a set X_E , a partition of E . Starting with $X_E = \emptyset$, we iterate over all $s \in E$ (line 6). After each iteration, the current $s \in E$ appears in one set in X_E : either as a singleton or as an additional state in an already existing set in X_E . We give more details on this loop (lines 6-14) below. After having partitioned E into X_E , we add all sets in X_E to the new

■ **Algorithm 3** Approximate Partition Refinement.

Input: An LMC $\mathcal{M}' = \langle S, L, \tau_\epsilon, \ell \rangle$, a compression parameter ϵ_2
Output: A partition X over S

```

1  $i := 0; X_0 := \{S\}$ 
2 repeat
3    $i := i + 1; X_i := \emptyset$ 
4   foreach  $E \in X_{i-1}$  do
5      $X_E := \emptyset$ 
6     for  $s \in E$  do
7        $ESet := \{E' \in X_{i-1} : \text{for all } t \in E' \text{ we have } \ell(s) = \ell(t) \text{ and}$ 
8          $\|(\tau_\epsilon(s)(E))_{E \in X_{i-1}} - (\tau_\epsilon(t)(E))_{E \in X_{i-1}}\|_1 \leq \epsilon_2\}$ 
9       if  $ESet = \emptyset$  then  $E' := \{s\}$ 
10      else
11         $E' := \arg \min_{E' \in ESet} \left\{ \frac{\sum_{t \in E'} \|(\tau_\epsilon(s)(E))_{E \in X_{i-1}} - (\tau_\epsilon(t)(E))_{E \in X_{i-1}}\|_1}{|E'|} \right\}$ 
12        remove  $E'$  from  $X_E$ ;  $E' := E' \cup \{s\}$ 
13      end
14      add  $E'$  to  $X_E$ 
15    end
16     $X_i := X_i \cup X_E$ 
17 until  $X_i = X_{i-1}$ 

```

partition X_i . The way we split the sets ensures that for any two states from the same set in X_i the L_1 -distance between the successor distributions over X_{i-1} is at most ϵ_2 . The algorithm terminates when no splitting can be done. Let X be the final partition produced by the approximate partition refinement. For any two states $s, t \in E$ where $E \in X$, we have $\ell(s) = \ell(t)$ and $\|(\tau_\epsilon(s)(E'))_{E' \in X} - (\tau_\epsilon(t)(E'))_{E' \in X}\|_1 \leq \epsilon_2$.

Let us give more details on the loop (lines 6-14) that partitions an $E \in X_i$. For a state $s \in E$, a candidate set $ESet$ is computed such that for all $E' \in ESet$ the state s and all $x \in E'$ have the same label and the L_1 -distance between the successor distributions over X_{i-1} of s and any $x \in E'$ is at most ϵ_2 (line 7). If $ESet$ is empty, we add the singleton $\{s\}$ into X_E (line 8 and 13). If there is only one set E' in $ESet$, we add s to the set E' . Otherwise, if there are multiple elements in $ESet$ that satisfy the condition, we select the one as E' such that the average L_1 -distance between the successor distributions of s and $x \in E'$ is the smallest (line 10). We add s to the selected set E' and include E' in X_E (line 10-13).

Minimisation Algorithm Using Approximate Partition Refinement

The minimisation algorithm using approximate partition refinement is shown in Algorithm 4. The input is the same as the first minimisation algorithm: an LMC \mathcal{M}' and a compression parameter ϵ_2 . An index i is initialised to 0. Similar to the approximate minimisation algorithm using local bisimilarity distances, we also start by computing the quotient LMC $\mathcal{Q}_0 = \mathcal{M}' / \sim_{\mathcal{M}'}$. It then steps into a loop. We compute the approximate partition X_i of \mathcal{Q}_i on line 4 and construct a new LMC \mathcal{M}_{i+1} by setting X_i as its state space on line 5. For any state $E \in X_i$, we set the probability distribution as the average probability distribution over

■ **Algorithm 4** LMC Minimisation by Approximate Partition Refinement.

Input: An LMC $\mathcal{M}' = \langle S, L, \tau_\epsilon, \ell \rangle$, a compression parameter ϵ_2
Output: An LMC \mathcal{Q}_i

- 1 $i := 0$
- 2 $\mathcal{Q}_i := \mathcal{M}_i / \sim_{\mathcal{M}_i}$ where $\mathcal{M}_i = \mathcal{M}'$ and $\mathcal{Q}_i = \langle S^{\mathcal{Q}_i}, L, \tau^{\mathcal{Q}_i}, \ell^{\mathcal{Q}_i} \rangle$
- 3 **repeat**
- 4 Compute X_i by running Algorithm 3 with \mathcal{Q}_i and ϵ_2 as input
- 5 Construct an LMC $\mathcal{M}_{i+1} := \langle X_i, L, \tau^{\mathcal{M}_{i+1}}, \ell^{\mathcal{M}_{i+1}} \rangle$ from \mathcal{Q}_i where

$$\tau^{\mathcal{M}_{i+1}}(E) := \sum_{u \in E} \frac{(\tau^{\mathcal{Q}_i}(u)(E'))_{E' \in X_i}}{|E|} \text{ and } \ell^{\mathcal{M}_{i+1}}(E) := \ell^{\mathcal{Q}_i}(x) \text{ for all } E \in X_i \text{ and any } x \in E$$
- 6 $\mathcal{Q}_{i+1} := \mathcal{M}_{i+1} / \sim_{\mathcal{M}_{i+1}}$
- 7 $i := i + 1$
- 8 **until** $|S^{\mathcal{Q}_i}| = |S^{\mathcal{Q}_{i-1}}|$

X_i from all $u \in E$. The label of any $E \in X$ is set to $\ell^{\mathcal{Q}_i}(u)$ where u can be any state from E . A new approximate quotient \mathcal{Q}_{i+1} is obtained by taking the exact quotient of \mathcal{M}_{i+1} . We increment i at the end of the iteration and continue another iteration if the size of the state space of the new approximate quotient decreases. Otherwise, the algorithm terminates as we have no states to merge. As there are finitely many states, the algorithm always terminates.

Let $i \in \mathbb{N}$. The following theorem applies to both the LMCs \mathcal{Q}_i from Algorithm 2 and those from Algorithm 4.

► **Theorem 7.** *For all $i \in \mathbb{N}$, we have that \mathcal{Q}_{i+1} is an ϵ_2 -quotient of \mathcal{Q}_i . Furthermore, by the additivity lemma, we have that \mathcal{Q}_i is an $i\epsilon_2$ -quotient of \mathcal{M}' .*

In the case that $\mathcal{M}' = \langle S, L, \tau_\epsilon, \ell \rangle$ is a slightly perturbed version of $\mathcal{M} = \langle S, L, \tau, \ell \rangle$, that is, for all $s \in S$ we have $\|\tau(s) - \tau_\epsilon(s)\|_1 \leq \epsilon$, the following corollary holds:

► **Corollary 8.** *For all $i \in \mathbb{N}$, we have that \mathcal{Q}_i is an $(\epsilon + i\epsilon_2)$ -quotient of \mathcal{M} .*

5 Active LMC Learning

We apply our approximate minimisation algorithms in a setting of active learning. Before that, we first describe how to obtain a perturbed LMC \mathcal{M}' by sampling. Assume that we want to learn the transition probabilities of an LMC \mathcal{M} , that is, the state space, the labelling and the transitions are known. We also assume the system under learning (SUL) \mathcal{M} could answer the query *next* which takes a state s as input and returns a successor state of s according to the transition probability distribution $\tau(s)$.

Given a state s of the LMC. We denote by x_s the number of successor states of s and by n_s the number of times we query the SUL on *next*(s). Let $N_{s,t}$ be the frequency counts of the query result t , that is, the number of times a successor state t appears as the result returned by the queries. We approximate the transition probability distribution by $\tau_\epsilon(s)$ where $\tau_\epsilon(s)(t) = \frac{N_{s,t}}{n_s}$ for all successor states t of s . (Such an estimator is called an empirical estimator in the literature.)

Intuitively, the more queries we ask the SUL, the more accurate the approximate probability distribution $\tau_\epsilon(s)$ would be. In fact, the following theorem holds [2, Section 6.4], [6].

► **Theorem 9.** *Let $\epsilon > 0$ be an error parameter and $\delta > 0$ be an error bound. Let $s \in S$. We have $\Pr(\|\tau(s) - \tau_\epsilon(s)\|_1 \leq \epsilon) \geq 1 - \delta$ for $n_s \geq \frac{1}{2\epsilon^2} \ln(\frac{2x_s}{\delta})$.*

For each state $s \in S$, we query the SUL on $next(s)$ for $n_s \geq \frac{1}{2\epsilon^2} \ln(\frac{2x_s}{\delta})$ times. We can make δ small since it appears in the logarithmic term. We then approximate the transition function by τ_ϵ and construct a hypothesis LMC $\mathcal{M}' = \langle S, L, \tau_\epsilon, \ell \rangle$. Since the queries $next(s)$ and $next(t)$ for all $s, t \in S$ and $s \neq t$ are mutually independent, by Theorem 9, we have that $\Pr(\forall s \in S : \|\tau(s) - \tau_\epsilon(s)\|_1 \leq \epsilon) \geq (1 - \delta)^{|S|}$.

We then apply the minimisation algorithms with compression parameter ϵ_2 on \mathcal{M}' and obtain a minimised system \mathcal{Q}_i which is an $i\epsilon_2$ -quotient of \mathcal{M}' , the LMC constructed by sampling. Since with high probability the LMC \mathcal{M}' (or its exact quotient \mathcal{Q}_0) has small distance ϵ with the SUL \mathcal{M} , it follows from Corollary 8 that with high probability the minimised system \mathcal{Q}_i is an ϵ' -quotient of \mathcal{M} where ϵ' is small: for all $i \in \mathbb{N}$, we have $\Pr(\mathcal{Q}_i \text{ is an } \epsilon'\text{-quotient of } \mathcal{M} \text{ with } \epsilon' \leq \epsilon + i\epsilon_2) \geq (1 - \delta)^{|S|}$. The probability does not come from our minimisation algorithms and depends solely on the sampling procedure.

6 Experiments

In this section, we evaluate the performance of approximate minimisation algorithms on a number of LMCs. These LMCs model randomised algorithms and probabilistic protocols that are part of the probabilistic model checker PRISM [16]. The LMCs we run experiments on have less than 100,000 states and model the following protocols or randomised algorithms: Herman's self-stabilisation algorithm [13], the synchronous leader election protocol by Itai and Rodeh [14], the bounded retransmission protocol [7], the Crowds protocol [17] and the contract signing protocol by Even, Goldreich and Lempel [11].

We implemented algorithms to obtain the slightly perturbed LMCs \mathcal{M}' . We call LMCs with fewer than 300 states small; otherwise we call them large. For small LMCs, we sample the successor distribution for each state and obtain an approximation of it with error parameter ϵ and error bound δ . For large LMCs, sampling is not practical as the sample size required by Theorem 9 is very large. For these LMCs, we perturb the successor distribution by adding small noise to the successor transition probabilities so that for each state with at least probability $1 - \delta$ the L_1 -distance of the successor distributions in the perturbed and unperturbed systems is at most ϵ and otherwise the L_1 -distance is 2ϵ . We vary the error parameter ϵ in the range of $\{0.00001, 0.0001, 0.001, 0.01\}$ and fix the error bound $\delta = 0.01$. For each unperturbed LMC and a pair of ϵ and δ , we generate 5 perturbed LMCs.

We also implemented the two minimisation algorithms in Java: Algorithm 2 and Algorithm 4. The source code is publicly available². We show some representative results in [15]. The full experimental results are publicly available³.

For the small LMCs, we apply both approximate minimisation algorithms to the perturbed LMCs with $\epsilon_2 \in \{0.00001, 0.0001, 0.001, 0.01, 0.1\}$. The results for a small LMC which models the Herman's self-stabilisation algorithm is shown on the left of Table 1. For the large LMCs, we only apply the approximate minimisation algorithm using approximate partition refinement to the perturbed LMCs, since the other minimisation algorithm could not finish on the large LMCs with timeout of two hours. The results for a large LMC which models the bounded retransmission protocol is shown on the right of Table 1.

For almost all models, given a perturbed LMC, we are able to recover the structure of the quotient of the unperturbed LMC when ϵ_2 is appropriately chosen, that is, ϵ_2 is no less than ϵ and is not too big; for example, see Table 1 where the rows are highlighted in yellow. However, when ϵ_2 is too big, the approximate minimisation algorithms may aggressively

² <https://github.com/qiyitang71/approximate-quotienting>

³ <https://bit.ly/3vcpb1Y>

■ **Table 1** In the tables, local and apr stand for the minimisation algorithms using local bisimilarity distance and approximate partition refinement, respectively. The tables show the results for the first perturbed LMC (labeled with #1) among the five perturbed LMCs generated by sampling or perturbing with $\epsilon = 0.0001$. (Left) Results of running the two minimisation algorithms on the LMC that models Herman’s self-stabilisation algorithm with 5 processes. (Right) Results of running apr on the LMC that models the bounded retransmission protocol with $N = 32$ and $MAX = 2$.

Herman5	# states	# trans	# iter	BRP32-2	# states	# trans	# iter
$\mathcal{M} \& \mathcal{M}'$	32	244		$\mathcal{M} \& \mathcal{M}'$	1349	1731	
$\mathcal{M}/\sim_{\mathcal{M}}$	4	11		$\mathcal{M}/\sim_{\mathcal{M}}$	647	903	
$\mathcal{M}'/\sim_{\mathcal{M}'}$	23	167		$\mathcal{M}'/\sim_{\mathcal{M}'}$	961	1343	
Perturbed LMC #1				Perturbed LMC #1			
$\epsilon_2 = 0.00001$				$\epsilon_2 = 0.00001$			
local & apr	23	167	0	apr	879	1230	2
$\epsilon_2 = 0.0001$				$\epsilon_2 = 0.0001$			
local & apr	22	143	1	apr	705	986	2
$\epsilon_2 \in \{0.001, 0.01, 0.1\}$				$\epsilon_2 \in \{0.001, 0.01\}$			
local	22	143	1	apr	647	903	1
apr	4	11	1	$\epsilon_2 = 0.1$			
				apr	196	387	1

merge some states in the perturbed LMC and result in a quotient whose size is even smaller than that of the quotient of the unperturbed LMC, as highlighted in red in Table 1. Also, we find that, as expected, the exact partition refinement in general could not recover the structure of quotient of the original LMCs, except for the LMCs which model the synchronous leader election protocol by Itai and Rodeh. Furthermore, compared to the other approximate minimisation algorithm using the local bisimilarity distance, the one using approximate partition refinement performs much better in terms of running time and the ability to recover the structure of the quotient of the original model.

One might ask whether the minimisation algorithm using approximate partition refinement always performs better than the one using the local bisimilarity distances. In general, this is not the case as shown by Example 10.

► **Example 10.** Consider the LMC $\mathcal{M} = \langle S, L, \tau, \ell \rangle$ shown in Figure 8. Let $\epsilon_2 = 0.1$. First, we run Algorithm 2. It proceeds in two iterations. In the first iteration, it computes the local bisimilarity distances for all pairs of states with the same label. We have $d_{local}(s_1, s_2) = d_{local}(s_2, s_3) = 0.54$ and $d_{local}(s_1, s_3) = 0.04$. It then selects the pair s_1 and s_3 of which the local bisimilarity distance is less than ϵ_2 and is the smallest. These two states are merged into s_{13} in the LMC shown on the left of Figure 9. In the second iteration, the only pair of states with the same label are s_{13} and s_2 . Since $d_{local}(s_{13}, s_2) = 0.06 \leq \epsilon_2$, they are merged and we arrive at the final LMC shown on the right of Figure 9.

Next, we run Algorithm 4 with the same inputs. In the first iteration, we run approximate partition refinement on line 5 (Algorithm 3) and present Table 2 as the possible partitions of the algorithm. At the beginning of the approximate partition refinement, we have partition X_0 as all states are in the same set. The states are then split by the labels and we get partition X_1 . Next, we work on the set $\{s_1, s_2, s_3\}$. Suppose that we see s_1 and s_2 before s_3 . We have s_1 and s_2 remain together as $\|(\tau(s_1)(E))_{E \in X_1} - (\tau(s_2)(E))_{E \in X_1}\|_1 = 0.08 \leq \epsilon_2$. However, since $\|(\tau(s_3)(E))_{E \in X_1} - (\tau(s_2)(E))_{E \in X_1}\|_1 = 0.16 > \epsilon_2$, we have $ESet = \emptyset$ for s_3 on line 9 of Algorithm 3 and it is split out. In the next iteration, since

$\|(\tau(s_1)(E))_{E \in X_2} - (\tau(s_2)(E))_{E \in X_2}\|_1 = 0.54 > \epsilon_2$, $\{s_1, s_2\}$ is split into two singleton sets. The final partition X_3 in which all sets are singletons suggests no merging can be done and we are left with the original LMC \mathcal{M} .

This example also shows that the order of iterating through the states matters for the approximate partition refinement algorithm. Indeed, suppose we iterate through s_1 and s_3 before s_2 after arriving at the partition X_1 , we will have Table 3 as the partitions and finally get the LMC on the right of Figure 9 just as the other minimisation algorithm.

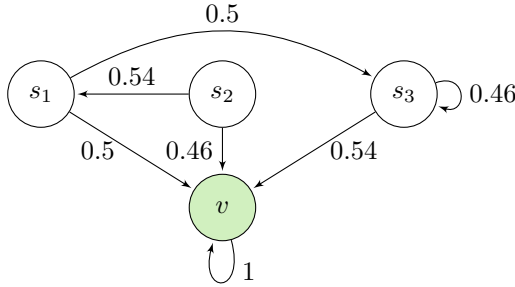


Figure 8 The LMC for which Algorithm 2 may perform better than Algorithm 4.

Table 2 Example of running Algorithm 3 on the LMC in Figure 8. (Suppose we iterate through s_1 and s_2 before s_3 .)

$X_0 = \{S\}$
$X_1 = \{\{s_1, s_2, s_3\}, \{v\}\}$
$X_2 = \{\{s_1, s_2\}, \{s_3\}, \{v\}\}$
$X_3 = \{\{s_1\}, \{s_2\}, \{s_3\}, \{v\}\}$

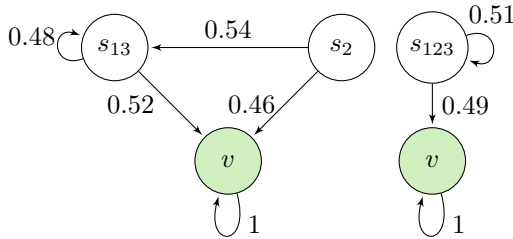


Figure 9 Two Steps of Running Algorithm 2.

Table 3 Example of running Algorithm 3 on the LMC in Figure 8. (Suppose we iterate through s_1 and s_3 before s_2 .)

$X_0 = \{S\}$
$X_1 = \{\{s_1, s_3, s_2\}, \{v\}\}$
$X_2 = \{\{s_1, s_3\}, \{s_2\}, \{v\}\}$

7 Conclusion

We have developed and analysed algorithms for minimising probabilistic systems via approximate bisimulation. These algorithms are based on ϵ -quotients, a novel yet natural notion of approximate quotients. We have obtained theoretical bounds on the discrepancy between the minimised and the non-minimised systems. In our experiments, approximate partition refinement does well in minimising labelled Markov chains with perturbed transition probabilities, suggesting that approximate partition refinement is a practical approach for “recognising” and exploiting approximate bisimulation.

Future work might consider the following questions: Does approximate minimisation allow for further forms of active learning? Can our techniques be transferred to Markov decision processes?

References

- 1 Giovanni Bacci, Giorgio Bacci, Kim G. Larsen, and Radu Mardare. On the metric-based approximate minimization of Markov chains. *J. Log. Algebraic Methods Program.*, 100:36–56, 2018. doi:10.1016/j.jlamp.2018.05.006.
- 2 Hugo Bazille, Blaise Genest, Cyrille Jégourel, and Jun Sun. Global PAC bounds for learning discrete time Markov chains. In Shuvendu K. Lahiri and Chao Wang, editors, *Computer Aided Verification - 32nd International Conference, CAV 2020, Los Angeles, CA, USA, July 21-24, 2020, Proceedings, Part II*, volume 12225 of *Lecture Notes in Computer Science*, pages 304–326. Springer, 2020. doi:10.1007/978-3-030-53291-8_17.
- 3 Gaoang Bian and Alessandro Abate. On the relationship between bisimulation and trace equivalence in an approximate probabilistic context. In Javier Esparza and Andrzej S. Murawski, editors, *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10203 of *Lecture Notes in Computer Science*, pages 321–337, 2017. doi:10.1007/978-3-662-54458-7_19.
- 4 Patrick Billingsley. *Probability and measure*. Wiley Series in Probability and Statistics. Wiley, New York, NY, USA, 3rd edition, 1995.
- 5 Di Chen, Franck van Breugel, and James Worrell. On the complexity of computing probabilistic bisimilarity. In Lars Birkedal, editor, *Foundations of Software Science and Computational Structures - 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24 - April 1, 2012. Proceedings*, volume 7213 of *Lecture Notes in Computer Science*, pages 437–451. Springer, 2012. doi:10.1007/978-3-642-28729-9_29.
- 6 Jianhua Chen. Properties of a new adaptive sampling method with applications to scalable learning. *Web Intelligence*, 13(4):215–227, 2015. doi:10.3233/WEB-150322.
- 7 P. D’Argenio, B. Jeannot, H. Jensen, and K. Larsen. Reachability analysis of probabilistic systems by successive refinements. In L. de Alfaro and S. Gilmore, editors, *Proc. 1st Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modelling and Verification (PAPM/PROBMIV’01)*, volume 2165 of *LNCS*, pages 39–56. Springer, 2001.
- 8 Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labeled Markov systems. In Jos Baeten and Sjouke Mauw, editors, *Proceedings of the 10th International Conference on Concurrency Theory*, volume 1664 of *Lecture Notes in Computer Science*, pages 258–273, Eindhoven, The Netherlands, August 1999. Springer-Verlag.
- 9 Josée Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov processes. *Theor. Comput. Sci.*, 318(3):323–354, 2004. doi:10.1016/j.tcs.2003.09.013.
- 10 Josée Desharnais, François Laviolette, and Mathieu Tracol. Approximate analysis of probabilistic processes: Logic, simulation and games. In *Fifth International Conference on the Quantitative Evaluation of Systems (QEST 2008), 14-17 September 2008, Saint-Malo, France*, pages 264–273. IEEE Computer Society, 2008. doi:10.1109/QEST.2008.42.
- 11 S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.
- 12 Christian Hensel, Sebastian Junges, Joost-Pieter Katoen, Tim Quatmann, and Matthias Volk. The probabilistic model checker Storm. *CoRR*, abs/2002.07080, 2020. arXiv:2002.07080.
- 13 T. Herman. Probabilistic self-stabilization. *Information Processing Letters*, 35(2):63–67, 1990.
- 14 Aron Itai and Michael Rodeh. Symmetry breaking in distributed networks. *Information and Computation*, 88(1):60–87, September 1990.
- 15 Stefan Kiefer and Qiyi Tang. Approximate bisimulation minimisation, 2021. arXiv:2110.00326.

- 16 Marta Kwiatkowska, Gethin Norman, and David Parker. Prism 4.0: Verification of probabilistic real-time systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Proceedings of the 23rd International Conference on Computer Aided Verification*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591, Snowbird, UT, USA, July 2011. Springer-Verlag. doi:10.1007/978-3-642-22110-1_47.
- 17 M. Reiter and A. Rubin. Crowds: Anonymity for web transactions. *ACM Transactions on Information and System Security (TISSEC)*, 1(1):66–92, 1998.
- 18 Qiyi Tang and Franck van Breugel. Deciding probabilistic bisimilarity distance one for labelled markov chains. In Hana Chockler and Georg Weissenbacher, editors, *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I*, volume 10981 of *Lecture Notes in Computer Science*, pages 681–699. Springer, 2018.
- 19 Mathieu Tracol, Joséé Desharnais, and Abir Zhioua. Computing distances between probabilistic automata. In Mieke Massink and Gethin Norman, editors, *Proceedings Ninth Workshop on Quantitative Aspects of Programming Languages, QAPL 2011, Saarbrücken, Germany, April 1-3, 2011*, volume 57 of *EPTCS*, pages 148–162, 2011. doi:10.4204/EPTCS.57.11.