

# Computing Probabilistic Bisimilarity Distances via Policy Iteration\*

Qiyi Tang<sup>1</sup> and Franck van Breugel<sup>2</sup>

1 DisCoVeri Group, Department of Electrical Engineering and Computer Science  
York University, Toronto, Canada

2 DisCoVeri Group, Department of Electrical Engineering and Computer Science  
York University, Toronto, Canada

---

## Abstract

A transformation mapping a labelled Markov chain to a simple stochastic game is presented. In the resulting simple stochastic game, each vertex corresponds to a pair of states of the labelled Markov chain. The value of a vertex of the simple stochastic game is shown to be equal to the probabilistic bisimilarity distance, a notion due to Desharnais, Gupta, Jagadeesan and Panangaden, of the corresponding pair of states of the labelled Markov chain. Bacci, Bacci, Larsen and Mardare introduced an algorithm to compute the probabilistic bisimilarity distances for a labelled Markov chain. A modification of a basic version of their algorithm for a labelled Markov chain is shown to be the policy iteration algorithm applied to the corresponding simple stochastic game. Furthermore, it is shown that this algorithm takes exponential time in the worst case.

**1998 ACM Subject Classification** D.2.4 Software/Program Verification, F.1.1 Models of Computation, G.3 Probability and Statistics

**Keywords and phrases** labelled Markov chain, simple stochastic game, probabilistic bisimilarity, pseudometric, value function, policy iteration

**Digital Object Identifier** 10.4230/LIPIcs.CONCUR.2016.22

## 1 Introduction

A *behavioural equivalence* answers the fundamental question “when are two states of a model considered behaviourally the same?” The most prominent behavioural equivalence is *bisimilarity*, due to Milner [21] and Park [24]. We refer the reader to, for example, [25, page 1–4], for an extensive discussion of the importance of behavioural equivalences such as bisimilarity.

A *behavioural pseudometric* is a quantitative generalization of a behavioural equivalence. Such a pseudometric assigns to each pair of states a number in the unit interval  $[0, 1]$ . The smaller this number, the more alike the states behave. Those states that have distance zero are considered behaviourally equivalent. As first observed by Giacalone, Jou and Smolka [14], behavioural equivalences are *not robust* for models that include quantitative information such as time and probabilities. For these models, behavioural pseudometrics are an essential complement to behavioural equivalences. For a more detailed discussion of the merits of behavioural pseudometrics, we refer the reader to, for example, [23, Chapter 8].

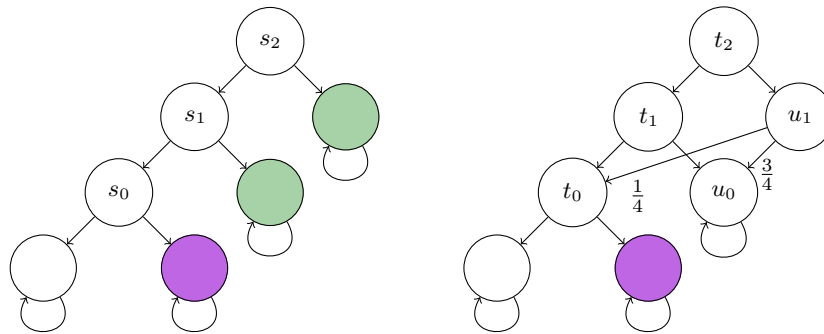
---

\* This research has been supported by a grant of the Natural Sciences and Engineering Research Council of Canada.



## 22:2 Computing Probabilistic Bisimilarity Distances

Systems with probabilistic behaviour are often modelled as *labelled Markov chains*. An example of such a Markov chain is depicted below. In a labelled Markov chain, each state has a label. In the example, the label is represented by the colour of the state. These labels are used to capture that particular properties of interest hold in some states and do not hold in other states. In diagrams, like the one below, only if the probabilities of the outgoing transitions of a state are not all the same, as is the case for state  $u_1$ , we denote the actual probabilities.



The most prominent behavioural equivalence for labelled Markov chains is *probabilistic bisimilarity*, due to Larsen and Skou [19]. Numerous quantitative generalizations of this behavioural equivalence have been proposed, the *probabilistic bisimilarity pseudometric* due to Desharnais et al. [12] being the most notable one. In this paper, we focus on this probabilistic bisimilarity pseudometric.

In order to exploit behavioural pseudometrics such as the probabilistic bisimilarity pseudometric, it is essential to be able to approximate or compute these behavioural distances. The first algorithm to approximate these distances was presented by Van Breugel, Sharma and Worrell in [5]. In their algorithm, the distance between states  $s$  and  $t$ , denoted  $\delta(s, t)$ , is computed as follows. Since  $\delta(s, t) < q$ , for some rational  $q$ , can be expressed in the existential fragment of the first order theory over the reals as shown by Van Breugel et al., and this theory is decidable as shown by Tarski [28], one can use binary search to approximate  $\delta(s, t)$ . The satisfiability problem for the existential fragment of the first order theory over the reals can be solved in polynomial space [7]. The algorithm of Van Breugel et al. can only handle labelled Markov chains with a handful of states.

Subsequently, Chen, Van Breugel and Worrell [8] presented a polynomial time algorithm to compute the distances. They showed that the distances are rational and that those distances can be computed by means of Khachiyan’s ellipsoid method [18]. In particular, they showed that the distance function can be expressed as the solution of a linear program. In this case, the separation algorithm, which is an integral part of the ellipsoid method, boils down to solving a minimum cost flow problem. The network simplex algorithm solves the latter problem in polynomial time [22]. In practice, it is several orders of magnitude slower than the algorithm of Bacci et al. which we will discuss next.

Bacci, Bacci, Larsen and Mardare [2] put forward yet another algorithm to compute the bisimilarity distances. In their paper, they showed that their algorithm, in contrast to the two algorithms mentioned above, can handle labelled Markov chains of 50 states. Their algorithm can be viewed as a basic algorithm, enhanced with an optimization. The key idea behind this optimization is to compute the distances “on-the-fly.” Roughly speaking, to compute  $\delta(s, t)$  we only need to compute  $\delta(u, v)$  where  $s$  and  $t$  can reach  $u$  and  $v$  in  $n$  transitions for some  $n > 0$ . In this paper, we will not consider this optimization, but focus on the basic algorithm only.

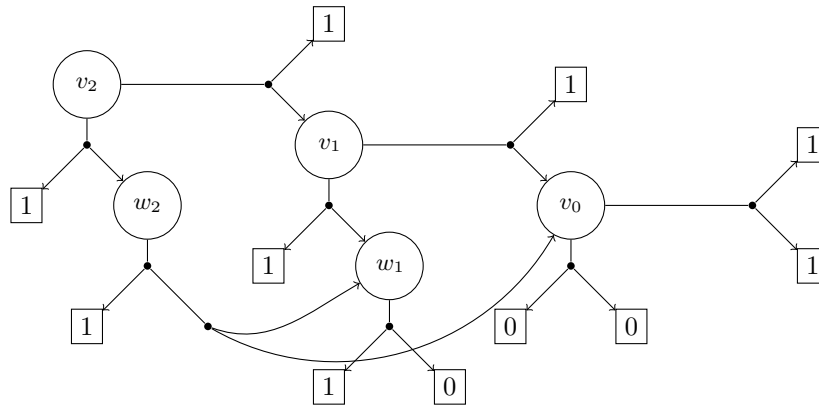
Stochastic games were introduced by Shapley [26]. A simplified version of these games, called *simple stochastic games*, were studied by Condon [9]. A simple stochastic game is played with a single token by two players, called min and max, on a finite directed graph. The graph has five types of vertices: min, max and average vertices, 0-sinks and 1-sinks. The min, max and average vertices have two outgoing edges, whereas the 0-sinks and 1-sinks have no outgoing edges. Whenever the token is in a min (max) vertex, the token is moved to one of the two successors of the vertex, chosen by the min (max) player. If the token is in an average vertex, the successor is chosen randomly. The min (max) player's objective is to minimize (maximize) the probability of reaching a 1-sink.

In its strategy, the min player chooses for each min vertex one of its two successors. Similarly, a strategy for the max player assigns to each max vertex one of its successors. Strategies are also known as *policies*. Given a strategy for the min player and a strategy for the max player, we can define a function that maps the vertices to the interval  $[0, 1]$ . This function maps each vertex  $v$  to the probability that the max player wins the game, provided that the game starts in vertex  $v$  and the min and max player play according to their strategies. When the strategy of both players is optimal, this function is called the *value function*.

A variety of algorithms has been developed to compute the value function of a simple stochastic game. Several of these algorithms use *policy iteration*. As long as there exists a choice in the strategy of the min or max player that is not locally optimal, switch that choice for one that is locally optimal. Hoffman and Karp [15] introduced a policy iteration algorithm for stochastic games in which all non-optimal choices are switched in each iteration. Condon [10] presented a similar algorithm, known as *simple policy iteration*, that switches only one non-optimal choice per iteration.

Most of the main results of this paper rely on a transformation mapping each labelled Markov chain to a simple stochastic game. The resulting simple stochastic game does not have any max vertices. Each min vertex  $v$  of the simple stochastic game corresponds to a pair  $(s, t)$  of states of the labelled Markov chain. In Section 4, we will show that the value of  $v$  is the distance of  $s$  and  $t$ . In [6], Van Breugel and Worrell present a similar transformation. They map a probabilistic automaton, which is a more general model as it includes not only probabilistic choices but also nondeterministic choices, to a simple stochastic game. They also show that values and distances correspond, and use this correspondence to prove a complexity result for computing behavioural distances for probabilistic automata, but they do not present any algorithm.

Below we present (part of) the simple stochastic game corresponding to the labelled Markov chain presented earlier in this introduction. The min vertices  $v_i$  correspond to the state pairs  $(s_i, t_i)$  and the max vertices  $w_i$  correspond to the state pairs  $(s_{i-1}, u_{i-1})$ . The average vertices are denoted by bullets. The 0- and 1-sinks are labelled with zeroes and ones.



In Section 5, we will prove that a small modification of the basic algorithm of Bacci et al. corresponds to Condon’s simple policy iteration. That is, the modified basic algorithm for computing the distances of all state pairs of a labelled Markov chain can be viewed as simple policy iteration applied to the corresponding simple stochastic game. As a consequence, the proof by Condon [9, Lemma 4] that simple policy iteration computes the value function of a simple stochastic game also shows that the modified basic algorithm computes the distances.

Let us highlight a technical detail here. Condon’s proof that simple policy iteration computes the value function of a simple stochastic game relies on the assumption that the game halts with probability one. That is, no matter which strategy the min and max player use, the game reaches a 0- or 1-sink with probability one. To be able to use Condon’s proof in our setting, we need to show that the simple stochastic game resulting from the labelled Markov chain halts with probability one. As we will see, this is accomplished by mapping state pairs  $(s, t)$  of the labelled Markov chain, for which  $s$  and  $t$  are probabilistic bisimilar, to a 0-sink in the simple stochastic game. Hence, before running the basic algorithm of Bacci et al., we first need to decide which states are probabilistic bisimilar, which can be done in polynomial time [3]. This is the small, yet essential, modification of their algorithm, to which we alluded earlier.

In Section 6, we will show that in the worst case, our algorithm takes exponential time. Many similar lower bounds have been proved for closely related algorithms by showing that the algorithms can be viewed as binary counters. We refer the reader to, for example, the thesis of Friedmann [13] for several such proofs. For simple stochastic games, Melekopoglou and Condon [20] showed that simple policy iteration takes exponential time in the worst case. We cannot directly use their result since no labelled Markov chain maps to the simple stochastic games they use in their proof. As we mentioned before, the labelled Markov chain depicted earlier gives rise to the simple stochastic game (of which a part is) depicted above. That simple stochastic game implements a 3-bit counter, as we will discuss next.

For the above simple stochastic game, a strategy of the min player consists of either going to the right (represented by 0) or down (represented by 1) in the vertices  $v_2, v_1$  and  $v_0$ . In the table below, we present for each strategy the values of the vertices  $v_2, v_1$  and  $v_0$ . Going from one column to the next, the strategy for either  $v_2, v_1$  or  $v_0$  is switched. As a result, none of the values increase and one of the values decreases. The table contains all eight 3-bit combinations and, hence, the simple stochastic game can be viewed as a 3-bit counter. As we will show in Section 6, for each  $n \in \mathbb{N}$  we can construct a labelled Markov chain of size  $O(n)$  that gives rise to a simple stochastic game that implements an  $n$ -bit counter. Hence, from a theoretical point of view, the algorithm of Bacci et al. is inferior to the algorithm of Chen et al.

strategy	$v_2$	0	1	1	0	0	1	1	0
	$v_1$	0	0	1	1	1	1	0	0
	$v_0$	0	0	0	0	1	1	1	1
value	$v_2$	1	$\frac{15}{16}$	$\frac{15}{16}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{13}{16}$	$\frac{13}{16}$	$\frac{3}{4}$
	$v_1$	1	1	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{2}$
	$v_0$	1	1	1	1	0	0	0	0

As Bacci et al. already showed in [2], in practice their algorithm outperforms the other algorithms to compute bisimilarity distances. Recall that we only consider their basic algorithm. We have shown that our modification of this algorithm also performs very well in practice. We ran the algorithm on a variety of labelled Markov chains obtained from examples of probabilistic model checkers such as PRISM<sup>1</sup> and MRMC<sup>2</sup>. The performance of our modified algorithm, which has to decide probabilistic bisimilarity first, is comparable to the performance of their basic algorithm. We expect the same results by adapting our basic algorithm to the on-the-fly setting.

## 2 The Probabilistic Bisimilarity Pseudometric

In this section, we review the model of interest, labelled Markov chains, and the probabilistic bisimilarity pseudometric due to Desharnais et al. [12]. We denote the set of probability distributions on a set  $S$  by  $Dist(S)$ .

► **Definition 1.** A labelled Markov chain is a tuple  $\langle S, L, \tau, \ell \rangle$  consisting of

- a set  $S$  of states,
- a set  $L$  of labels,
- a transition function  $\tau : S \rightarrow Dist(S)$ ,
- a labelling function  $\ell : S \rightarrow L$ .

We restrict our attention to labelled Markov chains with finitely many states and the transition probabilities of which are rationals. For the remainder of this section, we fix such a labelled Markov chain  $\langle S, L, \tau, \ell \rangle$ .

► **Definition 2.** Let  $\mu, \nu \in Dist(S)$ . The set  $\Omega(\mu, \nu)$  of couplings of  $\mu$  and  $\nu$  is defined by

$$\Omega(\mu, \nu) = \left\{ \omega \in Dist(S \times S) \mid \sum_{t \in S} \omega(s, t) = \mu(s) \wedge \sum_{s \in S} \omega(s, t) = \nu(t) \right\}.$$

Note that  $\omega \in \Omega(\mu, \nu)$  is a joint probability distribution with marginals  $\mu$  and  $\nu$ .

► **Definition 3.** The function  $\mathcal{T} : Dist(S) \times Dist(S) \times [0, 1]^{S \times S} \rightarrow [0, 1]$  is defined by

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

<sup>1</sup> [www.prismmodelchecker.org](http://www.prismmodelchecker.org)

<sup>2</sup> [www.mrmc-tool.org](http://www.mrmc-tool.org)

## 22:6 Computing Probabilistic Bisimilarity Distances

The function  $\mathcal{T}$  can be viewed as the minimal cost of a transportation problem. Consider two disjoint copies of  $S$ , one representing sources and the other representing targets. For each  $s, t \in S$ ,  $\mu(s)$  represents the supply at source  $s$  and  $\nu(t)$  represents the demand at target  $t$ . The cost of transporting one unit from source  $s$  to target  $t$  is captured by  $d(s, t)$ . The amount transported from source  $s$  to target  $t$  is captured by  $\omega(s, t)$ . The distance function  $\mathcal{T}(\mu, \nu, d)$  is known as the Kantorovich metric [17].

► **Definition 4.** The function  $\Delta : [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$  is defined by

$$\Delta(d)(s, t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \mathcal{T}(\tau(s), \tau(t), d) & \text{otherwise} \end{cases}$$

To define the behavioural pseudometric, we use the Knaster-Tarski fixed point theorem (see, for example, [11, Chapter 2]). To apply that theorem, we need to define an order on  $[0, 1]^{S \times S}$ . For  $d, e \in [0, 1]^{S \times S}$  we write  $d \sqsubseteq e$  if  $d(s, t) \leq e(s, t)$  for all  $s, t \in S$ . The set  $[0, 1]^{S \times S}$  endowed with the order  $\sqsubseteq$  forms a complete lattice. Since  $\Delta$  is a monotone function, we can conclude from the Knaster-Tarski fixed point theorem that  $\Delta$  has a least fixed point. We denote this fixed point by  $\delta$ . This is the probabilistic bisimilarity pseudometric of Desharnais et al.

### 3 An Alternative Characterization of $\delta$

Next, we provide an alternative characterization of the probabilistic bisimilarity pseudometric  $\delta$  which can be found in the extended version of [8]. This characterization can also be found in the work of Bacci et al. [2, Theorem 9]. It provides the basis for their algorithm. The characterization relies on couplings, a notion from the theory of Markov chains.

► **Definition 5.** The set  $\mathcal{C}$  of couplings of the labelled Markov chain  $\langle S, L, \tau, \ell \rangle$  is defined by

$$\mathcal{C} = \{ T \in \text{Dist}(S \times S)^E \mid \forall (s, t) \in E : T(s, t) \in \Omega(\tau(s), \tau(t)) \}$$

where  $E = \{ (s, t) \in S \times S \mid \ell(s) = \ell(t) \}$ .

For the remainder of this section, we fix a labelled Markov chain  $\langle S, L, \tau, \ell \rangle$ . We also fix a coupling  $T \in \mathcal{C}$  of the labelled Markov chain. Note that  $\langle S \times S, T \rangle$  can be viewed as a Markov chain, where the states  $(s, t) \notin E$  are absorbing.

► **Definition 6.** The function  $\Gamma^T : [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$  is defined by

$$\Gamma^T(d)(s, t) = \begin{cases} 1 & \text{if } \ell(s) \neq \ell(t) \\ \sum_{u, v \in S} T(s, t)(u, v) d(u, v) & \text{otherwise} \end{cases}$$

Since  $[0, 1]^{S \times S}$  is a complete lattice and  $\Gamma^T$  is a monotone function, we can conclude from the Knaster-Tarski fixed point theorem that  $\Gamma^T$  has a least fixed point. We denote this fixed point by  $\gamma^T$ . Note that  $\gamma^T(s, t)$  is the probability of reaching a state  $(u, v) \notin E$  from the state  $(s, t)$  in the Markov chain  $\langle S \times S, T \rangle$ .

For all  $\mu, \nu \in \text{Dist}(S)$ , we denote the set of vertices of the convex polytope  $\Omega(\mu, \nu)$  by  $V(\Omega(\mu, \nu))$ . We define

$$V(\mathcal{C}) = \{ T \in \text{Dist}(S \times S)^E \mid \forall (s, t) \in E : T(s, t) \in V(\Omega(\tau(s), \tau(t))) \}.$$

The probabilistic bisimilarity pseudometric  $\delta$  can be characterized as follows.

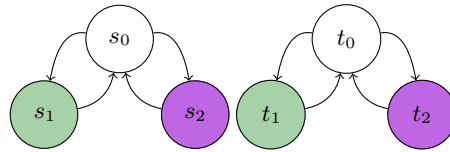
► **Theorem 7.**  $\delta = \min_{T \in V(\mathcal{C})} \gamma^T$ .

The basic algorithm of Bacci et al. relies on the fact that if a coupling  $T$  is locally optimal, that is,  $\Delta(\gamma^T) = \gamma^T$ , then it is globally optimal as well, that is,  $\gamma^T = \delta$  (see [2, Lemma 18]). However, as we will show next, this is not the case in general.

We denote the Dirac distribution concentrated at the pair of states  $(s, t)$  by  $\delta_{(s,t)}$ , that is,  $\delta_{(s,t)}(u, v) = 1$  if  $s = u, t = v$  and  $\delta_{(s,t)}(u, v) = 0$  otherwise.

► **Theorem 8.** *There exists a labelled Markov chain and  $T \in V(\mathcal{C})$  such that  $\Delta(\gamma^T) = \gamma^T$  and  $\gamma^T \neq \delta$ .*

**Proof.** Consider the following labelled Markov chain.



Note that

$$V(\Omega(\tau(s_0), \tau(t_0))) = \left\{ \frac{1}{2}\delta_{(s_1, t_1)} + \frac{1}{2}\delta_{(s_2, t_2)}, \frac{1}{2}\delta_{(s_1, t_2)} + \frac{1}{2}\delta_{(s_2, t_1)} \right\}$$

$$V(\Omega(\tau(s_1), \tau(t_1))) = \{ \delta_{(s_0, t_0)} \} \text{ and } V(\Omega(\tau(s_2), \tau(t_2))) = \{ \delta_{(s_0, t_0)} \}$$

Now take  $T, U \in V(\mathcal{C})$  such that

$$T(s_0, t_0) = \frac{1}{2}\delta_{(s_1, t_2)} + \frac{1}{2}\delta_{(s_2, t_1)} \text{ and } U(s_0, t_0) = \frac{1}{2}\delta_{(s_1, t_1)} + \frac{1}{2}\delta_{(s_2, t_2)}$$

Then we have

$$\gamma^T(s_0, t_0) = \gamma^T(s_1, t_1) = \gamma^T(s_2, t_2) = \gamma^T(s_1, t_2) = \gamma^T(s_2, t_1) = 1$$

$$\gamma^U(s_0, t_0) = 0$$

Furthermore,

$$\Delta(\gamma^T)(s_0, t_0) = \min\left\{ \frac{1}{2}\gamma^T(s_1, t_2) + \frac{1}{2}\gamma^T(s_2, t_1), \frac{1}{2}\gamma^T(s_1, t_1) + \frac{1}{2}\gamma^T(s_2, t_2) \right\} = 1$$

$$\Delta(\gamma^T)(s_1, t_1) = \gamma^T(s_0, t_0) = 1$$

$$\Delta(\gamma^T)(s_2, t_2) = \gamma^T(s_0, t_0) = 1$$

◀

## 4 Simple Stochastic Games

In this section, we present the transformation that maps each labelled Markov chain to a simple stochastic game such that distances correspond to values. Let us first formally define simple stochastic games.

► **Definition 9.** A simple stochastic game is a tuple  $(V, E, \pi)$  consisting of

- a finite directed graph  $(V, E)$  such that
  - $V$  is partitioned into the sets
    - \*  $V_{\min}$  of min vertices,
    - \*  $V_{\max}$  of max vertices,
    - \*  $V_{\text{rnd}}$  of random vertices,
    - \*  $V_0$  of 0-sinks, and
    - \*  $V_1$  of 1-sinks,

## 22:8 Computing Probabilistic Bisimilarity Distances

- the vertices in  $V_0$  and  $V_1$  have outdegree zero and all other vertices have outdegree at least one,
- a function  $\pi : V_{\text{rnd}} \rightarrow \text{Dist}(V)$  such that for each vertex  $v \in V_{\text{rnd}}$ ,  $\pi(v)(w) > 0$  iff  $(v, w) \in E$ .

In this paper, we consider a slightly more general definition than the one given by Condon in [9]. In particular, the outdegree of min, max and random vertices is at least one (instead of exactly two), there may be multiple 0-sinks and 1-sinks (rather than exactly one), and the outgoing edges of a random vertex are labelled with rationals (rather than  $\frac{1}{2}$ ). However, a simple stochastic game as defined above can be transformed in polynomial time into a simple stochastic game as defined in [9], as shown by Zwick and Paterson [29, page 355].

In the construction below, we need the notion of probabilistic bisimilarity. This notion can be captured as follows.

► **Definition 10.** An equivalence relation  $R \subseteq S \times S$  is a probabilistic bisimulation if for all  $(s, t) \in R$ ,  $\ell(s) = \ell(t)$  and there exists  $\omega \in \Omega(\tau(s), \tau(t))$  such that  $\text{support}(\omega) \subseteq R$ , where  $\text{support}(\omega) = \{(u, v) \in S \times S \mid \omega(u, v) > 0\}$ . Probabilistic bisimilarity, denoted  $\sim$ , is defined as the largest probabilistic bisimulation.

In [16, Theorem 4.6] it is shown that the above characterization of probabilistic bisimilarity is equivalent to the standard definition given in [19]. Now, we are ready to introduce the transformation that maps each labelled Markov chain to a simple stochastic game.

► **Definition 11.** Let  $\langle S, L, \tau, \ell \rangle$  be a labelled Markov chain. The simple stochastic game  $\langle V, E, \pi \rangle$  is defined by

- $V_{\text{min}} = \{(s, t) \in S \times S \mid s \not\sim t \wedge \ell(s) = \ell(t)\}$ ,
- $V_{\text{max}} = \emptyset$ ,
- $V_{\text{rnd}} = \bigcup \{V(\Omega(\tau(s), \tau(t))) \mid s, t \in S \wedge s \not\sim t \wedge \ell(s) = \ell(t)\}$ ,
- $V_0 = \{(s, t) \in S \times S \mid s \sim t\}$ ,
- $V_1 = \{(s, t) \in S \times S \mid \ell(s) \neq \ell(t)\}$ ,
- $E = \{\langle (s, t), \omega \rangle \mid (s, t) \in V_{\text{min}} \wedge \omega \in V(\Omega(\tau(s), \tau(t)))\} \cup \{\langle \omega, (u, v) \rangle \mid \omega(u, v) > 0\}$ , and
- $\pi(\omega)(u, v) = \omega(u, v)$ .

Note that the resulting simple stochastic game does not have any max vertices.<sup>3</sup> As a result, the max player never gets to move the token. Therefore, there is no need for a strategy of the max player and, hence, we only need to consider the strategy of the min player. Recall that a strategy of the min player maps each min vertex to one of its successors. That is, such a strategy maps  $(s, t)$ , with  $s \not\sim t$  and  $\ell(s) = \ell(t)$ , to  $\omega \in V(\Omega(\tau(s), \tau(t)))$ . Hence, we can view  $T \in V(\mathcal{C})$  as such a strategy by ignoring its values for  $(s, t)$  with  $s \sim t$ .

For the remainder of this section, we fix a labelled Markov chain  $\langle S, L, \tau, \ell \rangle$  and a coupling  $T \in V(\mathcal{C})$ . The following definition of  $\Theta^T$  is very similar to the definition of  $\Gamma^T$  given in Definition 6.

---

<sup>3</sup> The simple stochastic game of Definition 11 is a special type of simple stochastic game since it does not have any max vertices. It can also be viewed as a Markov decision process. Note, though, that it is also a special type of the Markov decision process since the reward function maps all transitions to zero apart from those that reach a 1-sink which it maps to one.



► **Definition 12.** The function  $\Theta^T : [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$  is defined by

$$\Theta^T(d)(s, t) = \begin{cases} 0 & \text{if } s \sim t \\ 1 & \text{if } \ell(s) \neq \ell(t) \\ \sum_{u, v \in S} T(s, t)(u, v) d(u, v) & \text{otherwise} \end{cases}$$

Since  $[0, 1]^{S \times S}$  is a complete lattice and  $\Theta^T$  is a monotone function, we can conclude from the Knaster-Tarski fixed point theorem that  $\Theta^T$  has a least fixed point. We denote this fixed point by  $\theta^T$ .

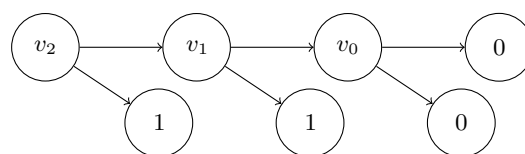
Note that  $\theta^T(s, t)$  is the value of the min vertex  $(s, t)$  with respect to the strategy  $T$ . As we will show next, the optimal strategy gives rise to the probabilistic bisimilarity pseudometric  $\delta$ , that is, the probabilistic bisimilarity distances of the states  $s$  and  $t$  of the labelled Markov chain are the values of the vertices  $(s, t)$  of the corresponding simple stochastic game.

► **Theorem 13.**  $\delta = \min_{T \in V(\mathcal{C})} \theta^T$ .

## 5 Simple Policy Iteration

As we already mentioned in the introduction, Condon’s simple policy iteration algorithm computes the values of a simple stochastic game, provided that the simple stochastic game halts with probability one. As we have shown in the previous section, the probabilistic bisimilarity distances of a labelled Markov chain are the values of the corresponding simple stochastic game defined in Definition 11. Hence, if that simple stochastic game halts with probability one, then we can use simple policy iteration to compute the probabilistic bisimilarity distances.

A simple stochastic game and a pair of strategies for the min and max player give naturally rise to a Markov chain, where the vertices of the simple stochastic game are the states of the Markov chain (see, for example, [9, Section 2.1] for details). We will call this the coupled Markov chain. For example, the coupled Markov chain induced by the simple stochastic game presented in the introduction, where the strategy for the min player (the values of the vertices  $v_2, v_1$  and  $v_0$ ) is 001, is given below.



A simple stochastic game halts with probability one if for each pair of strategies, in the coupled Markov chain each state reaches a 0-sink or 1-sink with probability one.

► **Theorem 14.** *The simple stochastic game defined in Definition 11 halts with probability one.*

**Proof.** Towards a contradiction, assume that the simple stochastic game does not halt with probability one. Then there exists a strategy for the min player, that is, an element  $T \in V(\mathcal{C})$ , and a vertex  $(s, t)$  which does not reach a 0- or 1-sink with probability one in the coupled Markov chain. Each state in a Markov chain reaches with probability one a closed communication class, also known as a bottom strongly connected component (see, for example, [4, Theorem 10.27]). Note that a 0-sink and a 1-sink each forms a closed

## 22:10 Computing Probabilistic Bisimilarity Distances

communication class. Since  $(s, t)$  does not reach a 0- or 1-sink with probability one,  $(s, t)$  reaches a closed communication class  $C$  not consisting of a 0- or 1-sink. We can also show that for each  $(s, t) \in C$ , we have  $s \sim t$ . However, in that case,  $(s, t)$  is a 0-sink, which contradicts the fact that  $C$  does not consist of a 0- or 1-sink. ◀

Simple policy iteration starts with an arbitrary strategy, that is, an arbitrary  $T \in V(\mathcal{C})$  (see line 1). As long as there is a min vertex which is not locally optimal with respect to the current strategy, the strategy at that vertex is improved to the locally optimal choice. Note that a min vertex  $(s, t)$  is not locally optimal if there exists a different choice for that vertex, that is,  $\omega \in V(\Omega(\tau(s), \tau(t)))$ , so that the value of the vertex decreases. This is captured in line 2. In line 3, we compute a locally optimal choice and update the strategy.

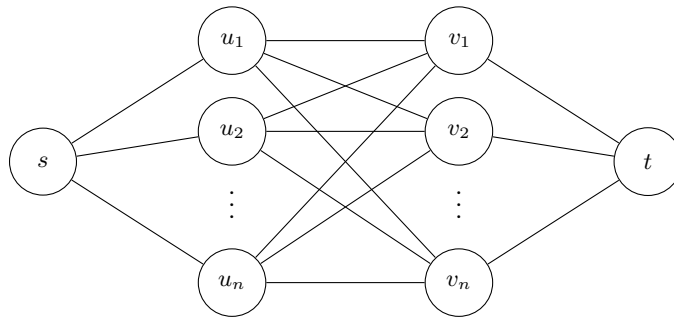
```

1  $T \leftarrow$  an element of  $V(\mathcal{C})$ 
2 while  $\exists (s, t) \in V_{\min} : \theta^T(s, t) > \Delta(\theta^T)(s, t)$ 
3    $T(s, t) \leftarrow \arg \min_{\omega \in V(\Omega(\tau(s), \tau(t)))} \sum_{u, v \in S} \omega(u, v) \theta^T(u, v)$ 

```

This is our modification of the basic algorithm of Bacci et al.

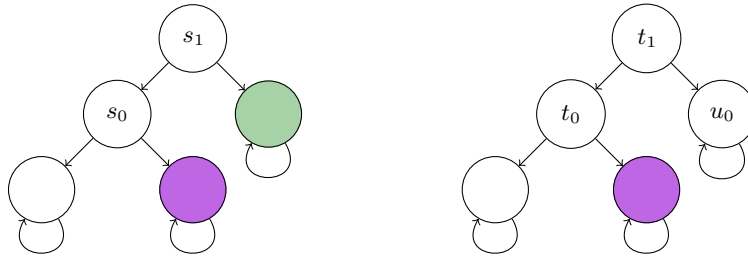
In line 1, an initial strategy  $T \in V(\mathcal{C})$  can be computed by the North-West corner method in polynomial time (see, for example, [27, page 180]). In line 2, rather than choosing an arbitrary min vertex that is not locally optimal, in the simple policy iteration a select procedure can be defined as follows: we number all the vertices in  $V_{\min}$  and select the one with the highest number [20, Section 2]. Note that  $\theta^T$  can be computed in polynomial time [4, Section 10.1.1]. In line 3, the computation can be viewed as a minimum-cost flow problem, where  $s$  is the source vertex and  $t$  is the sink vertex. Below we present the flow network, the sets  $\{u_1, \dots, u_n\}$  and  $\{v_1, \dots, v_n\}$  are copies of  $S$ . For the edge  $(s, u_i)$  and  $(v_j, t)$ , the capacity is  $\tau(s)(u_i)$  and  $\tau(t)(v_j)$  respectively. There is no cost transporting along these edges. Each edge  $(u_i, v_j) \in S \times S$  has a capacity of  $\min(\tau(s)(u_i), \tau(t)(v_j))$  and  $\theta^T(u_i, v_j)$  is the cost of edge  $(u_i, v_j)$ . The minimum cost of transporting one unit from  $s$  to  $t$  is captured by  $\Delta(\theta^T)(s, t)$ , which can be solved using the network simplex algorithm and is strongly polynomial time [1, Section 11.8]. Note that each piece is polynomial so that the complexity of the algorithm is determined by the number of iterations, as we will see in the next section.



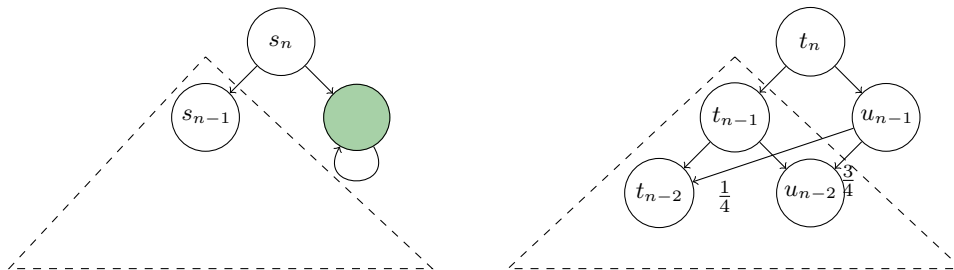
### 6 An Exponential Lower Bound

Below, we will prove an exponential lower bound for the algorithm we presented in the previous section. In particular, for each  $n \in \mathbb{N}$  we will construct a labelled Markov chain of size  $O(n)$ . Furthermore, we will show that our simple policy iteration algorithm takes  $\Omega(2^n)$  iterations for the resulting simple stochastic game.

► **Definition 15.** For  $n \in \mathbb{N}$ , the labelled Markov chain  $\mathcal{M}_n$  is defined as follows by induction on  $n$ . The labelled Markov chain  $\mathcal{M}_0$  is defined as



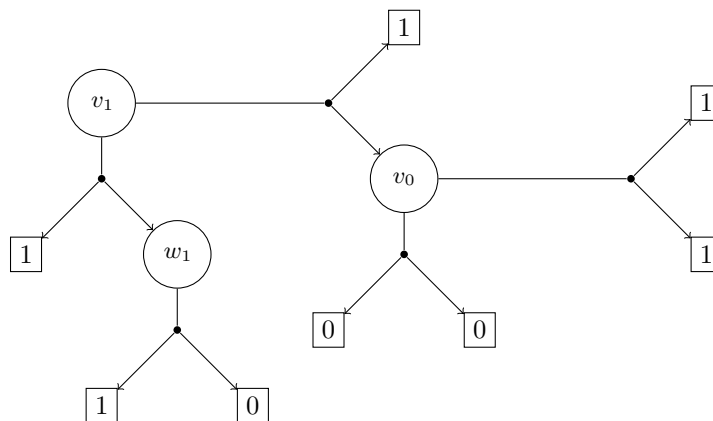
If  $n > 0$  then the labelled Markov chain  $\mathcal{M}_n$  is defined as



where the two dashed triangles together represent the labelled Markov chain  $\mathcal{M}_{n-1}$ .

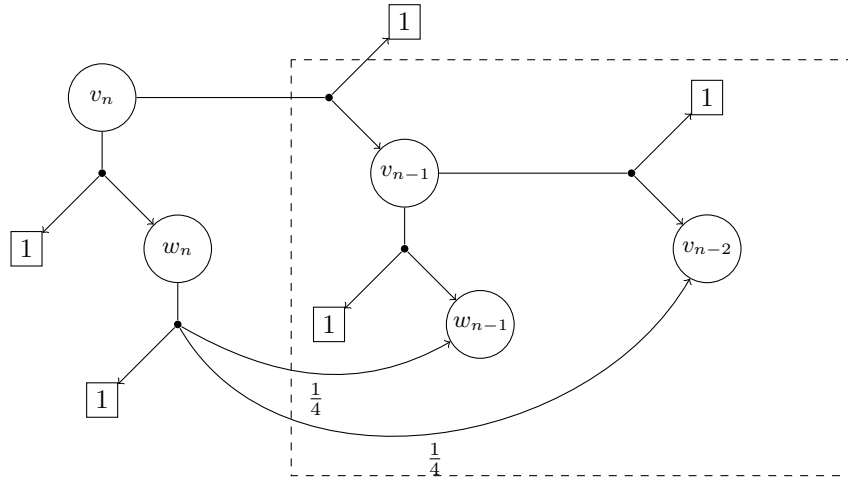
In the introductory section, we presented  $\mathcal{M}_1$ . Note that  $\mathcal{M}_n$  has  $4n + 10$  states and  $7n + 14$  transitions and, hence, is of size  $O(n)$ . Next, we give the simple stochastic games corresponding to the above defined labelled Markov chains according to transformation presented in Definition 11.

► **Definition 16.** For  $n \in \mathbb{N}$ , the simple stochastic game  $\mathcal{G}_n$  is defined as follows by induction on  $n$ . The simple stochastic game  $\mathcal{G}_0$  is defined as



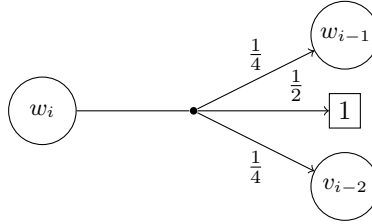
If  $n > 0$  then the simple stochastic game  $\mathcal{G}_n$  is defined as

22:12 Computing Probabilistic Bisimilarity Distances



where the dashed rectangle represents the simple stochastic game  $\mathcal{G}_{n-1}$ .

In the introductory section, we presented  $\mathcal{G}_1$ . In the above definition, we use  $v_i$  to denote the min node  $(s_i, t_i)$  and  $w_i$  to denote the min node  $(s_{i-1}, u_{i-1})$ . The transformation given in Definition 11 applied to labelled Markov chain  $\mathcal{M}_n$  gives rise to a simple stochastic game of which  $\mathcal{G}_n$  is only a part. In particular, for  $2 \leq i \leq n$  a random vertex and the following edges have not been included in  $\mathcal{G}_n$ , as they are never selected in any of the strategies we construct in our proofs.



Next, we consider the strategies for the simple stochastic game  $\mathcal{G}_n$ . Recall that the Dirac distribution  $\delta_v$  is defined by  $\delta_v(w) = 1$  if  $w = v$  and  $\delta_v(w) = 0$  otherwise. In order to avoid clutter, for  $1 \leq i \leq n$ , instead of  $T(v_i) = \frac{1}{2}\delta_1 + \frac{1}{2}\delta_{v_{i-1}}$  we write  $T(i) = 0$  and instead of  $T(v_i) = \frac{1}{2}\delta_1 + \frac{1}{2}\delta_{w_i}$  we write  $T(i) = 1$ . Also, instead of  $T(v_0) = \frac{1}{2}\delta_1 + \frac{1}{2}\delta_1$  we write  $T(0) = 0$  and instead of  $T(v_0) = \frac{1}{2}\delta_0 + \frac{1}{2}\delta_0$  we write  $T(0) = 1$ .

A vertex is switchable if it is not locally optimal.

► **Definition 17.** The vertex  $v_i$  is switchable with respect to  $T$  if  $\theta^T(v_i) > \theta^{\bar{T}^i}(v_i)$ , where

$$\bar{T}^i(j) = \begin{cases} 1 - T(j) & \text{if } j = i \\ T(j) & \text{otherwise} \end{cases}$$

Rather than starting from an arbitrary strategy, we pick a specific initial strategy (line 1–2). Furthermore, rather than choosing an arbitrary min vertex that is not locally optimal, we pick the  $v_i$  which is not locally optimal with the largest index (line 4).

```

1 for  $i = 0, \dots, n$ 
2    $T(i) \leftarrow 0$ 
3 while  $\exists 0 \leq i \leq n : v_i$  is switchable with respect to  $T$ 
4    $m \leftarrow \max\{i \mid v_i \text{ is switchable with respect to } T\}$ 
5    $T(m) \leftarrow 1 - T(m)$ 

```

The above simple policy iteration algorithm applied to the simple stochastic game  $\mathcal{G}_n$  gives rise to exponentially many iterations.

► **Theorem 18.** *For each  $n \in \mathbb{N}$ , there exists a labelled Markov chain of size  $O(n)$  such that simple policy iteration takes  $\Omega(2^n)$  iterations.*

## 7 Conclusion

Based on a correspondence between labelled Markov chains and simple stochastic games, we have shown that a modification of the basic algorithm of Bacci et al. for computing probabilistic bisimilarity distances of a labelled Markov chain is simple policy iteration applied to the corresponding simple stochastic game. The correspondence between labelled Markov chains and simple stochastic games also allows us to use Condon's correctness proof of simple policy iteration to show that our modification of the basic algorithm of Bacci et al. is correct. As we have shown in Theorem 8, that modification is essential. Such a modification also needs to be applied to the on-the-fly algorithm of Bacci et al.

Although Bacci et al. had already shown that their algorithm performs very well in practice, we show that in the worst case, their basic algorithm takes exponential time. Our example can also be used to show that their on-the-fly algorithm takes exponential time in the worst case as well. Our experimental results confirm that our modification gives rise to very little overhead. In several cases our modified algorithm even performs better than theirs. A detailed study of the performance of our modification of the on-the-fly algorithm is left for future research.

As we already mentioned in the introduction, the difference between Hoffman and Karp's policy iteration and Condon's simple policy iteration is that the former switches all locally non-optimal vertices in every iteration, whereas the latter only switches one of them. The question whether policy iteration, applied to a labelled Markov chain to compute its behavioural distances, may also give rise to exponentially many iterations is still open and left for future research. Note that an affirmative answer to this question would also prove an exponential lower bound for policy iteration for simple stochastic games, a problem that has been open for several decades.

**Acknowledgments.** The authors would like to thank Norm Ferns for extensive discussions. The authors are also grateful to the referees for their constructive feedback.

---

## References

- 1 Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Upper Saddle River, NJ, USA, 1993.
- 2 Giorgio Bacci, Giovanni Bacci, Kim Larsen, and Radu Mardare. On-the-fly exact computation of bisimilarity distances. In Nir Piterman and Scott Smolka, editors, *Proceedings of the 19th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, volume 7795 of *Lecture Notes in Computer Science*, pages 1–15, Rome, Italy, March 2013. Springer-Verlag.
- 3 Christel Baier. Polynomial time algorithms for testing probabilistic bisimulation and simulation. In Rajeev Alur and Thomas Henzinger, editors, *Proceedings of the 8th International Conference on Computer Aided Verification*, volume 1102 of *Lecture Notes in Computer Science*, pages 50–61, New Brunswick, NJ, USA, July/August 1996. Springer-Verlag.
- 4 Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. The MIT Press, Cambridge, MA, USA, 2008.

- 5 Franck van Breugel, Babita Sharma, and James Worrell. Approximating a behavioural pseudometric without discount. *Logical Methods in Computer Science*, 4(2), April 2008.
- 6 Franck van Breugel and James Worrell. The complexity of computing a bisimilarity pseudometric on probabilistic automata. In Franck van Breugel, Elham Kashefi, Catuscia Palamidessi, and Jan Rutten, editors, *Horizons of the Mind – A Tribute to Prakash Panangaden*, volume 8464 of *Lecture Notes in Computer Science*, pages 191–213. Springer-Verlag, 2014.
- 7 John Canny. Some algebraic and geometric computations in PSPACE. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 460–467, Chicago, IL, USA, May 1988. ACM.
- 8 Di Chen, Franck van Breugel, and James Worrell. On the complexity of computing probabilistic bisimilarity. In Lars Birkedal, editor, *Proceedings of the 15th International Conference on Foundations of Software Science and Computational Structures*, volume 7213 of *Lecture Notes in Computer Science*, pages 437–451, Tallinn, Estonia, March/April 2012. Springer-Verlag.
- 9 Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, February 1992.
- 10 Anne Condon. On algorithms for simple stochastic games. In Jin-Yi Cai, editor, *Advances in Computational Complexity Theory*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 51–73. American Mathematical Society, 1993.
- 11 Brian Davey and Hilary Priestley. *Introduction to Lattices and Order*. Cambridge University Press, Cambridge, United Kingdom, 2002.
- 12 Josee Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov processes. *Theoretical Computer Science*, 318(3):323–354, June 2004.
- 13 Oliver Friedmann. *Exponential Lower Bounds for Solving Infinitary Payoff Games and Linear Programs*. PhD thesis, Ludwig-Maximilians-University, Munich, Germany, 2011.
- 14 Alessandro Giacalone, Chi-Chang Jou, and Scott Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proceedings of the IFIP WG 2.2/2.3 Working Conference on Programming Concepts and Methods*, pages 443–458, Sea of Gallilee, Israel, April 1990. North-Holland.
- 15 Alan Hoffman and Richard Karp. On nonterminating stochastic games. *Management Science*, 12(5):359–370, January 1966.
- 16 Bengt Jonsson and Kim Larsen. Specification and refinement of probabilistic processes. In *Proceedings of the 6th Annual Symposium on Logic in Computer Science*, pages 266–277, Amsterdam, The Netherlands, July 1991. IEEE.
- 17 Leonid Kantorovich. On the transfer of masses (in Russian). *Doklady Akademii Nauk*, 5(1):1–4, 1942. Translated in *Management Science*, 5(1):1–4, October 1958.
- 18 Leonid Khachiyan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, 1979.
- 19 Kim Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, September 1991.
- 20 Mary Melekopoglou and Anne Condon. On the complexity of the policy iteration algorithm. Computer Science Technical Report 941, University of Wisconsin, Madison, WI, USA, June 1990.
- 21 Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, Germany, 1980.
- 22 James Orlin. A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming*, 78(2):109–129, August 1997.
- 23 Prakash Panangaden. *Labelled Markov Processes*. Imperial College Press, London, United Kingdom, 2009.

- 24 David Park. Concurrency and automata on infinite sequences. In Peter Deussen, editor, *Proceedings of 5th GI-Conference on Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183, Karlsruhe, Germany, March 1981. Springer-Verlag.
- 25 Davide Sangiorgi. *Introduction to Bisimulation and Coinduction*. Cambridge University Press, Cambridge, United Kingdom, 2012.
- 26 Lloyd Shapley. Stochastic games. *Proceedings of the Academy of Sciences*, 39(10):1095–1100, October 1953.
- 27 James K. Strayer. *Linear programming and its applications*. Springer-Verlag, New York, NY, USA, 1989.
- 28 Alfred Tarski. *A decision method for elementary algebra and geometry*. University of California Press, Berkeley, CA, USA, 1951.
- 29 Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1/2):343–359, May 1996.