

ON-THE-FLY COMPUTATION OF BISIMILARITY DISTANCES

GIORGIO BACCI, GIOVANNI BACCI, KIM G. LARSEN, AND RADU MARDARE

Dept. of Computer Science, Aalborg University, Denmark
e-mail address: grbacci@cs.aau.dk

Dept. of Computer Science, Aalborg University, Denmark
e-mail address: giovbacci@cs.aau.dk

Dept. of Computer Science, Aalborg University, Denmark
e-mail address: kgl@cs.aau.dk

Dept. of Computer Science, Aalborg University, Denmark
e-mail address: mardare@cs.aau.dk

ABSTRACT. We propose a distance between continuous-time Markov chains (CTMCs) and study the problem of computing it by comparing three different algorithmic methodologies: iterative, linear program, and on-the-fly.

In a work presented at FoSSaCS'12, Chen et al. characterized the bisimilarity distance of Desharnais et al. between discrete-time Markov chains as an optimal solution of a linear program that can be solved by using the ellipsoid method. Inspired by their result, we propose a novel linear program characterization to compute the distance in the continuous-time settings. Differently from previous proposals, ours has a number of constraints that is bounded by a polynomial in the size of the CTMC. This, in particular, proves that the distance we propose can be computed in polynomial time.

Despite its theoretical importance, the proposed linear program characterization turns out to be inefficient in practice. Nevertheless, driven by the encouraging results of our previous work presented at TACAS'13, we propose an efficient on-the-fly algorithm which, unlike the other mentioned solutions, computes the distances between two given states avoiding an exhaustive exploration of the state space. This technique works by successively refining over-approximations of the target distances using a greedy strategy which ensures that the state space is further explored only when the current approximations are improved.

Tests performed on a consistent set of (pseudo)randomly generated CTMCs show that our algorithm improves, on average, the efficiency of the corresponding iterative and linear program methods with orders of magnitude.

1998 ACM Subject Classification: G.3.I.1.4,I.6.4.

Key words and phrases: Markov chains, Continuous-time Markov chains, behavioral distances, on-the-fly algorithm, probabilistic systems.

Extended and improved version of the paper by the same authors appeared in TACAS 2013.

Work supported by Sapere Aude: DFF-Young Researchers Grant 10-085054 of the Danish Council for Independent Research, by the VKR Center of Excellence MT-LAB and by the Sino-Danish Basic Research Center IDEA4CPS. .

INTRODUCTION

Continuous-time Markov chains (CTMCs) are one of the most prominent models in performance and dependability analysis. They are exploited in a broad range of applications, and constitute the underlying semantics of many modeling formalisms for real-time probabilistic systems such as Markovian queuing networks, stochastic process algebras, and calculi for systems biology. An example of CTMC is presented in Figure 1(left). Here, state s_1 goes to state s_3 and s_4 with probability $\frac{1}{3}$ and $\frac{2}{3}$, respectively. Each state has an associated *exit-rate* representing the rate of an exponentially distributed random variable that characterizes the residence-time in the state. For example, the probability to move from s_1 to any other state within time $t \geq 0$ is given by $\int_0^t 3e^{-3x} dx = 1 - e^{-3t}$. A state with no outgoing transitions (as s_3 in Figure 1) is called *absorbing*, and represents a terminating state of the system.

A key concept for reasoning about the equivalence of probabilistic systems is Larsen and Skou’s *probabilistic bisimulation* for discrete-time Markov chains (MCs). This notion have been extended to several type of probabilistic systems, including CTMCs. In Figure 1(left) s_4 and s_5 are bisimilar. Moreover, although s_1 and s_2 move with different probabilities to state s_4 and s_5 , their probabilities to reach any bisimilarity class is the same, so that, also s_1 and s_2 are bisimilar.

However, when the numerical values of probabilities are based on statistical sampling or subject to error estimates, any behavioral analysis based on a notion of equivalence is too fragile, as it only relates processes with identical behaviors. This issue is illustrated in Figure 1(right), where the states t_1 and t_2 (i.e., the counterpart of s_1 and s_2 , respectively, after a perturbation of the transition probabilities) are not bisimilar. A similar situation occurs considering perturbations on the exit-rates or on associated labels, if one assumes they are taken from a metric space.

This is a common issue in applications such as, systems biology [TK10], planning [CP11], games [CdAMR10], or security [CG09], where one is interested in knowing whether two processes that may differ by a small amount in the real-valued parameters (probabilities, rates, etc.) have “sufficiently” similar behaviours. This motivated the development of the metric theory for probabilistic systems, initiated by Desharnais et al. [DGJP04] and greatly developed and explored by De Alfaro, van Breugel, Worrell, and others [dAMRS07, vBW06, vBSW08]. It consists in proposing a *bisimilarity distance* (*pseudometric*), which

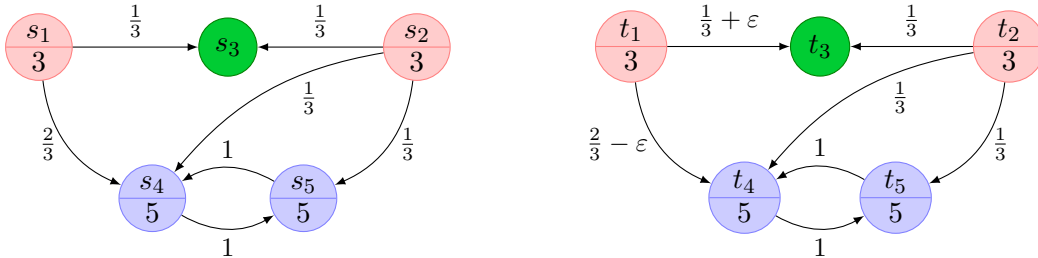


Figure 1: A CTMC (left) and an ε -perturbation of it (right), for some $\varepsilon \in (0, \frac{2}{3})$. Labels are represented by different colors; states are additionally labelled with their exit rates; and transitions with probability 0 are omitted.

measures the behavioral similarity of two models. These pseudometrics, e.g., the one proposed by Desharnais et al., are parametric in a *discount factor* that controls the significance of the future in the measurements.

Since van Breugel et al. have presented a fixed point characterization of the aforementioned pseudometric in [vBW01], several iterative algorithms have been developed in order to compute its approximation up to any degree of accuracy [FPP04, vBW06, vBSW08]. Recently, Chen et al. [CvBW12] proved that, for finite MCs with rational transition function, the bisimilarity pseudometrics can be computed exactly in polynomial time. The proof consists in describing the pseudometric as the solution of a linear program that can be solved using the *ellipsoid method*. Although the ellipsoid method is theoretically efficient, “*computational experiments with the method are very discouraging and it is in practice by no means a competitor of the, theoretically inefficient, simplex method*”, as stated in [Sch86]. Unfortunately, in this case the simplex method cannot be used to speed up performances in practice, since the linear program to be solved may have an exponential number of constraints in the number of states of the MC.

In this paper, we introduce a bisimilarity pseudometric over CTMCs that extends that of Desharnais et al. over MCs, and we consider the problem of computing it both from a theoretical and practical point of view. We show that the proposed distance can be computed in polynomial time in the size of the CTMC. This is obtained by reducing the problem of computing the distance to that of finding an optimal solution of a linear program that can be solved using the ellipsoid method. Notably, differently from the proposal in [CvBW12], our linear program characterization has a number of constraints that is bounded by a polynomial in the size of the CTMC. This, in particular, allows one to avoid the use of the ellipsoid algorithm in favor of the simplex or the interior point methods.

However, also in this case, the linear program characterization turns out to be inefficient in practice, even for small CTMCs. Nevertheless, supported by the encouraging results in our previous work [BBLM13], we propose to follow an on-the-fly approach for computing the distance. This is inspired by an alternative characterization of the bisimilarity pseudometric based on the notion of *coupling structure* for a CTMC. Each coupling structure is associated with a *discrepancy function* that represents an over-approximation of the distance. The problem of computing the pseudometric is then reduced to that of searching for an *optimal* coupling structure whose associated discrepancy coincides with the distance. The exploration of the coupling structures is based on a greedy strategy that, given a coupling structure, moves to a new one by ensuring an actual improvement of the current discrepancy function. This strategy will eventually find an optimal coupling structure. The method is sound independently from the initial starting coupling structure. Notably, the moving strategy is based on a *local* update of the current coupling structure. Since the update is local, when the goal is to compute the distance only between certain pairs of states, the construction of the coupling structures can be done *on-the-fly*, delimiting the exploration only on those states that are demanded during the computation.

The efficiency of our algorithm has been evaluated on a significant set of randomly generated CTMCs. The results show that our algorithm performs orders of magnitude better than the corresponding iterative and linear program implementations. Moreover, we provide empirical evidence that our algorithm enjoys good execution running times.

One of the main practical advantages of our approach consists in that one can focus on computing only the distances between states that are of particular interest. This is useful in practice, for instance when large systems are considered and visiting the entire state

space is computationally expensive. A similar issue has been considered by Comanici et al., in [CPP12] in the case of Markov decision processes with rewards, who noticed that for computing the approximated pseudometric one does not need to update the current value for all the pairs at each iteration, but it is sufficient only to focus on the pairs where changes are happening rapidly. In our approach, the termination condition is checked locally, still ensuring that the local optimum corresponds to the global one. Our methods can also be used in combination with approximation techniques as, for instance, to provide a least over-approximation of the behavioral distance given over-estimates of some particular distances.

Synopsis: The paper is organized as follows. In Section 1, we recall the basic preliminaries on continuous-time Markov chains and define the bisimilarity pseudometric. Section 2 is devoted to the analysis of the complexity of the problem of computing such a distance. Here, two approaches are considered: an approximated iterative method and a linear program characterization. In Section 3, we provide an alternative characterization of the distance based on the notion of coupling structure. This is the basis for the development of an on-the-fly algorithm (Section 5) for the computation of the pseudometric, whose correctness and termination is proven in Section 4. The efficiency of this algorithm is supported by experimental results, shown in Section 6. Final remarks and conclusions are in Section 7.

1. CONTINUOUS-TIME MARKOV CHAINS AND BISIMILARITY PSEUDOMETRICS

We recall the definitions of (finite) L -labelled continuous-time Markov chains (CTMCs) for a nonempty set of labels L , and *stochastic bisimilarity* over them. Then, we introduce a *behavioral pseudometric* over CTMCs to be considered as a quantitative generalization of stochastic bisimilarity.

Given a set X , a discrete probability distribution over it is a finitely supported function $\mu: X \rightarrow [0, 1]$ such that $\mu(X) = 1$, where $\mu(E) = \sum_{x \in E} \mu(x)$, for $E \subseteq X$. We denote the set of discrete probability distributions over X by $\mathcal{D}(X)$.

Definition 1.1 (Continuous-time Markov chain). An L -labelled *continuous-time Markov chain* is a tuple $\mathcal{M} = (S, A, \tau, \rho, \ell)$ consisting of a countable nonempty finite set S of *states*, a set $A \subseteq S$ of *absorbing states*, a *transition probability function* $\tau: S \setminus A \rightarrow \mathcal{D}(S)$, a *exit rate function* $\rho: S \setminus A \rightarrow \mathbb{R}_{>0}$, and a *labeling function* $\ell: S \rightarrow L$. ■

The labels in L represent properties of interest that hold in a particular state according to the labeling function $\ell: S \rightarrow L$. If $s \in S$ is the current state of the system and $E \subseteq S$ is a subset of states, $\tau(s)(E) \in [0, 1]$ corresponds to the probability that a transition from s to arbitrary $s' \in E$ is taken, and $\rho(s) \in \mathbb{R}_{>0}$ represents the rate of an exponentially distributed random variable that characterizes the residence time in the state s before any transition is taken. Therefore, the probability to make a transition from state s to any $s' \in E$ within time unit $t \in \mathbb{R}_{\geq 0}$ is given by $\tau(s)(E) \cdot \exp[-\rho(s)t]$, where $\exp[r](E) = \int_E r e^{-rx} dx$, for any Borel subset $E \subseteq \mathbb{R}_{\geq 0}$ and $r > 0$. Absorbing states in $A \subseteq S$ are used to represent termination or deadlock states. An example of CTMC is shown in Figure 1.

For discrete-time Markov chains, the standard notion of behavioral equivalence is probabilistic bisimulation of Larsen and Skou [LS91]. The following definition extends it to CTMCs. To ease the notation, for $A \subseteq S$, we introduce the relation $\equiv_A \subseteq S \times S$ defined by $s \equiv_A s'$ if either $s, s' \in A$ or $s, s' \notin A$.

Definition 1.2 (Stochastic Bisimulation). Let $\mathcal{M} = (S, A, \tau, \rho, \ell)$ be a CTMC. An equivalence relation $R \subseteq S \times S$ is a *stochastic bisimulation* on \mathcal{M} if whenever $s R t$, then

- (i) $s \equiv_A t$, $\ell(s) = \ell(t)$, and
- (ii) if $s, t \in A$, then $\rho(s) = \rho(t)$ and, for all $C \in S/R$, $\tau(s)(C) = \tau(t)(C)$.

Two states $s, t \in S$ are *bisimilar* with respect to \mathcal{M} , written $s \sim_{\mathcal{M}} t$, if they are related by some probabilistic bisimulation on \mathcal{M} .

Intuitively, two states are bisimilar if they have the same labels, they agree on being absorbing or not, and, in the case they are non-absorbing, their residence-time distributions and probability of moving by a single transition to any given class of bisimilar states is always the same. As an example of two stochastic bisimilar states, consider s_1 and s_2 in the CTMC depicted on the left hand side of in Figure 1. A bisimulation relation that relates them is the equivalence relation with equivalence classes given by $\{s_1, s_2\}$, $\{s_3\}$, and $\{s_4, s_5\}$.

1.1. Bisimilarity Pseudometrics on CTMCs. In this section, we introduce a family of pseudometrics on CTMCs parametric in a discount factor $\lambda \in (0, 1)$. Following the approach of [vBHMW07], given a CTMC $\mathcal{M} = (S, A, \tau, \rho, \ell)$ we define a (1-bounded) pseudometric on S as the least fixed point of an operator on the set $[0, 1]^{S \times S}$ of functions from $S \times S$ to $[0, 1]$. This pseudometric is then shown to be adequate with respect to stochastic bisimilarity: we prove that two states are stochastic bisimilar if and only if they have distance zero.

Recall that $d: X \times X \rightarrow \mathbb{R}_{\geq 0}$ is a *pseudometric* on a set X if $d(x, x) = 0$ (reflexivity), $d(x, y) = d(y, x)$ (symmetry) and $d(x, y) + d(y, z) \geq d(x, z)$ (triangular inequality), for arbitrary $x, y, z \in X$; it is a *metric* if, in addition, $d(x, y) = 0$ iff $x = y$. A pair (X, d) where d is a (pseudo)metric on X is called a (*pseudo*)*metric space*.

The operator we are going to introduce will use three key ingredients: a 1-bounded metric¹ $d_L: L \times L \rightarrow [0, 1]$ on the set of labels, a distance between residence-time distributions, and a distance between transition distributions. The first is meant to measure the *static* differences with respect to the labels associated with the states, the last two are meant to capture the differences in the *dynamics*, respectively, with respect to the continuous and discrete probabilistic choices.

To this end, we consider two distances over probability distributions. The first one is the, so called, *total variation metric*, defined for arbitrary distributions $\mu, \nu \in \mathcal{D}(\mathbb{R}_{\geq 0})$ as

$$\|\mu - \nu\|_{\text{TV}} = \sup_E |\mu(E) - \nu(E)|,$$

where the supremum is taken over the Borel measurable sets of $\mathbb{R}_{\geq 0}$. The second one is the *Kantorovich distance*, which is based on the notion of *coupling* of probability measures, which we introduce next in the case of probability distributions over finite sets.

Definition 1.3 (Coupling). Let S be a finite set, and let $\mu, \nu \in \mathcal{D}(S)$. A probability distribution $\omega \in \mathcal{D}(S \times S)$ is said a *coupling* for (μ, ν) if, for arbitrary $u, v \in S$

$$\sum_{v \in S} \omega(u, v) = \mu(u) \quad \text{and} \quad \sum_{u \in S} \omega(u, v) = \nu(v).$$

In other words, ω is a joint probability distribution with left and right marginal, respectively, given by μ and ν . We denote the set of couplings for (μ, ν) by $\Omega(\mu, \nu)$.

¹Since the set S of states is assumed to be finite, one may assume the set of labels to be so as well. Thus, the metric d_L on labels can be bounded without loss of generality.

For a finite set S and a 1-bounded distance $d: S \times S \rightarrow [0, 1]$ over it, the *Kantorovich distance* is defined, for arbitrary distributions $\mu, \nu \in \mathcal{D}(S)$ as follows

$$\mathcal{K}_d(\mu, \nu) = \min\{\sum_{u,v \in S} d(u, v) \cdot \omega(u, v) \mid \omega \in \Omega(\mu, \nu)\}.$$

Intuitively, \mathcal{K}_d lifts a (1-bounded) distance over S to a (1-bounded) distance over its probability distributions. One can show that \mathcal{K}_d is a (pseudo)metric if d is a (pseudo)metric.

Now, consider the following functional operator.

Definition 1.4. Let $\mathcal{M} = (S, A, \tau, \rho, \ell)$ be CTMC and $\lambda \in (0, 1)$ a *discount factor*. The function $\Delta_\lambda^{\mathcal{M}}: [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$ is defined as follows, for $d: S \times S \rightarrow [0, 1]$ and $s, t \in S$

$$\Delta_\lambda^{\mathcal{M}}(d)(s, t) = \begin{cases} 1 & \text{if } s \not\equiv_A t \\ \mathcal{L}(s, t) & \text{if } s, t \in A \\ \max\{\mathcal{L}(s, t), \lambda \cdot \mathcal{T}(d)(s, t)\} & \text{if } s, t \notin A \end{cases}$$

where $\mathcal{T}: [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$ and $\mathcal{L}, \mathcal{E}: S \times S \rightarrow [0, 1]$ are respectively defined by

$$\begin{aligned} \mathcal{T}(d)(s, t) &= \mathcal{E}(s, t) + (1 - \mathcal{E}(s, t)) \cdot \mathcal{K}_d(\tau(s), \tau(t)), \\ \mathcal{L}(s, t) &= d_L(\ell(s), \ell(t)), \quad \text{and} \quad \mathcal{E}(s, t) = \|\exp[\rho(s)] - \exp[\rho(t)]\|_{\text{TV}}. \end{aligned}$$

The functional $\Delta_\lambda^{\mathcal{M}}$ measures the difference of two states with respect to: their labels (by means of the pseudometric \mathcal{L}), their residence-time distributions (by means of the pseudometric \mathcal{E}), and their discrete probabilities to move to the next state (by means of the Kantorovich distance). If two states disagree on been absorbing (or not) they are considered incomparable, and their distance is set to 1. If both states are absorbing, they express no dynamic behavior, hence they are compared statically, and their distance corresponds to that occurring between their respective labels. Finally, if the states are non-absorbing, then they are compared with respect to both their static and dynamic features, namely, taking the maximum among their respective associated distances. Specifically, the value $\mathcal{E}(s, t)$ corresponds to the least probability that two transitions are taken independently from the states s and t at different moments in time. This value is used by the functional \mathcal{T} to measure the overall differences that might occur in the dynamics of the two states in combination with the Kantorovich distance between their transition probability distributions.

The set $[0, 1]^{S \times S}$ is endowed with the partial order \sqsubseteq defined as $d \sqsubseteq d'$ iff $d(s, t) \leq d'(s, t)$ for all $s, t \in S$ and it forms a complete lattice. The bottom element $\mathbf{0}$ is the constant 0 function, while the top element is the constant 1 function. For any subset $D \subseteq [0, 1]^{S \times S}$, the least upper bound $\bigsqcup D$, and greatest lower bound $\bigsqcap D$ are, respectively, given by $(\bigsqcup D)(s, t) = \sup_{d \in D} d(s, t)$ and $(\bigsqcap D)(s, t) = \inf_{d \in D} d(s, t)$, for all $s, t \in S$.

It is easy to check that, for any \mathcal{M} and $\lambda \in (0, 1)$, $\Delta_\lambda^{\mathcal{M}}$ is monotone (i.e., whenever $d \sqsubseteq d'$, then $\Delta_\lambda^{\mathcal{M}}(d) \sqsubseteq \Delta_\lambda^{\mathcal{M}}(d')$), thus, since $([0, 1]^{S \times S}, \sqsubseteq)$ is a complete lattice, by Tarski's fixed point theorem it admits least and greatest fixed points.

Definition 1.5 (Bisimilarity distance). Let \mathcal{M} be a CTMC and $\lambda \in (0, 1)$. The λ -discounted *bisimilarity pseudometric* on \mathcal{M} , denoted by $\delta_\lambda^{\mathcal{M}}$, is the least fixed point of $\Delta_\lambda^{\mathcal{M}}$.

The rest of the section is devoted to show that the least fixed point $\delta_\lambda^{\mathcal{M}}$ is indeed a pseudometric and, moreover, is adequate with respect to stochastic bisimilarity (Theorem 1.9). This justifies the definition above. To this end we need some technical lemmas. In particular, we prove that $\Delta_\lambda^{\mathcal{M}}$ preserves pseudometrics (Lemma 1.6) and it is non-expansive (Lemma 1.7).

Hereafter, unless mentioned otherwise, we fix a CTMC $\mathcal{M} = (S, A, \tau, \rho, \ell)$ and a discount factor $\lambda \in (0, 1)$. To ease the notation $\Delta_\lambda^{\mathcal{M}}$, $\delta_\lambda^{\mathcal{M}}$, and $\sim_{\mathcal{M}}$ will be denoted simply by Δ_λ , δ_λ , and \sim , respectively, whenever \mathcal{M} is clear from the context.

Lemma 1.6. *The operator Δ_λ preserves pseudometrics.*

Proof. Let $d: S \times S \rightarrow [0, 1]$ be a pseudometric. We want to prove that $\Delta_\lambda(d)$ is a pseudometric. Recall that $\mathcal{L}, \mathcal{E}: S \times S \rightarrow [0, 1]$ are pseudometrics. Thus, since the point wise maximum of pseudometrics is a pseudometric, it suffices to prove that \mathcal{T} preserves pseudometrics. Recall that, $\mathcal{K}_d: \mathcal{D}(S) \times \mathcal{D}(S) \rightarrow [0, 1]$ is a pseudometric, since d is so. Thus, reflexivity and symmetry are immediate. The only nontrivial case is triangular inequality. Let $s, t, u \in S$, we want to prove $\mathcal{T}(d)(s, t) \leq \mathcal{T}(d)(s, u) + \mathcal{T}(d)(u, t)$. First note that, for any $0 \leq \alpha, \beta \leq 1$ and $\alpha' \geq \alpha$, the following holds:

$$\begin{aligned} \alpha + (1 - \alpha)\beta &= \beta - \beta + \alpha + (1 - \alpha)\beta \\ &= \beta - \alpha\beta - (1 - \alpha)\beta + \alpha + (1 - \alpha)\beta && (0 \leq \alpha \leq 1) \\ &= \beta - \alpha\beta + \alpha = \beta + (1 - \beta)\alpha \\ &\leq \beta + (1 - \beta)\alpha' && (\alpha \leq \alpha' \text{ and } 0 \leq \beta \leq 1) \\ &= \alpha' + (1 - \alpha')\beta. \end{aligned}$$

Thus, since \mathcal{E} is a pseudometric, by triangular inequality and the above we have

$$\begin{aligned} \mathcal{T}(d)(s, t) &= \mathcal{E}(s, t) + (1 - \mathcal{E}(s, t)) \cdot \mathcal{K}_d(\tau(s), \tau(t)) && (\text{def. } \mathcal{T}) \\ &\leq \mathcal{E}(s, u) + \mathcal{E}(u, t) + (1 - (\mathcal{E}(s, u) + \mathcal{E}(u, t))) \cdot \mathcal{K}_d(\tau(s), \tau(t)) && (*) \end{aligned}$$

If we show that the last summand in (*) is \leq than the sum of $(1 - \mathcal{E}(s, u)) \cdot \mathcal{K}_d(\tau(s), \tau(u))$ and $(1 - \mathcal{E}(u, t)) \cdot \mathcal{K}_d(\tau(u), \tau(t))$, we get the following, and we are done

$$\begin{aligned} &\leq \mathcal{E}(s, u) + (1 - \mathcal{E}(s, u)) \cdot \mathcal{K}_d(\tau(s), \tau(u)) + \mathcal{E}(u, t) + (1 - \mathcal{E}(u, t)) \cdot \mathcal{K}_d(\tau(u), \tau(t)) \\ &= \mathcal{T}(d)(s, t) + \mathcal{T}(d)(s, t). && (\text{def. } \mathcal{T}) \end{aligned}$$

To this end, consider two cases. If $\mathcal{E}(s, u) + \mathcal{E}(u, t) > 1$ then the inequality holds trivially, since $1 - (\mathcal{E}(s, u) + \mathcal{E}(u, t)) < 0$, so that the last summand in (*) is negative. Instead, if $\mathcal{E}(s, u) + \mathcal{E}(u, t) \leq 1$ then $1 - (\mathcal{E}(s, u) + \mathcal{E}(u, t)) \geq 0$, so we have

$$\begin{aligned} &(1 - (\mathcal{E}(s, u) + \mathcal{E}(u, t))) \cdot \mathcal{K}_d(\tau(s), \tau(t)) \\ &\leq (1 - (\mathcal{E}(s, u) + \mathcal{E}(u, t))) \cdot (\mathcal{K}_d(\tau(s), \tau(u)) + \mathcal{K}_d(\tau(u), \tau(t))) && (\text{triang. } \mathcal{K}_d) \\ &= (1 - (\mathcal{E}(s, u) + \mathcal{E}(u, t))) \cdot \mathcal{K}_d(\tau(s), \tau(u)) + (1 - (\mathcal{E}(s, u) + \mathcal{E}(u, t))) \cdot \mathcal{K}_d(\tau(u), \tau(t)) \\ &\leq (1 - \mathcal{E}(s, u)) \cdot \mathcal{K}_d(\tau(s), \tau(u)) + (1 - \mathcal{E}(u, t)) \cdot \mathcal{K}_d(\tau(u), \tau(t)) \end{aligned}$$

and we are done. \square

The set $[0, 1]^{S \times S}$ can be turned into a metric space by means of the supremum norm $\|d - d'\| = \sup_{s, t \in S} |d(s, t) - d'(s, t)|$. Next we show that the λ -discounted functional operator Δ_λ is λ -non-expansive, that is $\|\Delta_\lambda(d') - \Delta_\lambda(d)\| \leq \lambda \cdot \|d' - d\|$, for any $d, d' \in [0, 1]^{S \times S}$.

Lemma 1.7. *The operator Δ_λ is λ -non-expansive*

Proof. By [vB12, Corollary 1], to prove that Δ_λ is λ -non-expansive it suffices to show that, whenever $d \sqsubseteq d'$ then, for all $s, t \in S$, $\Delta_\lambda(d')(s, t) - \Delta_\lambda(d)(s, t) \leq \lambda \cdot \|d' - d\|$. The only

nontrivial case is when $s, t \notin A$ and $\mathcal{L}(s, t) < \lambda \cdot \mathcal{T}(d)(s, t)$. To this end, assume that $\mathcal{K}_d(s, t)(\tau(s), \tau(t)) = \sum_{u, v \in S} d(u, v) \cdot \omega(u, v)$, for some $\omega \in \Omega(\tau(s), \tau(t))$, then we have

$$\begin{aligned}
\Delta_\lambda(d')(s, t) - \Delta_\lambda(d)(s, t) &\leq \lambda \cdot (\mathcal{T}(d')(s, s') - \mathcal{T}(d)(s, s')) \\
&\leq \lambda \cdot (\mathcal{K}_{d'}(s, t) - \mathcal{K}_d(s, t)) \\
&\leq \lambda \cdot (\sum_{u, v \in S} d'(u, v) \cdot \omega(u, v) - \sum_{u, v \in S} d(u, v) \cdot \omega(u, v)) \\
&= \lambda \cdot (\sum_{u, v \in S} (d'(u, v) - d(u, v)) \cdot \omega(u, v)) \\
&\leq \lambda \cdot (\sum_{u, v \in S} \|d' - d\| \cdot \omega(u, v)) \\
&= \lambda \cdot (\|d' - d\| \cdot \sum_{u, v \in S} \omega(u, v)) \\
&= \lambda \cdot \|d' - d\|.
\end{aligned}$$

□

It is standard that $[0, 1]^{S \times S}$ with the supremum norm forms a complete metric space (i.e, every Cauchy sequence converges). Therefore, since $\lambda \in (0, 1)$, a direct consequence of Lemma 1.7 and Banach's fixed point theorem is the following.

Theorem 1.8. *For any $\lambda \in (0, 1)$, δ_λ is the unique fixed point of Δ_λ . Moreover, for any $n \in \mathbb{N}$, and $d: S \times S \rightarrow [0, 1]$ we have $\|\delta_\lambda - \Delta_\lambda^n(d)\| \leq \frac{\lambda^n}{1-\lambda} \|\Delta_\lambda(d) - d\|$.*

Now we are ready to state the main theorem of this section.

Theorem 1.9 (Bisimilarity pseudometric). *The least fixed point δ_λ of Δ_λ is a pseudometric. Moreover, for any $s, t \in S$, $s \sim t$ iff $\delta_\lambda(s, t) = 0$.*

Proof. We first prove that δ_λ is a pseudometric. By Lemma 1.7 and Banach's fixed point theorem, $\delta_\lambda = \bigsqcup_{n \in \mathbb{N}} \Delta_\lambda^n(\mathbf{0})$. Clearly, $\mathbf{0}$ is a pseudometric. Thus by Lemma 1.6, a simple induction on n shows that, for all $n \in \mathbb{N}$, $\Delta_\lambda^n(\mathbf{0})$ is a pseudometric. Since the least upper bound with respect to \sqsubseteq preserves pseudometrics, we have that δ_λ is so.

Now we are left to prove that, for any $s, t \in S$, $s \sim t$ iff $\delta_\lambda(s, t) = 0$.

(\Rightarrow) We prove that $R = \{(s, s') \mid \delta_\lambda(s, t) = 0\}$ is a stochastic bisimulation. Clearly, R is an equivalence. Assume $(s, s') \in R$, then, by definition of Δ_λ , one of the following holds:

- (i) $s, t \in A$ and $\mathcal{L}(s, t) = 0$;
- (ii) $s, t \notin A$, $\mathcal{L}(s, t) = 0$, and $\mathcal{T}(\delta_\lambda)(s, t) = 0$.

If (i) holds, by $\mathcal{L}(s, t) = 0$ we get that $\ell(s) = \ell(t)$. If (ii) holds, we have $\mathcal{E}(s, t) = 0$ and $\mathcal{K}_{\delta_\lambda}(\tau(s), \tau(t)) = 0$. By $\mathcal{E}(s, t) = 0$ we get $\exp[\rho(s)] = \exp[\rho(t)]$ and hence $\rho(s) = \rho(t)$. By [FPP04, Lemma 3.1], $\mathcal{K}_{\delta_\lambda}(\tau(s), \tau(t)) = 0$ implies that, for all $C \in S/R$, $\tau(s)(C) = \tau(t)(C)$. Therefore R is a bisimulation.

(\Leftarrow) Let $R \subseteq S \times S$ be a stochastic bisimulation on \mathcal{M} , and define $d_R: S \times S \rightarrow [0, 1]$ by $d_R(s, t) = 0$ if $(s, t) \in R$ and $d_R(s, t) = 1$ if $(s, t) \notin R$. We show that $\Delta_\lambda(d_R) \sqsubseteq d_R$. If $(s, t) \notin R$, then $d_R(s, t) = 1 \geq \Delta_\lambda(d_R)(s, t)$. If $(s, t) \in R$, then $\ell(s) = \ell(t)$ and one of the following holds:

- (i) $s, t \in A$;
- (ii) $s, t \notin A$, $\rho(s) = \rho(t)$ and, $\forall C \in S/R. \tau(s)(C) = \tau(t)(C)$.

If (i) holds, $\Delta_\lambda(d_R)(s, t) = \mathcal{L}(s, t) = 0 = d_R(s, t)$. If (ii) holds, by [FPP04, Lemma 3.1] and the fact that, for all $C \in S/R$, $\tau(s)(C) = \tau(t)(C)$, we have $\mathcal{K}_{d_R}(\tau(s), \tau(t)) = 0$. Moreover $\mathcal{E}(s, t) = 0$. This gives that $\Delta_\lambda(d_R)(s, t) = 0 = d_R(s, t)$. By the generality of the

bisimulation relation R and by Tarski's fixed point theorem, we have that $s \sim s'$ implies $\delta_\lambda(s, s') = 0$. \square

2. COMPLEXITY AND LINEAR PROGRAMMING REPRESENTATION

In this section, we study the problem of computing the bisimilarity distance by considering two different approaches. The former is an iterative method that approximates δ_λ from below (resp. above) successively applying the operator Δ_λ starting from the least (resp. greatest) element in $[0, 1]^{S \times S}$. The latter is based on a linear program characterization of δ_λ that is based on the Kantorovich duality [Vil03]. In contrast to an analogous proposal in [CvBW12], our linear program has a number of constraints that is polynomially bounded in the size of the CTMC. As a consequence, the bisimilarity distance δ_λ can be computed in polynomial time in the size of the CTMC.

2.1. Iterative method. By Theorem 1.8, for any $\epsilon > 0$, it follows that to get ϵ -close to δ_λ , it suffices to iterate the application of the fixed point operator $\lceil \log_\lambda \epsilon \rceil$ times.

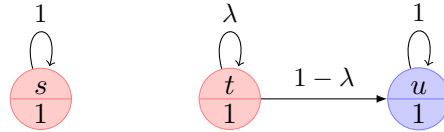
Proposition 2.1. *For any $\epsilon > 0$ and $d: S \times S \rightarrow [0, 1]$, $\|\delta_\lambda - \Delta_\lambda^{\lceil \log_\lambda \epsilon \rceil}(d)\| \leq \epsilon$.*

Proof. By Theorem 1.8 we have $\|\delta_\lambda - \Delta_\lambda^n(d)\| \leq \frac{\lambda^n}{1-\lambda} \|\Delta_\lambda(d) - d\|$ and by $\|\Delta_\lambda(d) - d\| \leq 1$, we have $\|\delta_\lambda - \Delta_\lambda^n(d)\| \leq \frac{\lambda^n}{1-\lambda}$. For $n = \log_\lambda(\epsilon - \epsilon\lambda)$, we have $\epsilon = \frac{\lambda^n}{1-\lambda}$. Therefore, by Lemma 1.7 and $\lceil \log_\lambda \epsilon \rceil \geq \log_\lambda(\epsilon - \epsilon\lambda)$, we have $\|\delta_\lambda - \Delta_\lambda^{\lceil \log_\lambda \epsilon \rceil}(d)\| \leq \epsilon$. \square

By the above result, we obtain a simple method for approximating δ_λ . If the starting point is $\mathbf{0}$ we obtain an under-approximation, whereas starting from $\mathbf{1}$ we get an over-approximation. Both the approximations can be taken arbitrary close to the exact value.

However, as shown in the following example, the exact distance value cannot be reached in general. This holds for any discount factor.

Example 2.2 ([CvBW12]). Consider the {red, blue}-labeled CTMC below.



Let $d_L: L \times L \rightarrow [0, 1]$ be the discrete metric over L , defined as $d_L(l, l') = 0$ if $l' = l$ and 1 otherwise. One can check that $\delta_\lambda(s, t) = \frac{\lambda - \lambda^2}{1 - \lambda^2}$ and, for all $n \in \mathbb{N}$, $\Delta_\lambda^n(\mathbf{0})(s, t) \leq \frac{\lambda - \lambda^{2n+1}}{1 + \lambda}$. Since, for all $n \in \mathbb{N}$, $\frac{\lambda - \lambda^{2n+1}}{1 + \lambda} < \frac{\lambda - \lambda^2}{1 - \lambda^2}$, we have that the fixed point cannot be reached in a finite number of iterations. \blacksquare

In [CvBW12] is shown that the bisimilarity distance of Desharnais et al. [DGJP04] can be computed *exactly* by iterating the fixed point operator up to a precision that allows one to use the continued fraction algorithm to yield the exact value of the fixed point. This method can be applied provided that the pseudometric has rational values. In their case, this is ensured assuming that the transition probabilities are rational. Unfortunately, in our

case this cannot be ensured under the same conditions. Indeed, the total variation distance between exponential distributions with rates $r, r' > 0$ is analytically solved as follows

$$\|\exp[r] - \exp[r']\|_{\text{TV}} = \begin{cases} 0 & \text{if } r = r' \\ \left| \left(\frac{r'}{r}\right)^{\frac{r}{r-r'}} - \left(\frac{r'}{r}\right)^{\frac{r'}{r-r'}} \right| & \text{otherwise} \end{cases} \quad (2.1)$$

thus, even restricting to rational exit-rates and probabilities, the distance may assume irrational values. As a consequence, we cannot assume to compute, in general, the exact distance values.

2.2. Linear Program Characterization. Our linear program characterization leverages on two key results. The first one is the unicity of the fixed point of Δ_λ (Theorem 1.8). The second one is a *dual* linear program characterization of the Kantorovich distance.

For S finite, $d: S \times S \rightarrow [0, 1]$, and $\mu, \nu \in \mathcal{D}(S)$, the value $\mathcal{K}_d(\mu, \nu)$ coincides with the optimal value of the following linear program

$$\begin{aligned} \mathcal{K}_d(\mu, \nu) = \min_{\omega} \quad & \sum_{s,t \in S} d(u, v) \cdot \omega_{u,v} \\ & \sum_v \omega_{u,v} = \mu(u) \quad \forall u \in S \\ & \sum_u \omega_{u,v} = \nu(v) \quad \forall v \in S. \end{aligned} \quad (2.2)$$

By a standard argument in linear optimization, the above can be alternatively represented by the following dual linear program

$$\begin{aligned} \mathcal{K}_d(\mu, \nu) = \max_y \quad & \sum_{u \in S} (\mu(u) - \nu(u)) \cdot y_u \\ & y_u - y_v \leq d(u, v) \quad \forall u, v \in S. \end{aligned} \quad (2.3)$$

This alternative characterization is a special case of a more general result commonly known as the *Kantorovich duality* and largely studied in linear optimization theory (see [Vil03]).

Let $n = |S|$ and $m = n - |A|$. Consider the linear program in Figure 2, hereafter denoted by $D_\lambda(\mathcal{M})$, with variables $d \in \mathbb{R}^{n^2}$, $y \in \mathbb{R}^{m^2+n}$ and $k, m \in \mathbb{R}^{m^2}$. The objective function of $D_\lambda(\mathcal{M})$ attains its optimal value when k and m are maximized in every component. Therefore, according to (2.3), an optimal solution $(d^*, y^*, k^*, m^*) \in D_\lambda(\mathcal{M})$ satisfies the following equalities

$$\begin{aligned} \forall s, t \notin A. \quad & m_{s,t}^* = \min\{\mathcal{L}(s, t), \lambda(\mathcal{E}(s, t) + (1 - \mathcal{E}(s, t))\mathcal{K}_{d^*}(\tau(s), \tau(t)))\} \\ \forall s, t \notin A. \quad & k_{s,t}^* = \mathcal{K}_{d^*}(\tau(s), \tau(t)) \end{aligned}$$

By the above equalities and by the feasibility of the solution, it follows that d^* is a fixed point of Δ_λ . Since the distance is the unique fixed point of Δ_λ , $d^* = \delta_\lambda$.

Theorem 2.3. *Let \mathcal{M} be a CTMC, $\lambda \in (0, 1)$ be a discount factor, and (d^*, y^*, k^*, m^*) be an optimal solution of $D_\lambda(\mathcal{M})$. Then, for all $s, t \in S$, $d_{s,t}^* = \delta_\lambda(s, t)$.*

$$\begin{array}{ll}
\arg \max_{d,y,k,m} \sum_{s,t \notin A} k_{s,t} + m_{s,t} & \\
d_{s,t} = 1 & \forall s,t \in S. s \not\equiv_A t \\
d_{s,t} = \mathcal{L}(s,t) & \forall s,t \in A \\
d_{s,t} = \mathcal{L}(s,t) + \lambda(\mathcal{E}(s,t) + (1 - \mathcal{E}(s,t))k_{s,t}) - m_{s,t} & \forall s,t \notin A \\
m_{s,t} \leq \mathcal{L}(s,t) & \forall s,t \notin A \\
m_{s,t} \leq \lambda(\mathcal{E}(s,t) + (1 - \mathcal{E}(s,t))k_{s,t}) & \forall s,t \notin A \\
k_{s,t} = \sum_{u \in S} (\tau(s)(u) - \tau(t)(u)) \cdot y_u^{s,t} & \forall s,t \notin A \\
y_u^{s,t} - y_v^{s,t} \leq d_{u,v} & \forall s,t \notin A, \forall u,v \in S
\end{array}$$

Figure 2: Linear program characterization of δ_λ for $\lambda \in (0, 1)$.

Proof. Let (d^*, y^*, k^*, m^*) be an optimal solution of $D_\lambda(\mathcal{M})$, and consider the following linear program, hereafter denoted by D'_λ .

$$\begin{array}{ll}
\arg \max_{y,k,m} \sum_{s,t \notin A} k_{s,t} + m_{s,t} & \\
m_{s,t} \leq \mathcal{L}(s,t) & \forall s,t \notin A \\
m_{s,t} \leq \lambda(\mathcal{E}(s,t) + (1 - \mathcal{E}(s,t))k_{s,t}) & \forall s,t \notin A \\
k_{s,t} = \sum_{u \in S} (\tau(s)(u) - \tau(t)(u)) \cdot y_u^{s,t} & \forall s,t \notin A \\
y_u^{s,t} - y_v^{s,t} \leq d_{u,v}^* & \forall s,t \notin A, \forall u,v \in S
\end{array}$$

Since d^* is part of a feasible solution of $D_\lambda(\mathcal{M})$ and the constraints of D'_λ are a subset of those of $D_\lambda(\mathcal{M})$, the optimal value for D'_λ is greater or equal than the one for $D_\lambda(\mathcal{M})$.

For each $s, t \notin A$, when $k_{s,t}$ is maximal also $m_{s,t}$ reaches its maximal value. Therefore, an optimal solution for D'_λ will be achieved when, for each $s, t \notin A$, the following holds

$$\begin{array}{l}
k_{s,t} = \max_y \sum_{u \in S} (\tau(s)(u) - \tau(t)(u)) \cdot y_u \\
y_u - y_v \leq d_{u,v}^* \quad \forall u, v \in S
\end{array} \tag{2.4}$$

By the Kantorowich duality (2.3), the optimal value of (2.4) is $\mathcal{K}_{d^*}(\tau(s), \tau(t))$. Since $m_{s,t}$ is a lower bound for $\{\mathcal{L}(s,t), \lambda(\mathcal{E}(s,t) + (1 - \mathcal{E}(s,t))k_{s,t})\}$, we obtain an optimal solution for D'_λ by instantiating the variables as follows

$$\forall s, t \notin A. \quad m_{s,t} = \min\{\mathcal{L}(s,t), \lambda(\mathcal{E}(s,t) + (1 - \mathcal{E}(s,t))\mathcal{K}_{d^*}(\tau(s), \tau(t)))\} \tag{2.5}$$

$$\forall s, t \notin A. \quad k_{s,t} = \mathcal{K}_{d^*}(\tau(s), \tau(t)) \tag{2.6}$$

$$\forall s, t \notin A. \quad y^{s,t} \in \arg \max_y \sum_{u \in S} (\tau(s)(u) - \tau(t)(u)) \cdot y_u \\
y_u - y_v \leq d_{u,v}^* \quad \forall u, v \in S \tag{2.7}$$

We will show that the above instantiation is part of a feasible solution of $D_\lambda(\mathcal{M})$, provided that d^* is a fixed point for Δ_λ . For any $s, t \notin A$ the following holds

$$\begin{aligned}
d_{s,t}^* &= \mathcal{L}(s, t) + \lambda(\mathcal{E}(s, t) + (1 - \mathcal{E}(s, t)) k_{s,t}) - m_{s,t} && \text{(by def. } D_\lambda(\mathcal{M})) \\
&= \mathcal{L}(s, t) + \lambda(\mathcal{E}(s, t) + (1 - \mathcal{E}(s, t)) \mathcal{K}_{d^*}(\tau(s), \tau(t))) - m_{s,t} && \text{(by (2.6))} \\
&= \mathcal{L}(s, t) + \mathcal{T}(d^*)(s, t) - m_{s,t} && \text{(by def. } \mathcal{T}) \\
&= \mathcal{L}(s, t) + \mathcal{T}(d^*)(s, t) - \min\{\mathcal{L}(s, t), \mathcal{T}(d^*)(s, t)\} && \text{(by (2.5) and def. } \mathcal{T}) \\
&= \max\{\mathcal{L}(s, t), \mathcal{T}(d^*)(s, t)\},
\end{aligned}$$

thus d^* is a fixed point of Δ_λ . The thesis follows by Theorem 1.8. \square

In [CvBW12] it has been shown that the bisimilarity distance of Desharnais et al. can be computed in polynomial time as the solution of a linear program that can be solved by using the ellipsoid method. However, in their proposal the number of constraints may be exponential in the size of the model. This is due to the fact that the Kantorovich distance is resolved listing all the couplings that correspond to vertices of the transportation polytopes involved in the definition of the distance [Dem61].

In contrast, our proposal has a number of constraints and unknowns² bounded by $4|S|^2$ and $|S|^3 + 3|S|^2$, respectively. This allows one to use general algorithms for solving LP problems (such as the *simplex* and the *interior point* methods) that, in practice, are more efficient than the ellipsoid method.

Moreover, this allows us to state the following complexity result.

Theorem 2.4. δ_λ can be computed in polynomial-time in the size of \mathcal{M} .

Proof. By Theorem 2.3, δ_λ can be computed within the time it takes to construct and solve $D_\lambda(\mathcal{M})$. $D_\lambda(\mathcal{M})$ has a number of constraints and unknowns that is bounded by a polynomial in the size of \mathcal{M} , therefore its construction can be performed in polynomial time. For the same reason, $D_\lambda(\mathcal{M})$ admits a polynomial time separation algorithm: whenever a solution is given, its feasibility is checked by scanning each inequality in $D_\lambda(\mathcal{M})$; otherwise the first encountered inequality that is not satisfied is returned as a separation hyperplane. Therefore, the thesis follows by solving $D_\lambda(\mathcal{M})$ using the ellipsoid method together with the naïve separation algorithm described above. \square

3. ALTERNATIVE CHARACTERIZATION OF THE PSEUDOMETRIC

In the following, we propose an alternative characterization of the bisimilarity distance δ_λ , based on the notion of *coupling structure*. Our result generalizes the one proposed in [CvBW12, BBLM13] for MCs, to the continuous-time settings.

Definition 3.1 (Coupling Structure). Let $\mathcal{M} = (S, A, \pi, \ell)$ be a CTMC. A *coupling structure* for \mathcal{M} is a function $\mathcal{C}: (S \setminus A) \times (S \setminus A) \rightarrow \mathcal{D}(S \times S)$ such that, for all $s, t \notin A$, $\mathcal{C}(s, t) \in \Omega(\tau(s), \tau(t))$.

Intuitively, a coupling structure for \mathcal{M} can be thought of as a joint transition probability distribution with left and right marginals point wisely equal to $\tau: S \setminus A \rightarrow \mathcal{D}(S)$.

The following definition adapts the definition of the operator Δ_λ (see Definition 1.4) with respect to the notion of coupling structure for a CTMC.

²Actually, the variables k are used only for ease the presentation but they can be removed by substitution.

Definition 3.2. Let $\mathcal{M} = (S, A, \tau, \rho, \ell)$ be CTMC, \mathcal{C} a coupling structure for \mathcal{M} , and $\lambda \in (0, 1)$ a *discount factor*. The function $\Gamma_\lambda^\mathcal{C}: [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$ is defined as follows, for $d: S \times S \rightarrow [0, 1]$ and $s, t \in S$

$$\Gamma_\lambda^\mathcal{C}(d)(s, t) = \begin{cases} 1 & \text{if } s \not\equiv_A t \\ \mathcal{L}(s, t) & \text{if } s, t \in A \\ \max\{\mathcal{L}(s, t), \lambda \cdot \Theta(d)(s, t)\} & \text{if } s, t \notin A \end{cases}$$

where $\mathcal{L}, \mathcal{E}: S \times S \rightarrow [0, 1]$ are as in Definition 1.4 and $\Theta: [0, 1]^{S \times S} \rightarrow [0, 1]^{S \times S}$ is given by

$$\Theta(d)(s, t) = \mathcal{E}(s, t) + (1 - \mathcal{E}(s, t)) \cdot \sum_{u, v \in S} d(u, v) \cdot \mathcal{C}(s, t)(u, v).$$

Recall that the Kantorovich distance between two distributions μ and ν is defined as $\mathcal{K}_d(\mu, \nu) = \min_\omega \sum_{u, v \in S} d(u, v) \cdot \omega(u, v)$, where the minimum is taken over all the possible couplings $\omega \in \Omega(\mu, \nu)$. Thus, the operator $\Gamma_\lambda^\mathcal{C}$ can intuitively be thought of as a possible instance of $\Delta_\lambda^\mathcal{M}$ with respect to a fixed choice of the couplings given by \mathcal{C} .

One can easily check that $\Gamma_\lambda^\mathcal{C}$ is monotone, thus, by Tarski's fixed point theorem, it admits least and greatest fixed points. The least fixed point, in particular, will be denoted by $\gamma_\lambda^\mathcal{C}$ and referred to as the λ -*discrepancy* of \mathcal{C} .

Theorem 3.3 (Minimum coupling). $\delta_\lambda = \min\{\gamma_\lambda^\mathcal{C} \mid \mathcal{C} \text{ coupling structure for } \mathcal{M}\}$.

Proof. We first prove that $\delta_\lambda \sqsubseteq \gamma_\lambda^\mathcal{C}$, for any coupling structure \mathcal{C} for \mathcal{M} . By Tarski's fixed point theorem, it suffices to prove that, for any $d: S \times S \rightarrow [0, 1]$, $\Delta_\lambda(d) \sqsubseteq \Gamma_\lambda^\mathcal{C}(d)$. The only nontrivial case is when $s, t \notin A$, which follows by definition of \mathcal{K}_d , by noticing that $\mathcal{T}(d)(s, t) \leq \Theta(d)(s, t)$ and that the maximum is order preserving. It remains to prove that the minimum is attained. To this end, define a coupling structure \mathcal{C}^* as $\mathcal{C}^*(s, t) = \omega_{s, t}$, for $s, t \notin A$, where $\omega_{s, t} \in \Omega(\tau(s), \tau(t))$ is such that $\mathcal{K}_{\delta_\lambda}(\tau(s), \tau(t)) = \sum_{u, v} \delta_\lambda(u, v) \cdot \omega_{s, t}(u, v)$. By construction, $\delta_\lambda = \Gamma^{\mathcal{C}^*}(\delta_\lambda)$, hence $\gamma_\lambda^{\mathcal{C}^*} \sqsubseteq \delta_\lambda$. Since \mathcal{C}^* is a coupling structure for \mathcal{M} , by what we have shown above we also have $\delta_\lambda \sqsubseteq \gamma_\lambda^{\mathcal{C}^*}$. Therefore, $\delta_\lambda = \gamma_\lambda^{\mathcal{C}^*}$. \square

4. GREEDY COMPUTATION OF THE BISIMILARITY DISTANCE

Inspired by the characterization given in Theorem 3.3, we propose a procedure to compute the bisimilarity pseudometric that is alternative to those previously described.

The set of coupling structures for \mathcal{M} can be endowed with the preorder \preceq_λ defined as $\mathcal{C} \preceq_\lambda \mathcal{C}'$ iff $\gamma_\lambda^\mathcal{C} \sqsubseteq \gamma_\lambda^{\mathcal{C}'}$. Theorem 3.3 suggests to look at all the coupling structures \mathcal{C} for \mathcal{M} in order to find an optimal one, i.e., minimal w.r.t. \preceq_λ . However, it is clear that the enumeration of all the couplings is unfeasible, therefore it is crucial to provide an efficient search strategy which prevents us to do that. Moreover we also need an efficient method for computing the λ -discrepancy associated with a coupling structure.

4.1. Computing the λ -Discrepancy. In this section we consider the problem of computing the λ -discrepancy associated with a coupling structure.

By Tarski's fixed point theorem, $\gamma_\lambda^\mathcal{C}$ corresponds to the least pre-fixed point of $\Gamma^\mathcal{C}$, that is $\gamma_\lambda^\mathcal{C} = \bigsqcap \{d \in [0, 1]^{S \times S} \mid \Gamma^\mathcal{C}(d) \sqsubseteq d\}$. This allows us to compute the λ -discrepancy associated

with \mathcal{C} as the optimal solution of the following linear program, denoted by $Discr_\lambda(\mathcal{C})$.

$$\begin{aligned} \arg \min_d \quad & \sum_{s,t \in S} d_{s,t} \\ & d_{s,t} \geq 1 && \text{if } s \not\equiv_A t \\ & d_{s,t} \geq \mathcal{L}(s,t) && \text{if } s \equiv_A t \\ & d_{s,t} \geq \lambda(\mathcal{E}(s,t) + (1 - \mathcal{E}(s,t)) \cdot \sum_{u,v \in S} d_{u,v} \cdot \mathcal{C}(s,t)(u,v)) && \text{if } s, t \notin A \end{aligned}$$

$Discr_\lambda(\mathcal{C})$ has a number of inequalities that is bounded by $2|S|^2$ and $|S|^2$ unknowns, thus, it can be efficiently solved using the interior point method.

Remark 4.1. If one is interested in computing the λ -discrepancy for a particular pair of states (s,t) , the method above can be applied on the least independent set of inequalities containing the variable $d_{s,t}$. Moreover, assuming that for some pairs the values associated to d are known, the set of constraints can be further decreased by substitution. \blacksquare

4.2. Greedy Strategy for Optimal Coupling Structures. In this section, we propose a *greedy strategy* that moves toward an optimal coupling structure starting from any given one. Then, we provide sufficient and necessary conditions for a coupling structure, to ensure that its associated λ -discrepancy coincides with δ_λ .

Hereafter we fix a CTMC $\mathcal{M} = (S, A, \tau, \rho, \ell)$ and a coupling structure \mathcal{C} for it. The greedy strategy takes a coupling structure and locally updates it at a given pair of states in such a way that it decreases it with respect to \preceq_λ . For $s, t \notin A$ and $\omega \in \Omega(\tau(s), \tau(t))$, we denote by $\mathcal{C}[(s,t)/\omega]$ the *update* of \mathcal{C} at (s,t) with ω , defined as $\mathcal{C}[(s,t)/\omega](u,v) = \mathcal{C}(u,v)$, for all $(u,v) \neq (s,t)$, and $\mathcal{C}[(s,t)/\omega](s,t) = \omega$; it is worth noting that, by construction, $\mathcal{C}[(s,t)/\omega]$ is a coupling structure of \mathcal{M} .

The next lemma gives a sufficient condition for an update to be effective for the strategy.

Lemma 4.2. *Let $s, t \notin A$ and $\omega \in \Omega(\tau(s), \tau(t))$. Then, for $\mathcal{D} = \mathcal{C}[(s,t)/\omega]$ and any $\lambda \in (0, 1]$, if $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}})(s,t) < \gamma_\lambda^{\mathcal{C}}(s,t)$ then $\gamma_\lambda^{\mathcal{D}} \sqsubset \gamma_\lambda^{\mathcal{C}}$.*

Proof. It suffices to show that $\Gamma^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}}) \sqsubset \gamma_\lambda^{\mathcal{C}}$, i.e., that $\gamma_\lambda^{\mathcal{C}}$ is a strict post-fixed point of $\Gamma_\lambda^{\mathcal{D}}$. Then, the thesis follows by Tarski's fixed point theorem.

Let $u, v \in S$. If $u \not\equiv_A v$, then $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}})(u,v) = 1 = \Gamma_\lambda^{\mathcal{C}}(\gamma_\lambda^{\mathcal{C}})(u,v) = \gamma_\lambda^{\mathcal{C}}(u,v)$. If $u, v \in A$, then $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}})(u,v) = \mathcal{L}(u,v) = \Gamma_\lambda^{\mathcal{C}}(\gamma_\lambda^{\mathcal{C}})(u,v) = \gamma_\lambda^{\mathcal{C}}(u,v)$. If $u, v \notin A$ and $(u,v) \neq (s,t)$, by definition of \mathcal{D} , we have that $\bar{\omega}(u,v) = \omega(u,v)$, hence $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}})(u,v) = \Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}})(u,v)$. The remaining case, i.e., $(u,v) = (s,t)$, holds by hypothesis. This proves $\Gamma^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}}) \sqsubset \gamma_\lambda^{\mathcal{C}}$. \square

Lemma 4.2 states that \mathcal{C} can be improved w.r.t. \preceq_λ by updating it at (s,t) , if $s, t \notin A$ and there exists a coupling $\omega \in \Omega(\tau(s), \tau(t))$ such that the following holds

$$\sum_{u,v \in S} \gamma_\lambda^{\mathcal{C}}(u,v) \cdot \omega(u,v) < \sum_{u,v \in S} \gamma_\lambda^{\mathcal{C}}(u,v) \cdot \mathcal{C}(s,t)(u,v).$$

A coupling that enjoys the above condition is $\omega \in TP(\gamma_\lambda^{\mathcal{C}}, \tau(s), \tau(t))$ where, for arbitrary $\mu, \nu \in \mathcal{D}(S)$ and $c: S \times S \rightarrow [0, 1]$

$$\begin{aligned} TP(c, \mu, \nu) = \arg \min_{\omega} \quad & \sum_{s,t \in S} c(u,v) \cdot \omega_{u,v} \\ & \sum_v \omega_{u,v} = \mu(u) && \forall u \in S \\ & \sum_u \omega_{u,v} = \nu(v) && \forall v \in S. \end{aligned} \tag{4.1}$$

The above problem is usually referred to as the (homogeneous) *transportation problem* with μ and ν as left and right marginals, respectively, and transportation costs c . This problem has been extensively studied and comes with (several) efficient polynomial algorithmic solutions [Dan51, FF56].

This gives us an efficient solution to update any coupling structure, that, together with Lemma 4.2 represents a strategy for moving toward δ_λ by successive improvements on the coupling structures.

Now we proceed giving a sufficient and necessary condition for termination.

Lemma 4.3. *If $\gamma_\lambda^{\mathcal{C}} \neq \delta_\lambda$, then there exist $s, t \notin A$ and a coupling structure $\mathcal{D} = \mathcal{C}[(s, t)/\omega]$ for \mathcal{M} such that $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}})(s, t) < \gamma_\lambda^{\mathcal{C}}(s, t)$.*

Proof. We proceed by contraposition. If for all $s, t \notin A$ and $\omega \in \Omega(\tau(s), \tau(t))$, $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}})(s, t) \geq \gamma_\lambda^{\mathcal{C}}(s, t)$, then $\gamma_\lambda^{\mathcal{C}} = \Delta_\lambda(\gamma_\lambda^{\mathcal{C}})$. Since, by Theorem 1.8, Δ_λ has a unique fixed point, $\gamma_\lambda^{\mathcal{C}} = \delta_\lambda$. \square

The above result ensures that, unless \mathcal{C} is optimal w.r.t \preceq_λ , the hypothesis of Lemma 4.2 are satisfied, so that, we can further improve \mathcal{C} as aforesaid.

The next statement proves that this search strategy is correct.

Theorem 4.4. *$\delta_\lambda = \gamma_\lambda^{\mathcal{C}}$ iff there is no coupling \mathcal{D} for \mathcal{M} such that $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}}) \sqsubset \gamma_\lambda^{\mathcal{C}}$.*

Proof. We prove: $\delta_\lambda \neq \gamma_\lambda^{\mathcal{C}}$ iff there exists \mathcal{D} such that $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}}) \sqsubset \gamma_\lambda^{\mathcal{C}}$. (\Rightarrow) Assume $\delta_\lambda \neq \gamma_\lambda^{\mathcal{C}}$. By Lemma 4.3, there exist a pair of states $s, t \in S$ and a coupling $\omega \in \Omega(\tau(s), \tau(t))$ such that $\lambda \cdot \sum_{u, v \in S} \gamma_\lambda^{\mathcal{C}}(u, v) \cdot \omega(u, v) < \gamma_\lambda^{\mathcal{C}}(s, t)$. As in the proof of Lemma 4.2, we have that $\mathcal{D} = \mathcal{C}[(s, t)/\omega]$ satisfies $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}}) \sqsubset \gamma_\lambda^{\mathcal{C}}$. (\Leftarrow) Let \mathcal{D} be such that $\Gamma_\lambda^{\mathcal{D}}(\gamma_\lambda^{\mathcal{C}}) \sqsubset \gamma_\lambda^{\mathcal{C}}$. By Tarski's fixed point theorem $\gamma_\lambda^{\mathcal{D}} \sqsubset \gamma_\lambda^{\mathcal{C}}$. By Theorem 3.3, $\delta_\lambda \sqsubseteq \gamma_\lambda^{\mathcal{D}} \sqsubset \gamma_\lambda^{\mathcal{C}}$. \square

Remark 4.5. Note that, in general there could be an infinite number of couplings structures for a given CTMC. However, for each fixed $d \in [0, 1]^{S \times S}$, the linear function mapping ω to $\sum_{u, v \in S} d(u, v) \cdot \omega(u, v)$ achieves its minimum at some vertex in the transportation polytope $\Omega(\tau(s), \tau(t))$. Since the number of such vertices are finite, using the optimal TP schedule for the update, ensures that the search strategy is always terminating. \blacksquare

5. THE ON-THE-FLY ALGORITHM

In this section we describe an on-the-fly technique for computing the bisimilarity distance δ_λ fully exploiting the greedy strategy of Section 4.2.

Let $Q \subseteq S \times S$ and consider the problem of computing $\delta_\lambda(s, t)$ for all $(s, t) \in Q$. Recall that the strategy proposed in Section 4.2 consists in a traversal $\mathcal{C}_0 \triangleright_\lambda \mathcal{C}_1 \triangleright_\lambda \dots \triangleright_\lambda \mathcal{C}_n$ of the set of coupling structures for \mathcal{M} that starts from an arbitrary coupling structure \mathcal{C}_0 and leads to an optimal one \mathcal{C}_n . We observe that, for any $i < n$

- (1) the improvement of each coupling structure \mathcal{C}_i is obtained by a *local update* at some pair of states $u, v \notin A$, namely $\mathcal{C}_{i+1} = \mathcal{C}_i[(u, v)/\omega]$ for some $\omega \in TP(\gamma_\lambda^{\mathcal{C}_i}, \tau(u), \tau(v))$;
- (2) the pair (u, v) is chosen according to an optimality check that is performed *locally* among the couplings in $\Omega(\tau(u), \tau(v))$, i.e., $\mathcal{C}_i(u, v) \notin TP(\gamma_\lambda^{\mathcal{C}_i}, \tau(u), \tau(v))$;
- (3) whenever a coupling structure \mathcal{C}_i is considered, its associated λ -discrepancy $\gamma_\lambda^{\mathcal{C}_i}$ can be computed by solving the linear program $Discr_\lambda(\mathcal{C}_i)$ described in Section 4.1.

Among the observations above, only the last one requires to look at the coupling structure \mathcal{C}_i . However, as noticed in Remark 4.1, the value $\gamma_\lambda^{\mathcal{C}_i}(s, t)$ can be computed without considering the entire set of constraints of $Discr_\lambda(\mathcal{C}_i)$, but only the least independent set of inequalities that contains the variable $d_{s,t}$. Moreover, provided that for some pairs of states $E \subseteq S \times S$ the value of the distance is known, the linear program $Discr_\lambda(\mathcal{C}_i)$ can be further reduced by substituting the occurrences of the unknown $d_{u,v}$ by the constant $\delta_\lambda(u, v)$, for each $(u, v) \in E$. This suggests that we do not need to store the entire coupling structures, but they can be constructed *on-the-fly* during the calculation. Specifically, the couplings that are demanded to compute $\gamma_\lambda^{\mathcal{C}_i}(s, t)$ are only those $\mathcal{C}_i(u, v)$ such that $(s, t) \rightsquigarrow_{\mathcal{C}_i, E}^* (u, v)$, where $\rightsquigarrow_{\mathcal{C}_i, E}^*$ is the reflexive and transitive closure of $\rightsquigarrow_{\mathcal{C}_i, E} \subseteq S^2 \times S^2$, defined by

$$(s', t') \rightsquigarrow_{\mathcal{C}_i, E} (u', v') \quad \text{iff} \quad \mathcal{C}_i(s', t')(u', v') > 0 \text{ and } (u', v') \notin E.$$

The computation of the bisimilarity pseudometric is implemented by Algorithm 1. It takes as input a finite CTMC $\mathcal{M} = (S, A, \tau, \rho, \ell)$, a discount factor $\lambda \in (0, 1)$, and a query set $Q \subseteq S \times S$. We assume the following global variables to store:

- \mathcal{C} : the current (partial) coupling structure;
- d : the λ -discrepancy associated with \mathcal{C} ;
- $ToCompute$: the pairs of states for which the distance has to be computed;
- $Exact$: the set pairs of states (s, t) such that $d(s, t) = \delta_\lambda(s, t)$, hence those pairs which do not need to be further improved³;
- $Visited$: the set of pairs of states that as been visited so far.

Moreover, $\mathcal{R}_{s,t}(Exact, \mathcal{C})$ will denote the set $\{(u, v) \mid (s, t) \rightsquigarrow_{\mathcal{C}, Exact}^* (u, v)\}$.

At the beginning (line 1–2) both the coupling structure \mathcal{C} and the discrepancy d are empty, there are no visited states, no exact computed distances, and the pairs to be computed are those in the input query.

While there are still pairs left to be computed (line 3), we pick one (line 4), say (s, t) . According to the definition of δ_λ , if $s \not\equiv_A t$ then $\delta_\lambda(s, t) = 1$; if $s = t$ then $\delta_\lambda(s, t) = 0$ and if $s, t \in A$ then $\delta_\lambda(s, t) = \mathcal{L}(s, t)$, so that, $d(s, t)$ is set accordingly, and (s, t) is added to $Exact$ (lines 5–10). Otherwise, if (s, t) was not previously visited, a coupling $\omega \in \Omega(\tau(s), \tau(t))$ is guessed, and the routine *SetPair* updates the coupling structure \mathcal{C} at (s, t) with ω (line 14), then the routine *Discrepancy* updates d with the λ -discrepancy associated with \mathcal{C} (line 16). According to the greedy strategy, \mathcal{C} is successively improved and d is consequently updated, until no further improvements are possible (lines 17–21). Each improvement is obtained by replacing a sub-optimal coupling $C(u, v)$, for some $(u, v) \in \mathcal{R}_{s,t}(Exact, \mathcal{C})$, by one taken from $TP(d, \tau(u), \tau(v))$ (line 17). Note that, each improvement actually affects the current value of $d(s, t)$, since the update is performed on a pair in $\mathcal{R}_{s,t}(Exact, \mathcal{C})$. It is worth to note that \mathcal{C} and $Exact$ are constantly updated, hence $\mathcal{R}_{s,t}(Exact, \mathcal{C})$ may differ from one iteration to another.

When line 22 is reached, for each $(u, v) \in \mathcal{R}_{s,t}(Exact, \mathcal{C})$, we are guaranteed that $d(u, v) = \delta_\lambda(s, t)$, therefore $\mathcal{R}_{s,t}(Exact, \mathcal{C})$ is added to $Exact$ and, for these pairs, d will no longer be updated. At this point (line 23), the couplings associated with the pairs in $Exact$ can be removed from \mathcal{C} . In line 25, the exact pairs computed so far are removed from $ToCompute$. Finally, if no more pairs need be considered, the exact distance on Q is returned (line 27).

³Actually, the set $Exact$ contains those pairs such that $|d(s, t) - \delta_\lambda(s, t)| \leq \epsilon$ where ϵ corresponds to the precision of the machine. In our implementation $\epsilon = 10^{-9}$.

Algorithm 1 On-the-Fly Bisimilarity Pseudometric

Input: CTMC $\mathcal{M} = (S, A, \tau, \rho, \ell)$; discount factor $\lambda \in (0, 1)$; query $Q \subseteq S \times S$.

1. $\mathcal{C} \leftarrow \text{empty}$; $\text{Dom}_{\mathcal{C}} \leftarrow \emptyset$; $d \leftarrow \text{empty}$; —initialize data structures—
2. $\text{Visited} \leftarrow \emptyset$; $\text{Exact} \leftarrow \emptyset$; $\text{ToCompute} \leftarrow Q$
3. **while** $\text{ToCompute} \neq \emptyset$ **do**
4. pick $(s, t) \in \text{ToCompute}$
5. **if** $s \not\equiv_A t$ **then**
6. $d(s, t) \leftarrow 1$; $\text{Exact} \leftarrow \text{Exact} \cup \{(s, t)\}$; $\text{Visited} \leftarrow \text{Visited} \cup \{(s, t)\}$
7. **else if** $s = t$ **then**
8. $d(s, t) \leftarrow 0$; $\text{Exact} \leftarrow \text{Exact} \cup \{(s, t)\}$; $\text{Visited} \leftarrow \text{Visited} \cup \{(s, t)\}$
9. **else if** $s, t \in A$ **then**
10. $d(s, t) \leftarrow \mathcal{L}(s, t)$; $\text{Exact} \leftarrow \text{Exact} \cup \{(s, t)\}$; $\text{Visited} \leftarrow \text{Visited} \cup \{(s, t)\}$
11. **else** —if (s, t) is nontrivial—
12. **if** $(s, t) \notin \text{Visited}$ **then** —if (s, t) has not been encountered so far—
13. pick $\omega \in \Omega(\tau(s), \tau(t))$ —guess a coupling—
14. $\text{SetPair}(\mathcal{M}, (s, t), \omega)$ —update the current coupling structure—
15. **end if**
16. $\text{Discrepancy}(\lambda, (s, t))$ —update d as the λ -discrepancy for \mathcal{C} —
17. **while** $\exists (u, v) \in \mathcal{R}_{s,t}(\text{Exact}, \mathcal{C})$ such that $\mathcal{C}(u, v) \notin \text{TP}(d, \tau(u), \tau(v))$ **do**
18. $\omega \in \text{TP}(d, \tau(u), \tau(v))$ —pick an optimal coupling for s, t w.r.t. d —
19. $\text{SetPair}(\mathcal{M}, (u, v), \omega)$ —improve the current coupling structure—
20. $\text{Discrepancy}(\lambda, (s, t))$ —update d as the λ -discrepancy for \mathcal{C} —
21. **end while**
22. $\text{Exact} \leftarrow \text{Exact} \cup \mathcal{R}_{\mathcal{C}}(s, t)$ —add new exact distances—
23. remove from \mathcal{C} all the couplings associated with a pair in Exact
24. **end if**
25. $\text{ToCompute} \leftarrow \text{ToCompute} \setminus \text{Exact}$ —remove exactly computed pairs—
26. **end while**
27. **return** $d|_Q$ —return the distance for all pairs in Q —

Algorithm 1 calls the subroutines *SetPair* and *Discrepancy*. The former is used to construct and update the coupling structure \mathcal{C} , the latter to update the current over-approximation d during the computation. Next, we explain how they work.

SetPair (Algorithm 2) takes as input a CTMC $\mathcal{M} = (S, A, \tau, \rho, \ell)$, a pair of states $s, t \in S$, and a coupling $\omega \in \Omega(\tau(s), \tau(t))$. In lines 1–2, the coupling structure \mathcal{C} is set to ω at (s, t) , then (s, t) is added to *Visited*. The on-the-fly construction of the coupling structure is recursively propagated to the demanded successor pairs of (s, t) according to the information accumulated so far. During this construction, if some states with trivial distances are encountered, d and *Exact* are updated accordingly (lines 5–7).

Discrepancy (Algorithm 3) takes as input a discount factor $\lambda \in (0, 1)$ and a pair of states $s, t \notin A$. It constructs the least independent linear program obtained from $\text{Discr}_{\lambda}(\mathcal{C})$, that can compute $\gamma_{\lambda}^{\mathcal{C}}(s, t)$ using the information accumulated so far (line 1). In lines 3–7 the current λ -discrepancy is updated accordingly; and those pairs $(u, v) \in \mathcal{R}_{s,t}(\text{Exact}, \mathcal{C})$ for which the current λ -discrepancy coincide with the distance are added to *Exact*.

Algorithm 2 $SetPair(\mathcal{M}, (s, t), \omega)$

Input: CTMC $\mathcal{M} = (S, A, \tau, \rho, \ell)$; $s, t \in S$; $\omega \in \Omega(\tau(s), \tau(t))$

1. $\mathcal{C}(s, t) \leftarrow \omega$ —update the coupling at (s, t) with ω —
2. $Visited \leftarrow Visited \cup \{(s, t)\}$ —set (s, t) as visited—
3. **for all** $(u, v) \notin Visited$ such that $(s, t) \rightsquigarrow_{\mathcal{C}, Exact} (u, v)$ **do** —for all demanded pairs—
4. $Visited \leftarrow Visited \cup \{(u, v)\}$
5. **if** $u = v$ **then** $d(u, v) \leftarrow 0$; $Exact \leftarrow Exact \cup \{(u, v)\}$;
6. **if** $u \not\equiv_A v$ **then** $d(u, v) \leftarrow 1$; $Exact \leftarrow Exact \cup \{(u, v)\}$;
7. **if** $u, v \in A$ **then** $d(u, v) \leftarrow \mathcal{L}(u, v)$; $Exact \leftarrow Exact \cup \{(u, v)\}$;
8. // propagate the construction
9. **if** $(u, v) \notin Exact$ **then**
10. pick $\omega' \in \Omega(\tau(u), \tau(v))$ —guess a matching—
11. $SetPair(\mathcal{M}, (u, v), \omega')$
12. **end if**
13. **end for**

Algorithm 3 $Discrepancy(\lambda, (s, t))$

Input: discount factor $\lambda \in (0, 1)$; $s, t \in S \setminus A$

1. Let LP be the linear program obtained from $Discr_\lambda(\mathcal{C})$ by keeping only the inequalities associated with pairs in $\mathcal{R}_{s,t}(Exact, \mathcal{C})$ and replacing the unknown $d_{u,v}$ by the constant $d(u, v)$, for all $(u, v) \in Exact$.
2. $d^* \leftarrow$ optimal solution of LP
3. **for all** $(u, v) \in \mathcal{R}_{s,t}(Exact, \mathcal{C})$ **do** —update distances—
4. $d(u, v) \leftarrow d_{u,v}^*$
5. **if** $d(u, v) = 0$ or $d(u, v) = \mathcal{L}(u, v)$ **then**
6. $Exact \leftarrow Exact \cup \{(u, v)\}$
7. **end if**
8. **end for**

Next, we present a simple example of Algorithm 1, showing the main features of our method: (1) the on-the-fly construction of the (partial) coupling, and (2) the restriction only to those variables which are demanded for the solution of the system of linear equations.

Example 5.1 (On-the-fly computation). Consider the CTMC in Figure 3, and assume we want to compute the λ -discounted bisimilarity distance between states s_1 and s_4 , for $\lambda = \frac{1}{2}$.

Algorithm 1 starts by guessing an initial coupling structure \mathcal{C}_0 . This is done by considering only the pairs of states which are really needed in the computation. Starting from the pair (s_1, s_4) a coupling in $\omega_{1,4} \in \Omega(\tau(s_1), \tau(s_4))$ is guessed as in Figure 3 and assigned to $\mathcal{C}_0(s_1, s_4)$. This demands for the exploration of the pairs (s_2, s_3) , (s_2, s_4) , (s_1, s_2) and the guess of three new couplings $\omega_{2,3} \in \Omega(\tau(s_2), \tau(s_3))$, $\omega_{2,4} \in \Omega(\tau(s_2), \tau(s_4))$, and $\omega_{1,2} \in \Omega(\tau(s_1), \tau(s_2))$, to be associated in \mathcal{C}_0 with their corresponding pairs. Since no other pairs are demanded, the construction of \mathcal{C}_0 terminates as shown in Figure 3. The λ -discrepancy associated with \mathcal{C}_0 for the pair (s_1, s_4) is obtained as the solution of the

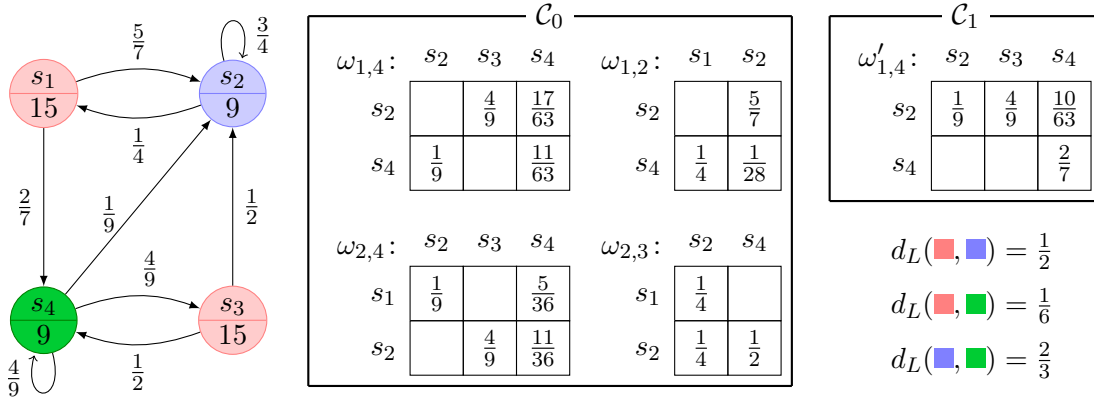


Figure 3: Execution trace for the computation of $\delta_{\frac{1}{2}}(1, 4)$ (details in Example 5.1).

following reduced linear program

$$\begin{aligned} & \arg \min_d (d_{1,4} + d_{2,3} + d_{2,4} + d_{1,2}) \\ & d_{1,4} \geq \frac{1}{6} \\ & d_{1,4} \geq \frac{\alpha}{2} + \frac{(1-\alpha)}{2} \cdot \left(\frac{4}{9} \cdot d_{2,3} + \frac{8}{21} \cdot d_{2,4} + \frac{11}{63} \cdot \overbrace{d_{4,4}}^{=0} \right) \\ & d_{2,3} \geq \frac{1}{2} \\ & d_{2,3} \geq \frac{\alpha}{2} + \frac{(1-\alpha)}{2} \cdot \left(\frac{1}{4} \cdot d_{1,2} + \frac{1}{4} \cdot \overbrace{d_{2,2}}^{=0} + \frac{1}{2} \cdot d_{2,4} \right) \\ & d_{2,4} \geq \frac{2}{3} \\ & d_{2,4} \geq \frac{1}{2} \cdot \left(\frac{1}{9} \cdot d_{1,2} + \frac{5}{36} \cdot d_{1,4} + \frac{4}{9} \cdot d_{2,3} + \frac{11}{36} \cdot d_{2,4} \right) \\ & d_{1,2} \geq \frac{1}{2} \\ & d_{1,2} \geq \frac{\alpha}{2} + \frac{(1-\alpha)}{2} \cdot \left(\frac{5}{7} \cdot \overbrace{d_{2,2}}^{=0} + \frac{1}{4} \cdot d_{1,4} + \frac{1}{28} \cdot d_{2,4} \right) \end{aligned}$$

where $\alpha = \|\exp[15] - \exp[9]\|_{\text{TV}} = \frac{\sqrt[6]{3/5}}{25}$ (by Equation (2.1)). Note that, the bisimilarity distance for the pairs (s_2, s_2) and (s_4, s_4) is always 0, thus $d_{2,2}$ and $d_{4,4}$ are substituted accordingly. The solution of the above linear program is $d^{\mathcal{C}_0}(s_1, s_4) = \frac{\alpha}{2} + \frac{5(1-\alpha)}{21}$, $d^{\mathcal{C}_0}(s_2, s_3) = \frac{1}{2}$, $d^{\mathcal{C}_0}(s_2, s_4) = \frac{2}{3}$, and $d^{\mathcal{C}_0}(s_1, s_2) = \frac{1}{2}$.

Since, the λ -discrepancy for (s_2, s_3) , (s_2, s_4) , and (s_1, s_2) equals the distance \mathcal{L} between their labels, it coincides with the bisimilarity distance, hence it cannot be further decreased. Consequently, the pairs of states are added to the set *Exact* and their associated couplings are removed from \mathcal{C}_0 . Note that, these pairs will no longer be considered in the construction of a coupling structure.

In order to decrease the λ -discrepancy of (s_1, s_4) , Algorithm 1 constructs a new coupling structure \mathcal{C}_1 . According to our greedy strategy, \mathcal{C}_1 is obtained from \mathcal{C}_0 updating $\mathcal{C}_0(s_1, s_4)$ (i.e., the only coupling left) by the coupling $\omega'_{1,4} \in \Omega(\tau(s_1), \tau(s_4))$ (shown in Figure 3) that

is obtained as the solution of a transportation problem with marginals $\tau(s_1)$ and $\tau(s_4)$, where the current λ -discrepancy is taken as cost function. The resulting coupling does not demand for the exploration of new pairs in the CTMC, hence the construction of \mathcal{C}_1 terminates. The reduced linear program associated with \mathcal{C}_1 is given by

$$\begin{aligned} & \arg \min_d d_{1,4} \\ & d_{1,4} \geq \frac{1}{6} \\ & d_{1,4} \geq \frac{\alpha}{2} + \frac{(1-\alpha)}{2} \cdot \left(\frac{1}{9} \cdot \overbrace{d_{2,2}}^{=0} + \frac{4}{9} \cdot \overbrace{d_{2,3}}^{=\frac{1}{2}} + \frac{10}{63} \cdot \overbrace{d_{2,4}}^{=\frac{2}{3}} + \frac{2}{7} \cdot \overbrace{d_{4,4}}^{=0} \right) \end{aligned}$$

whose solution is $d^{\mathcal{C}_1}(s_1, s_4) = \frac{\alpha}{2} + \frac{31(1-\alpha)}{189}$.

Solving again a new transportation problem with the improved current λ -discrepancy as cost function, we discover that the coupling structure \mathcal{C}_1 cannot be further improved, hence we stop the computation, returning $\delta_\lambda(s_1, s_4) = d^{\mathcal{C}_1}(s_1, s_4) = \frac{\alpha}{2} + \frac{31(1-\alpha)}{189}$. ■

Remark 5.2. Algorithm 1 can also be used for computing over-approximated distances. Indeed, assuming over-estimates for some particular distances are already known, they can be taken as inputs and used in our algorithm simply storing them in the variable d and treated as “exact” values. In this way our method will return the least over-approximation of the distance agreeing with the given over-estimates. This modification of the algorithm can be used to further decrease the exploration of the CTMC. Moreover, it can be employed in combination with approximated algorithms, having the advantage of an on-the-fly state space exploration. ■

6. EXPERIMENTAL RESULTS

In this section, we evaluate the performances of the on-the-fly algorithm on a collection of randomly generated CTMCs⁴.

First, we compare the execution times of the on-the-fly algorithm with those of the iterative method proposed in Section 2.1. Since the iterative method only allows for the computation of the distance for all state pairs at once, the comparison is (in fairness) made with respect to runs of our on-the-fly algorithm with input query being the set of all state pairs. For each input instance, the comparison involves the following steps:

- (a) we run the on-the-fly algorithm, storing both execution time and the number of solved transportation problems,
- (b) then, on the same instance, we execute the iterative method until the running time exceeds that of step 1. We report the number of iterations and the number of solved transportation problems.
- (c) Finally, we calculate the approximation error between the exact solution δ_λ computed by our method at step 1 and the approximate result d obtained in step 2 by the iterative method, as $\|\delta_\lambda - d\|$.

⁴ The tests have been performed on a prototype implementation coded in Wolfram Mathematica[®] 9 (available at <http://people.cs.aau.dk/~giovbacci/tools.html>) running on an Intel Core-i7 3.4 GHz processor with 12GB of RAM.

# States	On-the-Fly (exact)		Iterating (approximated)		Approx. Error
	Time (s)	# TPs	# Iterations	# TPs	
10	0.352	10.500	2.660	266.667	0.0339
12	0.772	19.700	2.850	410.403	0.0388
14	2.496	35.800	3.880	760.480	0.0318
16	4.549	50.607	5.142	1316.570	0.0230
18	13.709	78.611	6.638	2151.021	0.0206
20	22.044	109.146	7.243	2897.560	0.0149
22	50.258	140.727	7.409	3586.010	0.0145
24	67.049	175.481	7.826	4508.310	0.0141
26	112.924	219.255	9.509	6428.150	0.0025
28	247.583	295.533	11.133	8728.530	0.0004
30	284.252	307.698	10.679	9611.320	0.0006
40	296.633	330.824	11.294	18070.600	0.0004
50	807.522	368.500	16.900	42250.000	0.00001

Table 1: Comparison between the on-the-fly algorithm and the iterative method.

# States	out-deg = 3		$3 \leq \text{out-deg} \leq \# \text{ States}/2$	
	Time (s)	# TPs	Time (s)	# TPs
30	0.304	0.383	18.113	21.379
40	2.045	0.954	34.582	22.877
50	7.832	16.304	50.258	139.427

# States	out-deg = 3	
	Time (s)	# TPs
60	34.858	12.053
70	48.016	14.166
80	73.419	29.383
90	75.591	13.116
100	158.027	20.301

Table 2: Average performances of the on-the-fly algorithm on single-pair queries. Execution times and number of performed TPs are reported for CTMCs with different out-degree. For instances with more than 50 states the out-degree is fixed to 3;

This has been made on a collection of CTMCs varying from 10 to 50 states. For each $n = 10, \dots, 30$, we have considered 40 randomly generated CTMCs per out-degree, varying from 3 to n ; whereas for $n = 40$ and 50, the out-degree varies from 3 to 10. Table 1 reports the average results of the comparison obtained for a discount factor $\lambda = \frac{1}{2}$.

As it can be seen, our use of a greedy strategy in the construction of the couplings leads to a significant improvement in the performances. We are able to compute the exact solution before the iterative method can under-approximate it with an absolute error of ≈ 0.03 , which is a non-negligible error for a value within the interval $[0, 1]$.

So far, we only examined the case when the on-the-fly algorithm is run on all state pairs at once. Now, we show how the performance of our method is improved even further when the distance is computed only for single pairs of states. Table 2 shows the average execution times and number of solved transportation problems for (nontrivial) single-pair queries for randomly generated CTMCs with number of states varying from 30 to 100. In the first two columns we consider CTMCs with out-degree equal to 3, while the last two columns show the average values for out-degrees varying from 3 to half of the number of states of the CTMCs. The results show that, when the out-degree of the CTMCs is low, our algorithm performs orders of magnitude better than in the general case.

Notably, our on-the-fly method scales well when the out-degree is small and successive computation of the current λ -discrepancy are performed on a relatively small set of pairs.

As for the linear program characterization of the bisimilarity distance illustrated in Section 2.2, tests performed on small CTMCs show that solving $D_\lambda(\mathcal{M})$ is inefficient in practice, both using the simplex and the interior-point methods⁵. Even for CTMCs with less than 20 states, the computation times are in the order of hours. For this reason, the efficiency of our on-the-fly technique is by no mean comparable to the linear program solution.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a bisimilarity pseudometric for measuring the behavioral similarity between CTMCs, that extends that on MCs introduced by Desharnais et al. in [DGJP04]. Moreover, we gave a novel linear program characterization of the distance that, differently from similar previous proposals, have a number of constraints which is polynomial in the size of the CTMC. This proved that the bisimilarity pseudometric can be computed in polynomial time. Finally, we defined an on-the-fly algorithm for computing the bisimilarity distance. We demonstrated that, using on-the-fly techniques the computation time is improved with orders of magnitude with respect to the corresponding iterative and linear program approaches. Moreover, our technique allows for the computation on a set of target distances that might be done by only investigating a significantly reduced set of states, and for further improvement of speed.

Our algorithm can be practically used to address a large spectrum of problems. For instance, it can be seen as a method to decide whether two states of a given CTMC are probabilistic bisimilar, to identify bisimilarity classes, or to solve lumpability problems. It is sufficiently robust to be used with approximation techniques as, for instance, to provide a least over-approximation of the behavioural distance given over-estimates of some particular distances. It can be integrated with other approximate algorithms, having the advantage of the efficient on-the-fly state space exploration.

Having a practically efficient tool to compute bisimilarity distances opens the perspective of new applications already announced in previous research papers. One of these is the state space reduction problem for CTMCs. Our technique can be used in this context as an indicator for the sets of neighbour states that can be collapsed due to their similarity; it also provides a tool to estimate the difference between the initial CTMC and the reduced one, hence a tool for the approximation theory of CTMCs.

ACKNOWLEDGEMENT

We would like to thank Franck van Breugel for helpful discussions and for providing us the proofs that were omitted in the proceeding publication of [CvBW12].

⁵The implementation is done in Wolfram Mathematica[®] 9 and uses the Linear Program solvers available in the standard library.

REFERENCES

- [BBLM13] Giorgio Bacci, Giovanni Bacci, Kim G. Larsen, and Radu Mardare. On-the-Fly Exact Computation of Bisimilarity Distances. In *TACAS*, volume 7795 of *Lecture Notes in Computer Science*, pages 1–15, 2013.
- [CdAMR10] Krishnendu Chatterjee, Luca de Alfaro, Rupak Majumdar, and Vishwanath Raman. Algorithms for Game Metrics. *Logical Methods in Computer Science*, 6(3), 2010.
- [CG09] Xiaojuan Cai and Yonggen Gu. Measuring Anonymity. In *ISPEC '09*, pages 183–194, Berlin, Heidelberg, 2009. Springer-Verlag.
- [CP11] Gheorghe Comanici and Doina Precup. Basis function discovery using spectral clustering and bisimulation metrics. In *AAMAS '11*, volume 3, pages 1079–1080, Richland, SC, 2011. International Foundation for Autonomous Agents and Multiagent Systems.
- [CPP12] Gheorghe Comanici, Prakash Panangaden, and Doina Precup. On-the-Fly Algorithms for Bisimulation Metrics. *International Conference on Quantitative Evaluation of Systems*, 0:94–103, 2012.
- [CvBW12] Di Chen, Franck van Breugel, and James Worrell. On the Complexity of Computing Probabilistic Bisimilarity. In Lars Birkedal, editor, *FoSSaCS*, volume 7213 of *Lecture Notes in Computer Science*, pages 437–451. Springer, 2012.
- [dAMRS07] Luca de Alfaro, Rupak Majumdar, Vishwanath Raman, and Mariëlle Stoelinga. Game Relations and Metrics. In *LICS*, pages 99–108, July 2007.
- [Dan51] George B. Dantzig. Application of the Simplex method to a transportation problem. In T.C. Koopmans, editor, *Activity analysis of production and allocation*, pages 359–373. J. Wiley, New York, 1951.
- [Dem61] O. Demuth. Poznámka k dopravnímu [Czech, with German summary; A remark on the transportation problem]. *Časopis pro Pěstování Matematiky*, (86):103–110, 1961.
- [DGJP04] Josee Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004.
- [FF56] Lester Randolph Ford and Delbert Ray Fulkerson. Solving the Transportation Problem. *Management Science*, 3(1):24–32, 1956.
- [FPP04] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite Markov Decision Processes. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, UAI, pages 162–169. AUAI Press, 2004.
- [LS91] Kim Guldstrand Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.
- [Sch86] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, Inc., New York, NY, USA, 1986.
- [TK10] David Thorsley and Eric Klavins. Approximating stochastic biochemical processes with Wasserstein pseudometrics. *IET Systems Biology*, 4(3):193–211, 2010.
- [vB12] Franck van Breugel. On behavioural pseudometrics and closure ordinals. *Information Processing Letters*, 112(19):715–718, 2012.
- [vBHMW07] Franck van Breugel, Claudio Hermida, Michael Makkai, and James Worrell. Recursively defined metric spaces without contraction. *Theoretical Computer Science*, 380(1-2):143–163, 2007.
- [vBSW08] Franck van Breugel, Babita Sharma, and James Worrell. Approximating a Behavioural Pseudometric without Discount for Probabilistic Systems. *Logical Methods in Computer Science*, 4(2):1–23, 2008.
- [vBW01] Franck van Breugel and James Worrell. Towards quantitative verification of probabilistic transition systems. In *ICALP*, volume 2076 of *LNCS*, pages 421–432, 2001.
- [vBW06] Franck van Breugel and James Worrell. Approximating and computing behavioural distances in probabilistic transition systems. *Theoretical Computer Science*, 360(1-3):373–385, 2006.
- [Vil03] Cédric Villani. *Topics in Optimal Transportation*. Number 58 in Graduate Study in Mathematics. American Mathematical Society, Providence, RI, 2003.