Algorithms for Game Metrics (Full Version) *

Krishnendu Chatterjee[1] Luca de Alfaro[1] Rupak Majumdar[2] Vishwanath Raman[1]

[1]School of Engineering
UC Santa Cruz, CA, USA
{c_krish,luca,vishwa}@soe.ucsc.edu

[2]Department of Computer Science
UC Los Angeles, CA, USA
rupak@cs.ucla.edu

Abstract

Simulation and bisimulation metrics for stochastic systems provide a quantitative generalization of the classical simulation and bisimulation relations. These metrics capture the similarity of states with respect to quantitative specifications written in the quantitative μ-calculus and related probabilistic logics.

We present algorithms for computing the metrics on Markov decision processes (MDPs), turn-based stochastic games, and concurrent games. For turn-based games and MDPs, we provide a polynomial-time algorithm for the computation of the one-step metric distance between states. The algorithm is based on linear programming; it improves on the previous known exponential-time algorithm based on a reduction to the theory of reals. We then present PSPACE algorithms for both the decision problem and the problem of approximating the metric distance between two states, matching the best known algorithms for Markov chains. For the bisimulation kernel of the metric, which corresponds to probabilistic bisimulation, our algorithm works in time $\mathcal{O}(n^4)$ for both turn-based games and MDPs; improving the previously best known $\mathcal{O}(n^9 \cdot \log(n))$ time algorithm for MDPs.

For a concurrent game G, we show that computing the exact distance between states is at least as hard as computing the value of concurrent reachability games and the square-root-sum problem in computational geometry. We show that checking whether the metric distance is bounded by a rational r, can be accomplished via a reduction to the theory of real closed fields, involving a formula with three quantifier alternations, yielding $\mathcal{O}(|G|^{\mathcal{O}(|G|^5)})$ time complexity, improving the previously known reduction, which yielded $\mathcal{O}(|G|^{\mathcal{O}(|G|^7)})$ time complexity. These algorithms can be iterated to approximate the metrics using binary search.

1 Introduction

System metrics constitute a quantitative generalization of system relations. The bisimulation relation captures state *equivalence:* two states s and t are bisimilar if and only if they cannot be distinguished by any formula of the μ-calculus [5]. The bisimulation *metric* captures the *degree of difference* between two states: the bisimulation distance between s and t is a real number that provides a tight bound for the difference in value of formulas of the *quantitative* μ-calculus at s and t [10]. A similar connection holds between the simulation relation and the simulation metric.

The classical system relations are a basic tool in the study of *boolean* properties of systems, that is, the properties that yield a truth value. As an example, if a state s of a transition system can reach a set of target states R, written $s \models \diamond R$ in temporal logic, and t can simulate s, then we can conclude $t \models \diamond R$. System metrics play a similarly fundamental role in the study of the quantitative

*This work has been partially supported by NSF CCR-0132780, NSF CNS-0720884.

behavior of systems. As an example, if a state s of a Markov chain can reach a set of target states R with probability 0.8, written $s \models \mathbb{P}_{\geq 0.8} \Diamond R$, and if the metric simulation distance from t to s is 0.3, then we can conclude $t \models \mathbb{P}_{\geq 0.5} \Diamond R$. The simulation relation is at the basis of the notions of system refinement and implementation, where qualitative properties are concerned. In analogous fashion, simulation metrics provide a notion of approximate refinement and implementation for quantitative properties.

We consider three classes of systems:

- *Markov decision processes.* In these systems there is one player. At each state, the player can choose a move; the current state and the move determine a probability distribution over the successor states.

- *Turn-based games.* In these systems there are two players. At each state, only one of the two players can choose a move; the current state and the move determine a probability distribution over the successor states.

- *Concurrent games.* In these systems there are two players. At each state, both players choose moves simultaneously and independently; the current state and the chosen moves determine a probability distribution over the successor states.

System metrics were first studied for Markov chains and Markov decision processes (MDPs) [10, 25, 26, 11, 12], and they have recently been extended to two-player turn-based and concurrent games [9]. The fundamental property of the metrics is that they provide a tight bound for the difference in value that formulas belonging to quantitative specification languages assume at the states of a system. More precisely, let $q\mu$ indicate the *quantitative μ-calculus,* a specification language in which many of the classical specification properties, including reachability and safety properties, can be written [8]. The metric bisimulation distance between two states s and t, denoted $[s \simeq_g t]$, has the property that $[s \simeq_g t] = \sup_{\phi \in q\mu} |\phi(s) - \phi(t)|$, where $\phi(s)$ and $\phi(t)$ are the values ϕ assumes at s and t. To each metric is associated a *kernel:* the kernel of a metric d is the relation that relates the pairs of states that have distance 0; to each metric corresponds a metric kernel relation. The kernel of the simulation metric is *probabilistic simulation;* the kernel of the bisimulation metric is *probabilistic bisimulation* [21].

Metric as bound for discounted and long-run average payoff. Our first result is that the metrics developed in [9] provide a bound for the difference in *long-run average* and *discounted average* properties across states of a system. These average rewards play a central role in the theory of stochastic games, and in its applications to optimal control and economics [4, 15]. Thus, the metrics of [9] are useful both for system verification, and for performance evaluation, supporting our belief that they constitute the canonical metrics for the study of the similarity of states in a game.

Algorithmic results. Next, we investigate algorithms for the computation of the metrics. The metrics can be computed in iterative fashion, following the inductive way in which they are defined. A metric d can be computed as the limit of a monotonically increasing sequence of approximations $d_0, d_1, d_2, \ldots$, where $d_0(s,t)$ is the difference in value that variables can have at states s and t. For $k \geq 0$, d_{k+1} is obtained from d_k via $d_{k+1} = H(d_k)$, where the operator H depends on the metric (bisimulation, or simulation), and on the type of system. Our main results are as follows:

1. *Metrics for turn-based games and MDPs.* We show that for turn-based games, and MDPs, the one-step metric operator H for both bisimulation and simulation can be computed in

polynomial time, via a reduction to linear programming (LP). The only previously known algorithm, which can be inferred from [9], had EXPTIME complexity and relied on a reduction to the theory of real closed fields; the algorithm thus had more a complexity-theoretic, than a practical, value. The key step in obtaining our polynomial-time algorithm consists in transforming the original sup-inf *non-linear* optimization problem (which required the theory of reals) into a quadratic-size inf *linear* optimization problem that can be solved via LP. We then present PSPACE algorithms for both the decision problem of the metric distance between two states and for the problem of computing the approximate metric distance between two states for turn-based games and MDPs. Our algorithms match the complexity of the best known algorithms for the sub-class of Markov chains [24].

2. *Metrics for concurrent games.* For concurrent games, our algorithms for the H operator still rely on decision procedures for the theory of real closed fields, leading to an EXPTIME procedure. However, the algorithms that could be inferred from [9] had time-complexity $\mathcal{O}(|G|^{\mathcal{O}(|G|^7)})$, where $|G|$ is the size of a game; we improve this result by presenting algorithms with $\mathcal{O}(|G|^{\mathcal{O}(|G|^5)})$ time-complexity.

3. *Hardness of metric computation in concurrent games.* We show that computing the exact distance of states of concurrent games is at least as hard as computing the value of concurrent reachability games [13, 7], which is known to be at least as hard as solving the square-root-sum problem in computational geometry [16]. These two problems are known to lie in PSPACE, and have resisted many attempts to show that they are in NP.

4. *Kernel of the metrics.* We present polynomial time algorithms to compute the simulation and bisimulation kernel of the metrics for turn-based games and MDPs. Our algorithm for the bisimulation kernel of the metric runs in time $\mathcal{O}(n^4)$ (assuming a constant number of moves) as compared to the previous known $\mathcal{O}(n^9 \cdot \log(n))$ algorithm of [28] for MDPs, where n is the size of the state space. For concurrent games the simulation and the bisimulation kernel can be computed in time $\mathcal{O}(|G|^{\mathcal{O}(|G|^3)})$, where $|G|$ is the size of a game.

Our formulation of probabilistic simulation and bisimulation differs from the one previously considered for MDPs in [2]: there, the names of moves (called "labels") must be preserved by simulation and bisimulation, so that a move from a state has at most one candidate simulator move at another state. Our problem for MDPs is closer to the one considered in [28], where labels must be preserved, but where a label can be associated with multiple probability distributions (moves).

For turn-based games and MDPs, the algorithms for probabilistic simulation and bisimulation can be obtained from the LP algorithms that yield the metrics. For probabilistic simulation, the algorithm we obtain coincides with the algorithm previously published in [28]. The algorithm requires the solution of feasibility-LP problems with a number of variables and inequalities that is quadratic in the size of the system. For probabilistic bisimulation, we are able to improve on this result by providing an algorithm that requires the solution of feasibility-LP problems that have linearly many variables and constraints. Precisely, as for ordinary bisimulation, the kernel is computed via iterative refinement of a partition of the state space [19]. Given two states that belong to the same partition, to decide whether the states need to be split in the next partition-refinement step, we present an algorithm that requires the solution of a feasibility-LP problem with a number of variables equal to the number of moves available at the states, and number of constraints linear in the number of equivalence classes. Overall, our algorithm for bisimulation runs in time $\mathcal{O}(n^4)$ (assuming a constant number of moves), considerably improving the $\mathcal{O}(n^9 \cdot \log(n))$ algorithm of [28] for MDPs, and providing for the first time a polynomial algorithm for turn-based games.

2 Definitions

Valuations. Let $[\theta_1, \theta_2] \subseteq \mathbb{R}$ be a fixed, non-singleton real interval. Given a set of states S, a *valuation over* S is a function $f : S \mapsto [\theta_1, \theta_2]$ associating with every state $s \in S$ a value $\theta_1 \leq f(s) \leq \theta_2$; we let $\mathcal{F}$ be the set of all valuations. For $c \in [\theta_1, \theta_2]$, we denote by $\mathbf{c}$ the constant valuation such that $\mathbf{c}(s) = c$ at all $s \in S$. We order valuations pointwise: for $f, g \in \mathcal{F}$, we write $f \leq g$ iff $f(s) \leq g(s)$ at all $s \in S$; we remark that $\mathcal{F}$, under $\leq$, forms a lattice. Given $a, b \in \mathbb{R}$, we write $a \sqcup b = \max\{a, b\}$, and $a \sqcap b = \min\{a, b\}$; we extend $\sqcap, \sqcup$ to valuations by interpreting them in pointwise fashion.

Game structures. For a finite set A, let $\mathrm{Dist}(A)$ denote the set of probability distributions over A. We say that $p \in \mathrm{Dist}(A)$ is *deterministic* if there is $a \in A$ such that $p(a) = 1$. We assume a fixed finite set $\mathcal{V}$ of *observation variables*.

A (two-player, concurrent) *game structure* $G = \langle S, [\cdot], \mathit{Moves}, \Gamma_1, \Gamma_2, \delta \rangle$ consists of the following components [1, 7]:

- A finite set S of states.
- A variable interpretation $[\cdot] : \mathcal{V} \mapsto S \mapsto [\theta_1, \theta_2]$, which associates with each variable $v \in \mathcal{V}$ a valuation $[v]$.
- A finite set *Moves* of moves.
- Two move assignments Γ_1, Γ_2: $S \mapsto 2^{\mathit{Moves}} \setminus \emptyset$. For $i \in \{1, 2\}$, the assignment Γ_i associates with each state $s \in S$ the nonempty set $\Gamma_i(s) \subseteq \mathit{Moves}$ of moves available to player i at state s.
- A probabilistic transition function δ: $S \times \mathit{Moves} \times \mathit{Moves} \mapsto \mathrm{Dist}(S)$, that gives the probability $\delta(s, a_1, a_2)(t)$ of a transition from s to t when player 1 plays move a_1 and player 2 plays move a_2.

At every state $s \in S$, player 1 chooses a move $a_1 \in \Gamma_1(s)$, and simultaneously and independently player 2 chooses a move $a_2 \in \Gamma_2(s)$. The game then proceeds to the successor state $t \in S$ with probability $\delta(s, a_1, a_2)(t)$. We let $\mathrm{Dest}(s, a_1, a_2) = \{t \in S \mid \delta(s, a_1, a_2)(t) > 0\}$. The *propositional distance* $p(s, t)$ between two states $s, t \in S$ is the maximum difference in the valuation of any variable:

$$p(s, t) = \max_{v \in \mathcal{V}} |[v](s) - [v](t)| \; .$$

The kernel of the propositional distance induces an equivalence on states: for states s, t, we let $s \equiv t$ if $p(s, t) = 0$. In the following, unless otherwise noted, the definitions refer to a game structure with components $G = \langle S, [\cdot], \mathit{Moves}, \Gamma_1, \Gamma_2, \delta \rangle$. We indicate the opponent of a player $i \in \{1, 2\}$ by $\sim\! i = 3 - i$. We consider the following subclasses of game structures.

Turn-based game structures. A game structure G is *turn-based* if we can write $S = S_1 \cup S_2$ with $S_1 \cap S_2 = \emptyset$ where $s \in S_1$ implies $|\Gamma_2(s)| = 1$, and $s \in S_2$ implies $|\Gamma_1(s)| = 1$, and further, there exists a special variable $\mathit{turn} \in \mathcal{V}$, such that $[\mathit{turn}]s = \theta_1$ iff $s \in S_1$, and $[\mathit{turn}]s = \theta_2$ iff $s \in S_2$.

Markov decision processes. For $i \in \{1, 2\}$, we say that a structure is an i-MDP if $\forall s \in S$, $|\Gamma_{\sim i}(s)| = 1$. For MDPs, we omit the (single) move of the player without a choice of moves, and write $\delta(s, a)$ for the transition function.

Moves and strategies. A *mixed move* is a probability distribution over the moves available to a player at a state. We denote by $\mathcal{D}_i(s) \subseteq \mathrm{Dist}(\mathit{Moves})$ the set of mixed moves available to player $i \in \{1,2\}$ at $s \in S$, where:

$$\mathcal{D}_i(s) = \{\mathcal{D} \in \mathrm{Dist}(\mathit{Moves}) \mid \mathcal{D}(a) > 0 \textit{ implies } a \in \Gamma_i(s)\} .$$

The moves in *Moves* are called *pure moves.* We extend the transition function to mixed moves by defining, for $s \in S$ and $x_1 \in \mathcal{D}_1(s)$, $x_2 \in \mathcal{D}_2(s)$,

$$\delta(s, x_1, x_2)(t) = \sum_{a_1 \in \Gamma_1(s)} \sum_{a_2 \in \Gamma_2(s)} \delta(s, a_1, a_2)(t) \cdot x_1(a_1) \cdot x_2(a_2) .$$

A *path* σ of G is an infinite sequence $s_0, s_1, s_2, ...$ of states in $s \in S$, such that for all $k \geq 0$, there are mixed moves $x_1^k \in \mathcal{D}_1(s_k)$ and $x_2^k \in \mathcal{D}_2(s_k)$ with $\delta(s_k, x_1^k, x_2^k)(s_{k+1}) > 0$. We write Σ for the set of all paths, and Σ_s the set of all paths starting from state s.

A *strategy* for player $i \in \{1,2\}$ is a function $\pi_i : S^+ \mapsto \mathrm{Dist}(\mathit{Moves})$ that associates with every non-empty finite sequence $\sigma \in S^+$ of states, representing the history of the game, a probability distribution $\pi_i(\sigma)$, which is used to select the next move of player i; we require that for all $\sigma \in S^*$ and states $s \in S$, if $\pi_i(\sigma s)(a) > 0$, then $a \in \Gamma_i(s)$. We write Π_i for the set of strategies for player i. Once the starting state s and the strategies π_1 and π_2 for the two players have been chosen, the game is reduced to an ordinary stochastic process, denoted $G_s^{\pi_1,\pi_2}$, which defines a probability distribution on the set Σ of paths. We denote by $\mathrm{Pr}_s^{\pi_1,\pi_2}(\cdot)$ the probability of a measurable event (sets of paths) with respect to this process, and denote by $\mathbb{E}_s^{\pi_1,\pi_2}(\cdot)$ the associated expectation operator. For $k \geq 0$, we let $X_k : \Sigma \to S$ be the random variable denoting the k-th state along a path.

One-step expectations and predecessor operators. Given a valuation $f \in \mathcal{F}$, a state $s \in S$, and two mixed moves $x_1 \in \mathcal{D}_1(s)$ and $x_2 \in \mathcal{D}_2(s)$, we define the expectation of f from s under x_1, x_2 by,

$$\mathbb{E}_s^{x_1,x_2}(f) = \sum_{t \in S} \delta(s, x_1, x_2)(t)\, f(t) .$$

For a game structure G, for $i \in \{1,2\}$ we define the *valuation transformer* $\mathrm{Pre}_i : \mathcal{F} \mapsto \mathcal{F}$ by, for all $f \in \mathcal{F}$ and $s \in S$ as,

$$\mathrm{Pre}_i(f)(s) = \sup_{x_i \in \mathcal{D}_i(s)} \inf_{x_{\sim i} \in \mathcal{D}_{\sim i}(s)} \mathbb{E}_s^{x_i, x_{\sim i}}(f) .$$

Intuitively, $\mathrm{Pre}_i(f)(s)$ is the maximal expectation player i can achieve of f after one step from s: this is the standard "one-day" or "next-stage" operator of the theory of repeated games [15].

Game bisimulation and simulation metrics. A *directed metric* is a function $d : S^2 \mapsto \mathbb{R}_{\geq 0}$ which satisfies $d(s,s) = 0$ and the *triangle inequality* $d(s,t) \leq d(s,u) + d(u,t)$ for all $s, t, u \in S$. We denote by $\mathcal{M} \subseteq S^2 \mapsto \mathbb{R}$ the space of all directed metrics; this space, ordered pointwise, forms a lattice which we indicate with $(\mathcal{M}, \leq)$. Since $d(s,t)$ may be zero for $s \neq t$, these functions are *pseudo-metrics* as per prevailing terminology [25]. In the following, we omit "directed" and simply say metric when the context is clear. For a metric d, we indicate with $C(d)$ the set of valuations $k \in \mathcal{F}$ where $k(s) - k(t) \leq d(s,t)$ for every $s, t \in S$. A metric transformer $H_{\preceq_1} : \mathcal{M} \mapsto \mathcal{M}$ is defined as follows, for all $d \in \mathcal{M}$ and $s, t \in S$:

$$H_{\preceq_1}(d)(s,t) = p(s,t) \sqcup \sup_{k \in C(d)} \big(\mathrm{Pre}_1(k)(s) - \mathrm{Pre}_1(k)(t)\big) . \tag{1}$$

The *player 1 game simulation metric* $[\preceq_1]$ is the least fixpoint of $H_{\preceq_1}$; the *game bisimulation metric* $[\simeq_1]$ is the least symmetrical fixpoint of $H_{\preceq_1}$ and is defined as follows, for all $d \in \mathcal{M}$ and $s, t \in S$:

$$H_{\simeq_1}(d)(s,t) = H_{\preceq_1}(d)(s,t) \sqcup H_{\preceq_1}(d)(t,s) \; . \tag{2}$$

The operator $H_{\preceq_1}$ is monotonic, non-decreasing and continuous in the lattice $(\mathcal{M}, \leq)$. We can therefore compute $H_{\preceq_1}$ using Picard iteration; we denote by $[\preceq_1^n] = H^n_{\preceq_1}(\mathbf{0})$ the n-iterate of this. From the determinacy of concurrent games with respect to ω-regular goals [17], we have that the game bisimulation metric is *reciprocal*, in that $[\simeq_1] = [\simeq_2]$; we will thus simply write $[\simeq_g]$. Similarly, for all $s, t \in S$ we have $[s \preceq_1 t] = [t \preceq_2 s]$.

The main result in [9] about these metrics is that they are logically characterized by the quantitative μ-calculus of [8]. We omit the formal definition of the syntax and semantics of the quantitative μ-calculus; we refer the reader to [8] for details. Given a game structure G, every closed formula ϕ of the quantitative μ-calculus defines a valuation $[\![\phi]\!] \in \mathcal{F}$. Let $q\mu$ (respectively, $q\mu_1^+$) consist of all quantitative μ-calculus formulas (respectively, all quantitative μ-calculus formulas with only the Pre_1 operator and all negations before atomic propositions). The result of [9] shows that for all states $s, t \in S$,

$$[s \preceq_1 t] = \sup_{\phi \in q\mu_1^+} ([\![\phi]\!](s) - [\![\phi]\!](t)) \qquad\qquad [s \simeq_g t] = \sup_{\phi \in q\mu} |[\![\phi]\!](s) - [\![\phi]\!](t)| \; . \tag{3}$$

Metric kernels. The kernel of the metric $[\simeq_g]$ defines an equivalence relation $\simeq_g$ on the states of a game structure: $s \simeq_g t$ iff $[s \simeq_g t] = 0$; the relation $\simeq_g$ is called the *game bisimulation* relation [9]. Similarly, we define the *game simulation* preorder $s \preceq_1 t$ as the kernel of the directed metric $[\preceq_1]$, that is, $s \preceq_1 t$ iff $[s \preceq_1 t] = 0$. For notational convenience, given a relation $R \subseteq S \times S$, we denote by $1_R : S \times S \mapsto \{0, 1\}$ its characteristic set, defined as $1_R(s,t) = 1$ iff $(s,t) \in R$. Given a relation $R \subseteq S \times S$, let $B(R) \subseteq \mathcal{F}$ consist of all valuations $k \in \mathcal{F}$ such that, for all $s, t \in S$, if sRt then $k(s) \leq k(t)$.

3 Bounds for Average and Discounted Payoff Games

From (3) it follows that the game bisimulation metric provides a tight bound for the difference in valuations of quantitative μ-calculus formulas. In this section, we show that the game bisimulation metric also provides a bound for the difference in average and discounted value of games. This lends further support for the game bisimulation metric, and its kernel, the game bisimulation relation, being the canonical game metrics and relations.

Discounted payoff games. Let π_1 and π_2 be strategies of player 1 and player 2 respectively. Let $\alpha \in [0, 1)$ be a discount factor. The α-*discounted* payoff $v_1(s, \pi_1, \pi_2)$ for player 1 at a state s for a variable $r \in \mathcal{V}$ and the strategies π_1 and π_2 is defined as

$$v_1^\alpha(s, \pi_1, \pi_2) = (1-\alpha) \cdot \sum_{n=0}^{\infty} \alpha^n \cdot \mathbb{E}_s^{\pi_1,\pi_2}\big([r](X_n)\big) \; . \tag{4}$$

The discounted payoff for player 2 is defined as $v_2^\alpha(s, \pi_1, \pi_2) = -v_1^\alpha(s, \pi_1, \pi_2)$. Thus, player 1 wins (and player 2 loses) the "discounted sum" of the valuations of r along the path, where the discount factor weighs future rewards with the discount α. Given a state $s \in S$, we are interested in finding the maximal payoff $v_i^\alpha(s)$ that player i can ensure against all opponent strategies, when the game starts from state $s \in S$. This maximal payoff is given by:

$$w_i^\alpha(s) = \sup_{\pi_i \in \Pi_i} \inf_{\pi_{\sim i} \in \Pi_{\sim i}} v_i(s, \pi_i, \pi_{\sim i}) \ .$$

These values can be computed as the limit of the sequence of α-discounted, n-step rewards, for $n \to \infty$. For $i \in \{1, 2\}$, we define a sequence of valuations $w_i^\alpha(0)(s)$, $w_i^\alpha(1)(s)$, $w_i^\alpha(2)(s)$, ... as follows: for all $s \in S$ and $n \geq 0$:

$$w_i^\alpha(n+1)(s) = (1-\alpha) \cdot [r](s) + \alpha \cdot \mathrm{Pre}_i(w_i^\alpha(n))(s) \ . \tag{5}$$

where the initial valuation $w_i^\alpha(0)$ is arbitrary. Shapley proved that $w_i^\alpha = \lim_{n\to\infty} w_i^\alpha(n)$ [22].

Average payoff games. Let π_1 and π_2 be strategies of player 1 and player 2 respectively. The *average* payoff $v_1(s, \pi_1, \pi_2)$ for player 1 at a state s for a variable $r \in \mathcal{V}$ and the strategies π_1 and π_2 is defined as

$$v_1(s, \pi_1, \pi_2) = \lim_{n\to\infty} \inf \frac{1}{n} \sum_{k=0}^{n-1} \mathbb{E}_s^{\pi_1,\pi_2}\big([r](X_k)\big) \ . \tag{6}$$

The reward for player 2 is $v_2(s, \pi_1, \pi_2) = -v_1(s, \pi_1, \pi_2)$. A game structure G with average payoff is called an average reward game. The *average value* of the game G at s for player $i \in \{1, 2\}$ is defined as

$$w_i(s) = \sup_{\pi_i \in \Pi_i} \inf_{\pi_{\sim i} \in \Pi_{\sim i}} v_i(s, \pi_i, \pi_{\sim i}) \ .$$

Mertens and Neyman established the determinacy of average reward games, and showed that the limit of the discounted value of a game as all the discount factors tend to 1 is the same as the average value of the game: for all $s \in S$ and $i \in \{1, 2\}$, we have $\lim_{\alpha \to 1} w_i^\alpha(s) = w_i(s)$ [18]. It is easy to show that the average value of a game is a valuation.

Metrics for discounted and average payoffs. We show that the game simulation metric $[\preceq_1]$ provides a bound for discounted and long-run rewards. In the following we consider player 1 rewards (the case for player 2 is identical). Our first result is that the difference in discounted rewards is bounded by the metric.

Theorem 1 *For all α-discounted rewards w^α, we have that $w^\alpha(s) - w^\alpha(t) \leq [s \preceq_1 t]$ and $|w^\alpha(s) - w^\alpha(t)| \leq [s \simeq_g t]$.*

Proof As the metric can be computed via Picard iteration, we have for all $n \geq 0$:

$$[s \preceq_1^n t] = p(s, t) \sqcup \sup_{k \in C([\preceq_1^{n-1}])} (\mathrm{Pre}_1(k)(s) - \mathrm{Pre}_1(k)(t)) \ . \tag{7}$$

We prove by induction on $n \geq 0$ that $w^\alpha(n)(s) - w^\alpha(n)(t) \leq [s \preceq_1^n t]$. The base case is immediate. Assume the result holds for $n - 1 \geq 0$. We have:

$$\begin{aligned} w^\alpha(n)(s) - w^\alpha(n)(t) &= (1-\alpha) \cdot [r](s) + \alpha \cdot \mathrm{Pre}_1(w^\alpha(n-1))(s) - \\ &\qquad (1-\alpha) \cdot [r](t) \ - \alpha \cdot \mathrm{Pre}_1(w^\alpha(n-1))(t) \\ &= (1-\alpha) \cdot \big([r](s) - [r](t)\big) \ + \\ &\qquad \alpha \cdot \big(\mathrm{Pre}_1(w^\alpha(n-1))(s) - \mathrm{Pre}_1(w^\alpha(n-1))(t)\big) \\ &\leq (1-\alpha) \cdot p(s,t) + \alpha \cdot [s \preceq_1^n t] \ \ \leq \ \ [s \preceq_1^n t], \end{aligned}$$

where the last step follows by (7), since by the induction hypothesis we have $w^\alpha(n-1) \in C([\preceq_1^{n-1}])$. ■

Using the fact that the limit of the discounted reward, for the discount factor that approaches 1, is equal to the average reward, we obtain that the metrics provide a bound for the difference of average values as well.

Theorem 2 *For all game structures G and states s and t, we have $w(s) - w(t) \leq [s \preceq_1 t]$ and $|w(s) - w(t)| \leq [s \simeq_g t]$.*

4 Algorithms for Turn-Based Games and MDPs

In this section, we present algorithms for computing the metric and its kernel for turn-based games and MDPs. We first present a polynomial time algorithm to compute the operator $H_{\preceq_i}(d)$ that gives the *exact* one-step distance between two states, for $i \in \{1,2\}$. We then present a PSPACE algorithm to decide whether the limit distance between two states s and t (i.e., $[s \preceq_1 t]$) is at most a rational value r. Our algorithm matches the best known bound known for the special class of Markov chains [24]. Finally, we present improved algorithms for the important case of the kernel of the metrics. For the bisimulation kernel our algorithm is significantly more efficient compared to previous algorithms.

4.1 Algorithms for the metrics

For turn-based games and MDPs, only one player has a choice of moves at a given state. We consider two player 1 states. A similar analysis applies to player 2 states. We remark that the distance between states in S_i and $S_{\sim i}$ is always $\theta_2 - \theta_1$ due to the existence of the variable *turn*. For a metric $d \in \mathcal{M}$, and states $s, t \in S_1$, computing $H_{\preceq_1}(d)(s,t)$, given that $p(s,t)$ is trivially computed by its definition, entails evaluating the expression, $\sup_{k \in C(d)} \sup_{x \in \mathcal{D}_1(s)} \inf_{y \in \mathcal{D}_1(t)} (\mathbb{E}_s^x(k) - \mathbb{E}_t^y(k))$. By expanding the expectations, we get the following form,

$$\sup_{k \in C(d)} \sup_{x \in \mathcal{D}_1(s)} \inf_{y \in \mathcal{D}_1(t)} \left(\sum_{u \in S} \sum_{a \in \Gamma_1(s)} \delta(s,a)(u) \cdot x(a) \cdot k(u) - \sum_{v \in S} \sum_{b \in \Gamma_1(t)} \delta(t,b)(v) \cdot y(b) \cdot k(v) \right) . \quad (8)$$

We observe that the one-step distance as defined in (8) is a *sup-inf non-linear (quadratic)* optimization problem. Using the following lemma we transform (8) to an *inf linear* optimization problem, which we solve by linear programming.

Lemma 1 *For all turn-based game structures G, for all player i states s and t, given a metric $d \in \mathcal{M}$, the following equality holds,*

$$\sup_{k \in C(d)} \sup_{x \in \mathcal{D}_i(s)} \inf_{y \in \mathcal{D}_i(t)} (\mathbb{E}_s^x(k) - \mathbb{E}_t^y(k)) = \sup_{a \in \Gamma_i(s)} \inf_{y \in \mathcal{D}_i(t)} \sup_{k \in C(d)} (\mathbb{E}_s^a(k) - \mathbb{E}_t^y(k)) .$$

Proof We prove the result for player 1 states s and t, with the proof being identical for player 2. Given a metric $d \in \mathcal{M}$, we have,

$$
\begin{aligned}
\sup_{k \in C(d)} \sup_{x \in \mathcal{D}_1(s)} \inf_{y \in \mathcal{D}_1(t)} (\mathbb{E}_s^x(k) - \mathbb{E}_t^y(k)) &= \sup_{k \in C(d)} (\sup_{x \in \mathcal{D}_1(s)} \mathbb{E}_s^x(k) - \sup_{y \in \mathcal{D}_1(t)} \mathbb{E}_t^y(k)) \\
&= \sup_{k \in C(d)} (\sup_{a \in \Gamma_1(s)} \mathbb{E}_s^a(k) - \sup_{y \in \mathcal{D}_1(t)} \mathbb{E}_t^y(k)) \qquad (9) \\
&= \sup_{k \in C(d)} \sup_{a \in \Gamma_1(s)} \inf_{y \in \mathcal{D}_1(t)} (\mathbb{E}_s^a(k) - \mathbb{E}_t^y(k)) \\
&= \sup_{a \in \Gamma_1(s)} \sup_{k \in C(d)} \inf_{y \in \mathcal{D}_1(t)} (\mathbb{E}_s^a(k) - \mathbb{E}_t^y(k)) \qquad (10) \\
&= \sup_{a \in \Gamma_1(s)} \inf_{y \in \mathcal{D}_1(t)} \sup_{k \in C(d)} (\mathbb{E}_s^a(k) - \mathbb{E}_t^y(k)) \qquad (11)
\end{aligned}
$$

For a fixed $k \in C(d)$, since pure optimal strategies exist at each state for turn-based games and MDPs, we replace the $\sup_{x \in \mathcal{D}_1(s)}$ with $\sup_{a \in \Gamma_1(s)}$ yielding (9). Since the difference in expectations is multi-linear, $y \in \mathcal{D}_1(t)$ is a probability distribution and $C(d)$ is a compact convex set, we can use the generalized minimax theorem [23], and interchange the innermost sup inf to get (11) from (10). ■

Therefore, given $d \in \mathcal{M}$, we can write the player 1 one-step distance between states s and t as follows,

$$\mathsf{OneStep}(s, t, d) = \sup_{a \in \Gamma_1(s)} \inf_{y \in \mathcal{D}_1(t)} \sup_{k \in C(d)} (\mathbb{E}_s^a(k) - \mathbb{E}_t^y(k)) \, . \qquad (12)$$

Hence we compute for all $a \in \Gamma_1(s)$, the expression $\mathsf{OneStep}(s, t, d, a) = \inf_{y \in \mathcal{D}_1(t)} \sup_{k \in C(d)} (\mathbb{E}_s^a(k) - \mathbb{E}_t^y(k))$, and then choose the maximum, i.e., $\max_{a \in \Gamma_1(s)} \mathsf{OneStep}(s, t, d, a)$. We now present a lemma that helps reduce the above $\inf - \sup$ optimization problem to a linear program. We first introduce some notation. We denote by λ the set of variables $\lambda_{u,v}$, for $u, v \in S$. Given $d \in \mathcal{M}$, $a \in \Gamma_1(s)$, and a distribution $y \in \mathcal{D}_1(t)$, we write $\lambda \in \Phi(d, a, y)$ if the following linear constraints are satisfied:

$$(1) \text{ for all } v \in S : \sum_{u \in S} \lambda_{u,v} = \delta(s, a)(v); \qquad (2) \text{ for all } u \in S : \sum_{v \in S} \lambda_{u,v} = \sum_{b \in \Gamma_1(t)} y(b) \cdot \delta(t, b)(u);$$

$$(3) \text{ for all } u, v \in S : \lambda_{u,v} \geq 0 \, .$$

Lemma 2 *For all turn-based games and MDPs, for all $d \in \mathcal{M}$, and for all $s, t \in S$, the following assertion holds:*

$$\sup_{a \in \Gamma_1(s)} \inf_{y \in \mathcal{D}_1(t)} \sup_{k \in C(d)} (\mathbb{E}_s^a(k) - \mathbb{E}_t^y(k)) = \sup_{a \in \Gamma_1(s)} \inf_{y \in \mathcal{D}_1(t)} \inf_{\lambda \in \Phi(d,a,y)} \Big(\sum_{u,v \in S} d(u, v) \cdot \lambda_{u,v} \Big) \, .$$

Proof From the LP duality based results of [25], we have that for all $a \in \Gamma_1(s)$ and $y \in \mathcal{D}_1(t)$,

$$\sup_{k \in C(d)} (\mathbb{E}_s^a(k) - \mathbb{E}_t^y(k)) = \inf_{\lambda \in \Phi(d,a,y)} \Big(\sum_{u,v \in S} d(u, v) \cdot \lambda_{u,v} \Big) \, .$$

The formula on the right hand side of the above equality is the *trans-shipping formulation*, which solves for the minimum cost of shipping the distribution $\delta(s, a)$ into $\delta(t, y)$, with edge costs d. The result of the lemma follows. ■

Using the above result we obtain the following LP for $\mathsf{OneStep}(s,t,d,a)$ over the variables: (a) $\{\lambda_{u,v}\}_{u,v\in S}$, and (b) y_b for $b \in \Gamma_1(t)$:

$$\text{Minimize} \quad \sum_{u,v\in S} d(u,v)\cdot\lambda_{u,v} \quad \text{subject to} \tag{13}$$

$$\text{(1) for all } v\in S : \sum_{u\in S}\lambda_{u,v} = \delta(s,a)(v); \qquad \text{(2) for all } u\in S : \sum_{v\in S}\lambda_{u,v} = \sum_{b\in\Gamma_1(t)} y_b\cdot\delta(t,b)(u);$$

$$\text{(3) for all } u,v\in S : \lambda_{u,v}\geq 0; \qquad \text{(4) for all } b\in\Gamma_1(t) : y_b \geq 0; \qquad \text{(5)} \sum_{b\in\Gamma_1(t)} y_b = 1\ .$$

Theorem 3 *For all turn-based games and MDPs, given $d \in \mathcal{M}$, for all states $s,t\in S$, we can compute $H_{\preceq_1}(d)(s,t)$ in polynomial time by the LP (13).*

Iteration of $\mathsf{OneStep}(s,t,d)$ converges to the exact distance. However, in general, there are no known bounds for the rate of convergence. We now present a decision procedure to check whether the exact distance between two states is at most a rational value r. We first show how to express the predicate $d = \mathsf{OneStep}(s,t,d)$, for a given $d\in\mathcal{M}$. We observe that since $H_{\preceq_1}$ is non-decreasing, we have $\mathsf{OneStep}(s,t,d)\geq d$. It follows that the equality $d = \mathsf{OneStep}(s,t,d)$ holds iff all the linear inequalities of LP (13) are satisfied, and $d(s,t) = \sum_{u,v\in S} d(u,v)\cdot\lambda_{u,v}$ holds. It then follows that $d = \mathsf{OneStep}(s,t,d)$ can be written as a predicate in the theory of real closed fields. Given a rational r, two states s and t, we present an existential theory of reals formula to decide whether $[s \preceq_1 t] \leq r$. Since $[s\preceq_1 t]$ is the least fixed point of $H_{\preceq_1}$, we define a formula $\Phi(r)$ that is true iff $[s\preceq_1 t]\leq r$, as follows:

$$\exists d\in\mathcal{M}.[(\mathsf{OneStep}(s,t,d) = d)\wedge(d(s,t)\leq r)]\ .$$

If the formula $\Phi(r)$ is true, then there exists a fixpoint that is bounded by r, which implies that the least fixpoint is bounded by r. Conversely, if the least fixpoint is bounded by r, then the least fixpoint is a witness d for $\Phi(r)$ being true. Since the existential theory of reals is decidable in PSPACE [6], we have the following result.

Theorem 4 (Decision complexity for exact distance). *For all turn-based games and MDPs, given a rational r, and two states s and t, whether $[s\preceq_1 t]\leq r$ can be decided in PSPACE.*

Given a rational $\epsilon > 0$, using binary search and $\mathcal{O}(\log(\frac{\theta_2-\theta_1}{\epsilon}))$ calls to check the formula $\Phi(r)$, we can obtain an interval $[l,u]$ with $u-l\leq\epsilon$ such that $[s\preceq_1 t]$ lies in the interval $[l,u]$.

Corollary 1 (Approximation for exact distance). *For all turn-based games and MDPs, given a rational ϵ, and two states s and t, an interval $[l,u]$ with $u-l\leq\epsilon$ such that $[s\preceq_1 t]\in[l,u]$ can be computed in PSPACE.*

4.2 Algorithms for the kernel

The kernel of the simulation metric $\preceq_1$ can be computed as the limit of the series $\preceq_1^0$, $\preceq_1^1$, $\preceq_1^2$, …, of relations. For all $s,t\in S$, we have $(s,t)\in\preceq_1^0$ iff $s\equiv t$. For all $n\geq 0$, we have $(s,t)\in\preceq_1^{n+1}$ iff $\mathsf{OneStep}(s,t,1_{\preceq_1^n}) = 0$. Checking the condition $\mathsf{OneStep}(s,t,1_{\preceq_1^n}) = 0$, corresponds to solving an LP feasibility problem for every $a\in\Gamma_1(s)$, as it suffices to replace the minimization goal

$\gamma = \sum_{u,v \in S} 1_{\preceq_1^n}(u,v) \cdot \lambda_{u,v}$ with the constraint $\gamma = 0$ in the LP (13). We note that this is the same LP feasibility problem that was introduced in [28] as part of an algorithm to decide simulation of probabilistic systems in which each label may lead to one or more distributions over states.

For the bisimulation kernel, we present a more efficient algorithm, which also improves on the algorithms presented in [28]. The idea is to proceed by partition refinement, as usual for bisimulation computations. The refinement step is as follows: given a partition, two states s and t belong to the same refined partition iff every pure move from s induces a probability distribution on equivalence classes that can be matched by mixed moves from t, and vice versa. Precisely, we compute a sequence $\mathcal{Q}^0$, $\mathcal{Q}^1$, $\mathcal{Q}^2$, ..., of partitions. Two states s, t belong to the same class of $\mathcal{Q}^0$ iff they have the same variable valuation (i.e., iff $s \equiv t$). For $n \geq 0$, since by the definition of the bisimulation metric given in (2), $[s \simeq_g t] = 0$ iff $[s \preceq_1 t] = 0$ and $[t \preceq_1 s] = 0$, two states s, t in a given class of $\mathcal{Q}^n$ remain in the same class in $\mathcal{Q}^{n+1}$ iff both (s,t) and (t,s) satisfy the set of feasibility LP problems $\mathsf{OneStepBis}(s,t,\mathcal{Q}^n)$ as given below:

> $\mathsf{OneStepBis}(s,t,\mathcal{Q})$ consists of one feasibility LP problem for each $a \in \Gamma(s)$. The problem for $a \in \Gamma(s)$ has set of variables $\{x_b \mid b \in \Gamma(t)\}$, and set of constraints:
>
> $$(1) \text{ for all } b \in \Gamma(t): \ x_b \geq 0, \qquad (2) \sum_{b \in \Gamma(t)} x_b = 1,$$
>
> $$(3) \text{ for all } V \in \mathcal{Q}: \sum_{b \in \Gamma(t)} \sum_{u \in V} x_b \cdot \delta(t,b)(u) \geq \sum_{u \in V} \delta(s,a)(u) \ .$$

Complexity. The number of partition refinement steps required for the computation of both the simulation and the bisimulation kernel is bounded by $O(|S|^2)$ for turn-based games and MDPs, where S is the set of states. At every refinement step, at most $O(|S|^2)$ state pairs are considered, and for each state pair (s,t) at most $|\Gamma(s)|$ LP feasibility problems needs to be solved. Let us denote by $\mathsf{LPF}(n,m)$ the complexity of solving the feasibility of m linear inequalities over n variables. We obtain the following result.

Theorem 5 *For all turn-based games and MDPs G, the following assertions hold:*

1. *the simulation kernel can be computed in $\mathcal{O}\big(n^4 \cdot m \cdot \mathsf{LPF}(n^2+m, n^2+2n+m+2)\big)$ time;*
2. *the bisimulation kernel can be computed in $\mathcal{O}\big(n^4 \cdot m \cdot \mathsf{LPF}(m, n+m+1)\big)$ time;*

where $n = |S|$ is the size of the state space, and $m = \max_{s \in S} |\Gamma(s)|$.

Remark 1 The best known algorithm for $\mathsf{LPF}(n,m)$ works in time $\mathcal{O}(n^{2.5} \cdot \log(n))$ [27] (assuming each arithmetic operation takes unit time). The previous algorithm for the bisimulation kernel checked two way simulation and hence has the complexity $\mathcal{O}(n^4 \cdot m \cdot (n^2+m)^{2.5} \cdot \log(n^2+m))$, whereas our algorithm works in time $\mathcal{O}(n^4 \cdot m \cdot m^{2.5} \cdot \log(m))$. For most practical purposes, the number of moves at a state is constant (i.e., m is constant). For the case when m is constant, the previous best known algorithm worked in $\mathcal{O}(n^9 \cdot \log(n))$ time, whereas our algorithm works in time $\mathcal{O}(n^4)$.

5 Algorithms for Concurrent Games

In this section we first show that the computation of the metric distance is at least as hard as the computation of optimal values in concurrent reachability games. The exact complexity of the latter is open, but it is known to be at least as hard as the square-root sum problem, which is in PSPACE but whose inclusion in NP is a long-standing open problem [14, 16]. Next, we present algorithms based on a decision procedure for the theory of real closed fields, for both checking the bounds of the exact distance and the kernel of the metrics. Our reduction to the theory of real closed fields removes one quantifier alternation when compared to the previous known formula (inferred from [9]). This improves the complexity of the algorithm.

5.1 Reduction of reachability games to metrics

We will use the following terms in the result. A *proposition* is a boolean observation variable, and we say a state is labeled by a proposition q iff q is true at s. A state t is *absorbing* in a concurrent game, if both players have only one action available at t, and the next state of t is always t (it is a state with a self-loop). For a proposition q, let $\Diamond q$ denote the set of paths that visit a state labeled by q at least once. In concurrent reachability games, the objective is $\Diamond q$, for a proposition q, and without loss of generality all states labeled by q are absorbing states.

Theorem 6 *Consider a concurrent game structure G, with a single proposition q, such that all states labeled by q are absorbing states. We can construct in linear-time a concurrent game structure G', with one additional state t', such that for all $s \in S$, we have $[s \preceq_1 t'] = \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathrm{Pr}_s^{\pi_1,\pi_2}(\Diamond q)$.*

Proof The concurrent game structure G' is obtained from G by adding an absorbing state t'. To prove the desired claim we show that for all $s \in S$ we have $[s \preceq_1 t'] = \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathrm{Pr}_s^{\pi_1,\pi_2}(\Diamond q)$. The states that are not labeled by q are labeled by its complement $\neg q$, and we label the additional state t' by the negation of q (i.e., $\neg q$). Observe there is only one proposition sequence from t', and it is $(\neg q)^\omega$. From a state s in G the possible proposition sequences can be expressed as the following ω-regular expression: $(\neg q)^\omega \cup (\neg q)^* \cdot q^\omega$. Since the proposition sequence from t' is $(\neg q)^\omega$, the supremum of the difference in values over $q\mu$ formulas at s and t' is obtained by satisfying the set of paths formalized as $(\neg q)^* \cdot q^\omega$ at s. The set of paths defined as $(\neg q)^* \cdot q^\omega$ is the same as reaching q in any number of steps, since all states labeled by q are absorbing. Hence $\sup_{\phi \in q\mu^+}([\![\phi]\!](s) - [\![\phi]\!](t')) = [\![\mu X.(q \sqcup \mathrm{Pre}_1(X))]\!](s)$. It follows from the results of [8] that for all $s \in S$ we have $[\![\mu X.(q \sqcup \mathrm{Pre}_1(X))]\!](s) = \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathrm{Pr}_s^{\pi_1,\pi_2}(\Diamond q)$. From the above equalities and the logical characterization result (3) we obtain the desired result. ■

5.2 Algorithms for the metrics

We first prove a lemma that helps obtain reduced-complexity algorithms for concurrent games. The lemma states that the distance $[s \preceq_1 t]$ is attained by restricting player 2 to pure moves at state t, for all states $s, t \in S$.

Lemma 3 *Given a game structure G and a distance $d \in \mathcal{M}$, we have*

$$\sup_{k \in C(d)} \sup_{x_1 \in \mathcal{D}_1(s)} \inf_{y_1 \in \mathcal{D}_1(t)} \sup_{y_2 \in \mathcal{D}_2(t)} \inf_{x_2 \in \mathcal{D}_2(s)} (\mathbb{E}_s^{x_1,x_2}(k)) - \mathbb{E}_t^{y_1,y_2}(k))$$
$$= \sup_{k \in C(d)} \sup_{x_1 \in \mathcal{D}_1(s)} \inf_{y_1 \in \mathcal{D}_1(t)} \sup_{b \in \Gamma_2(t)} \inf_{x_2 \in \mathcal{D}_2(s)} (\mathbb{E}_s^{x_1,x_2}(k) - \mathbb{E}_t^{y_1,b}(k)) \ . \quad (14)$$

Proof To prove our claim we fix $k \in C(d)$, and player 1 mixed moves $x \in \mathcal{D}_1(s)$, and $y \in \mathcal{D}_1(t)$. We then have,

$$\sup_{y_2 \in \mathcal{D}_2(t)} \inf_{x_2 \in \mathcal{D}_2(s)} (\mathbb{E}_s^{x,x_2}(k)) - \mathbb{E}_t^{y,y_2}(k)) = \inf_{x_2 \in \mathcal{D}_2(s)} \mathbb{E}_s^{x,x_2}(k) - \inf_{y_2 \in \mathcal{D}_2(t)} \mathbb{E}_t^{y,y_2}(k) \quad (15)$$
$$= \inf_{x_2 \in \mathcal{D}_2(s)} \mathbb{E}_s^{x,x_2}(k) - \inf_{b \in \Gamma_2(t)} \mathbb{E}_t^{y,b}(k) \quad (16)$$
$$= \sup_{b \in \Gamma_2(t)} \inf_{x_2 \in \mathcal{D}_2(s)} (\mathbb{E}_s^{x,x_2}(k) - \mathbb{E}_t^{y,b}(k)),$$

where (16) follows from (15) since the decomposition on the rhs of (15) yields two independent linear optimization problems; the optimal values are attained at a vertex of the convex hulls of the distributions induced by pure player 2 moves at the two states. This easily leads to the result. ■

We now present algorithms for metrics in concurrent games. Due to the reduction from concurrent reachability games, shown in Theorem 6, it is unlikely that we have an algorithm in NP for the metric distance between states. We therefore construct statements in the theory of real closed fields, firstly to decide whether $[s \preceq_1 t] \leq r$, for a rational r, so that we can approximate the metric distance between states s and t, and secondly to decide if $[s \preceq_1 t] = 0$ in order to compute the kernel of the game simulation and bisimulation metrics.

The statements improve on the complexity that can be achieved by a direct translation of the statements of [9] to the theory of real closed fields. The complexity reduction is based on the observation that using Lemma 3, we can replace a sup operator with finite conjunction, and therefore reduce the quantifier complexity of the resulting formula. Fix a game structure G and states s and t of G. We proceed to construct a statement in the theory of reals that can be used to decide if $[s \preceq_1 t] \leq r$, for a given rational r.

In the following, we use variables x_1, y_1 and x_2 to denote a set of variables $\{x_1(a) \mid a \in \Gamma_1(s)\}$, $\{y_1(a) \mid a \in \Gamma_1(t)\}$ and $\{x_2(b) \mid b \in \Gamma_2(s)\}$ respectively. We use k to denote the set of variables $\{k(u) \mid u \in S\}$, and d for the set of variables $\{d(u,v) \mid u,v \in S\}$. The variables $\alpha, \alpha', \beta, \beta'$ range over reals. For convenience, we assume $\Gamma_2(t) = \{b_1, \ldots, b_l\}$.

First, notice that we can write formulas that state that a variable x is a mixed move for a player at state s, and k is a constructible predicate (i.e., $k \in C(d)$):

$$\mathsf{IsDist}(x, \Gamma_1(s)) \equiv \bigwedge_{a \in \Gamma_1(s)} x(a) \geq 0 \wedge \bigwedge_{a \in \Gamma_1(s)} x(a) \leq 1 \wedge \sum_{a \in \Gamma_1(s)} x(a) = 1$$
$$\mathsf{kBounded}(k, d) \equiv \bigwedge_{u \in S} \Big[k(u) \geq \theta_1 \wedge k(u) \leq \theta_2 \Big] \wedge \bigwedge_{u,v \in S} (k(u) - k(v) \leq d(u,v)) \ .$$

In the following, we write bounded quantifiers of the form "$\exists x_1 \in \mathcal{D}_1(s)$" or "$\forall k \in C(d)$" which mean respectively $\exists x_1 . \mathsf{IsDist}(x_1, \Gamma_1(s)) \wedge \cdots$ and $\forall k . \mathsf{kBounded}(k, d) \rightarrow \cdots$.

Let $\eta(k, x_1, x_2, y_1, b)$ be the polynomial $\mathbb{E}_s^{x_1,x_2}(k) - \mathbb{E}_t^{y_1,b}(k)$. Notice that η is a polynomial of degree 3. We write $a = \max\{a_1, \ldots, a_l\}$ for variables $a, a_1, \ldots, a_l$ for the formula

$$(a = a_1 \wedge \bigwedge_{i=1}^{l} a_1 \geq a_i) \vee \ldots \vee (a = a_l \wedge \bigwedge_{i=1}^{l} a_l \geq a_i) \; .$$

We construct the formula for game simulation in stages. First, we construct a formula $\Phi_1(d, k, x, \alpha)$ with free variables d, k, x, α such that $\Phi_1(d, k, x_1, \alpha)$ holds for a valuation to the variables iff

$$\alpha = \inf_{y_1 \in \mathcal{D}_1(t)} \sup_{b \in \Gamma_2(t)} \inf_{x_2 \in \mathcal{D}_2(s)} (\mathbb{E}_s^{x_1,x_2}(k) - \mathbb{E}_t^{y_1,b}(k)) \; .$$

We use the following observation to move the innermost inf ahead of the sup over the finite set $\Gamma_2(t)$ (for a function f):

$$\sup_{b \in \Gamma_2(t)} \inf_{x_2 \in \mathcal{D}_2(s)} f(b, x_2, x) = \inf_{x_2^{b_1} \in \mathcal{D}_2(s)} \cdots \inf_{x_2^{b_l} \in \mathcal{D}_2(s)} \max(f(b_1, x_2^{b_1}, x), \ldots, f(b_l, x_2^{b_l}, x)) \; .$$

The formula $\Phi_1(d, k, x_1, \alpha)$ is given by:

$$\begin{aligned}
&\forall y_1 \in \mathcal{D}_1(t).\forall x_2^{b_1} \in \mathcal{D}_2(s) \ldots x_2^{b_l} \in \mathcal{D}_2(s).\forall w_1 \ldots w_l.\forall a.\forall \alpha'. \\
&\qquad \exists \hat{y}_1 \in \mathcal{D}_1(t).\exists \hat{x}_2^{b_1} \in \mathcal{D}_2(s) \ldots \hat{x}_2^{b_l} \in \mathcal{D}_2(s).\exists \hat{w}_1 \ldots \hat{w}_l.\exists \hat{a}. \\
&\qquad \left[\left\{ \begin{array}{c} \Big(w_1 = \eta(k, x_1, x_2^{b_1}, y_1, b_1)\Big) \\ \wedge \cdots \wedge \\ \Big(w_l = \eta(k, x_1, x_2^{b_l}, y_1, b_l)\Big) \wedge \\ (a = \max\{w_1, \ldots, w_l\}) \end{array} \right\} \to (a \geq \alpha) \right] \wedge \\
&\qquad\qquad \left[\left\{ \begin{array}{c} \Big(\hat{w}_1 = \eta(k, x_1, \hat{x}_2^{b_1}, \hat{y}_1, b_1)\Big) \\ \wedge \cdots \wedge \\ \Big(\hat{w}_l = \eta(k, x_1, \hat{x}_2^{b_l}, \hat{y}_1, b_l)\Big) \wedge \\ (\hat{a} = \max\{\hat{w}_1, \ldots, \hat{w}_l\} \wedge \hat{a} \geq \alpha') \end{array} \right\} \to (\alpha \geq \alpha') \right] .
\end{aligned}$$

Using Φ_1, we construct a formula $\Phi(d, \alpha)$ with free variables d and α such that $\Phi(d, \alpha)$ is true iff:

$$\alpha = \sup_{k \in C(d)} \sup_{x_1 \in \mathcal{D}_1(s)} \inf_{y_1 \in \mathcal{D}_1(t)} \sup_{b \in \Gamma_2(t)} \inf_{x_2 \in \mathcal{D}_2(s)} (\mathbb{E}_s^{x_1,x_2}(k) - \mathbb{E}_t^{y_1,b}(k)) \; .$$

The formula Φ is defined as follows:

$$\begin{aligned}
&\forall k \in C(d).\forall x_1 \in \mathcal{D}_1(s).\forall \beta.\forall \alpha'. \\
&\qquad \left[\begin{array}{c} \Phi_1(d, k, x_1, \beta) \to (\beta \leq \alpha) \wedge \\ (\forall k' \in C(d).\forall x_1' \in \mathcal{D}_1(s).\forall \beta'.\Phi_1(d, k', x_1', \beta') \wedge \beta' \leq \alpha') \to \alpha \leq \alpha' \end{array} \right] .
\end{aligned} \tag{17}$$

Finally, given a rational r, we can check if $[s \preceq_1 t] \leq r$ by checking if the following sentence is true:

$$\exists d \in \mathcal{M}.\exists a \in \mathcal{M}.[\Phi(d, a) \wedge (d = a) \wedge (d(s, t) \leq r)] \; . \tag{18}$$

The above sentence is true iff the least fixpoint is bounded by r. Like in the case of turn-based games and MDPs, given a rational $\epsilon > 0$, using binary search and $\mathcal{O}(\log(\frac{\theta_2 - \theta_1}{\epsilon}))$ calls to a decision procedure to check the sentence (18), we can compute an interval $[l, u]$ with $u - l \leq \epsilon$, such that $[s \preceq_1 t] \in [l, u]$.

Complexity. Note that Φ is of the form $\forall\exists\forall$, because Φ_1 is of the form $\forall\exists$, and appears in negative position in Φ. The formula Φ has $(|S| + |\Gamma_1(s)| + 3)$ universally quantified variables, followed by $(|S| + |\Gamma_1(s)| + 3 + 2(|\Gamma_1(t)| + |\Gamma_2(s)| \cdot |\Gamma_2(t)| + |\Gamma_2(t)| + 2))$ existentially quantified variables, followed by $2(|\Gamma_1(t)|+|\Gamma_2(s)|\cdot|\Gamma_2(t)|+|\Gamma_2(t)|+1)$ universal variables. The sentence (18) introduces $|S|^2 + |S|^2$ existentially quantified variables ahead of Φ. The matrix of the formula is of length at most quadratic in the size of the game, and the maximum degree of any polynomial in the formula is 3. We define the size of a game G as: $|G| = |S|+|T|$, where $|T| = \sum_{s,t \in S} \sum_{a,b \in Moves} |\delta(s, a, b)(t)|$. Using the complexity of deciding a formula in the theory of real closed fields [3], we get the following result.

Theorem 7 (Decision complexity for exact distance). *For all concurrent games G, given a rational r, and two states s and t, whether $[s \preceq_1 t] \leq r$ can be decided in time $\mathcal{O}(|G|^{\mathcal{O}(|G|^5)})$.*

Corollary 2 (Approximation for exact distance). *For all concurrent games G, given a rational ϵ, and two states s and t, an interval $[l, u]$ with $u - l \leq \epsilon$ such that $[s \preceq_1 t] \in [l, u]$ can be computed in time $\mathcal{O}(\log(\frac{\theta_2 - \theta_1}{\epsilon}) \cdot |G|^{\mathcal{O}(|G|^5)})$.*

In contrast, the formula to check whether $[s \preceq_1 t] \leq r$, for a rational r, as implied by the definition of $H_{\preceq_1}(d)(s, t)$, that does not use Lemma 3, has five quantifier alternations due to the inner sup, which when combined with the $2 \cdot |S|^2$ existentially quantified variables in the sentence (18), yields a decision complexity of $\mathcal{O}(|G|^{\mathcal{O}(|G|^7)})$.

5.3 Computing the kernels

Similar to the case of turn-based games and MDPs, the kernel of the simulation metric $\preceq_1$ for concurrent games can be computed as the limit of the series $\preceq_1^0, \preceq_1^1, \preceq_1^2, \ldots,$ of relations. For all $s, t \in S$, we have $(s, t) \in \preceq_1^0$ iff $s \equiv t$. For all $n \geq 0$, we have $(s, t) \in \preceq_1^{n+1}$ iff the following sentence Φ_s is true: $\forall a.\Phi(\preceq^n, a) \to a \leq 0$, where Φ is defined as in (17). At any step in the iteration, the distance between any pair of states $u, v \in S$ is defined as follows,

$$\forall u, v \in S.\ d(u, v) = \begin{cases} 0 & \text{if (s, t)} \in \preceq_1^n \\ 1 & \text{if (s, t)} \not\in \preceq_1^n \end{cases} .$$

To compute the bisimulation kernel, we again proceed by partition refinement. For a set of partitions $\mathcal{Q}^0, \mathcal{Q}^1, \ldots,$ $(s, t) \in \simeq^{n+1}$ iff the following sentence Φ_b is true for the state pairs (s, t) and (t, s): $\forall a.\Phi(\mathcal{Q}^n, a) \to a \leq 0$.

Complexity. In the worst case we need $\mathcal{O}(|S|^2)$ partition refinement steps for computing both the simulation and the bisimulation relation. At each partition refinement step the number of state pairs we consider is bounded by $\mathcal{O}(|S|^2)$. We can check if Φ_s and Φ_b are true using a decision procedure for the theory of real closed fields. Therefore, we need $\mathcal{O}(|S|^4)$ decisions to compute the kernels. The partitioning of states based on the decisions can be done by any of the partition refinement algorithms, such as [20].

Theorem 8 *For all concurrent games G, states s and t, whether $s \preceq_1 t$ can be decided in $\mathcal{O}(|G|^{\mathcal{O}(|G|^3)})$ time, and whether $s \simeq_g t$ can be decided in $\mathcal{O}(|G|^{\mathcal{O}(|G|^3)})$ time.*

References

[1] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating time temporal logic. *J. ACM*, 49:672–713, 2002.

[2] C. Baier. Polynomial time algorithms for testing probabilistic bisimulation and simulation. In *CAV*, pages 50–61. Springer-Verlag, 1996.

[3] S. Basu. New results on quantifier elimination over real closed fields and applications to constraint databases. *J. ACM*, 46(4):537–555, 1999.

[4] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995. Volumes I and II.

[5] M.C. Browne, E.M. Clarke, and O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59:115–131, 1988.

[6] J. F. Canny. Some algebraic and geometric computations in pspace. In *STOC*, pages 460–467. ACM Press, 1988.

[7] L. de Alfaro, T.A. Henzinger, and O. Kupferman. Concurrent reachability games. *Theoretical Computer Science*, 386(3):188–217, 2007.

[8] L. de Alfaro and R. Majumdar. Quantitative solution of omega-regular games. *Journal of Computer and System Sciences*, 68:374–397, 2004.

[9] L. de Alfaro, R. Majumdar, V. Raman, and M. Stoelinga. Game relations and metrics. In *LICS*, pages 99–108. IEEE Computer Society Press, 2007.

[10] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labelled Markov systems. In *CONCUR*, pages 258–273. Springer-Verlag, 1999.

[11] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Approximating labelled markov processes. *Information and Computation*, 2002.

[12] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. The metric analogue of weak bisimulation for probabilistic processes. In *LICS*, pages 413–422. ACM Press, 2002.

[13] K. Etessami and M. Yannakakis. Recursive concurrent stochastic games. In *ICALP (2)*, pages 324–335. Springer-Verlag, 2006.

[14] K. Etessami and M. Yannakakis. On the complexity of Nash equilibria and other fixed points (extended abstract). In *FOCS*, pages 113–123. IEEE Computer Society Press, 2007.

[15] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.

[16] M. R. Garey, R. L. Graham, and D. S. Johnson. Some NP-complete geometric problems. In *STOC*, pages 10–22. ACM Press, 1976.

[17] D.A. Martin. The determinacy of Blackwell games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998.

[18] J.F. Mertens and A. Neyman. Stochastic games. *International Journal of Game Theory*, 10:53–66, 1981.

[19] R. Milner. Operational and algebraic semantics of concurrent processes. In *Handbook of Theoretical Computer Science*, volume B, pages 1202–1242. Elsevier Science Publishers, 1990.

[20] R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.

[21] R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. In *CONCUR*, pages 481–496. Springer-Verlag, 1994.

[22] L.S. Shapley. Stochastic games. *Proc. Nat. Acad. Sci. USA*, 39:1095–1100, 1953.

[23] M. Sion. On general minimax theorems. *Pacific Journal of Mathematics.*, 8:171–176, 1958.

[24] F. van Breugel, B. Sharma, and J. Worrell. Approximating a behavioural pseudometric without discount for probabilistic systems. *CoRR*, abs/0803.3796, 2008.

[25] F. van Breugel and J. Worrell. An algorithm for quantitative verification of probabilistic transition systems. In *CONCUR*, pages 336–350. Springer-Verlag, 2001.

[26] F. van Breugel and J. Worrell. Towards quantitative verification of probabilistic transition systems. In *ICALP*, pages 421–432. Springer-Verlag, 2001.

[27] Y. Ye. Improved complexity results on solving real-number linear feasibility problems. *Math. Program.*, 106(2):339–363, 2006.

[28] L. Zhang and H. Hermanns. Deciding simulations on probabilistic automata. In *ATVA*, pages 207–222. Springer-Verlag, 2007.