POWVER

Technical Report 2017-09

Title: **Polynomial-Time Alternating Probabilistic Bisimulation for Interval MDPs**

Author: Vahid Hashemi, Andrea Turrini, Ernst Moritz Hahn, Holger Hermanns, Khaled Elbassioni

Report Number: 2017-09

ERC Project: Power to the People. Verified.

ERC Project ID: 695614

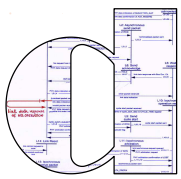
Funded Under: H2020-EU.1.1. – EXCELLENT SCIENCE

Host Institution: Universität des Saarlandes, Dependable Systems and Software

Published In: SETTA 2017

Please cite this publication as follows:

Vahid Hashemi, Andrea Turrini, Ernst Moritz Hahn, Holger Hermanns, Khaled Elbassioni.
Polynomial-Time Alternating Probabilistic Bisimulation for Interval MDPs.
Dependable Software Engineering. Theories, Tools, and Applications - 3rd International Symposium, SETTA 2017, Changsha, China, October 23-25, 2017, Proceedings. Lecture Notes in Computer Science 10606, Springer 2017, ISBN 978-3-319-69482-5. 25-41.



POWER TO THE PEOPLE.
VERIFIED.
erc

Polynomial-Time Alternating Probabilistic Bisimulation for Interval MDPs*

Vahid Hashemi[1], Andrea Turrini[2], Ernst Moritz Hahn[1,2], Holger Hermanns[1], and Khaled Elbassioni[3]

[1] Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
[2] State Key Laboratory of Computer Science, ISCAS, Beijing, China
[3] Masdar Institute of Science and Technology, Abu Dhabi, UAE

Abstract. Interval Markov decision processes (IMDPs) extend classical MDPs by allowing intervals to be used as transition probabilities. They provide a powerful modelling tool for probabilistic systems with an additional variation or uncertainty that relaxes the need of knowing the exact transition probabilities, which are usually difficult to get from real systems. In this paper, we discuss a notion of alternating probabilistic bisimulation to reduce the size of the IMDPs while preserving the probabilistic CTL properties it satisfies from both computational complexity and compositional reasoning perspectives. Our alternating probabilistic bisimulation stands on the competitive way of resolving the IMDP nondeterminism which in turn finds applications in the settings of the controller (parameter) synthesis for uncertain (parallel) probabilistic systems. By using the theory of linear programming, we improve the complexity of computing the bisimulation from the previously known EXPTIME to PTIME. Moreover, we show that the bisimulation for IMDPs is a congruence with respect to two facets of parallelism, namely synchronous product and interleaving. We finally demonstrate the practical effectiveness of our proposed approaches by applying them on several case studies using a prototypical tool.

1 Introduction

Markov Decision Processes (*MDP*s) are a widely and commonly used mathematical abstraction that permits to study properties of real world systems in a rigorous way. The actual system is represented by means of a model subsuming the states the system can be in and the transitions representing how the system evolves from one state to another; the actual properties are encoded as logical formulas that are then verified against the model.

* This work is supported by the ERC Advanced Investigators Grant 695614 (POWVER), by the CAS/SAFEA International Partnership Program for Creative Research Teams, by the National Natural Science Foundation of China (Grants No. 61550110506 and 61650410658), by the Chinese Academy of Sciences Fellowship for International Young Scientists, and by the CDZ project CAP (GZ 1023).

*MDP*s are suitable for modelling two core aspects of the behavior of the real world systems: *nondeterminism* and *probability*. A nondeterministic behavior can be introduced to model a behavior of the system that is just partially known (like receiving an asynchronous message, of which it is known it can be received in the current state but no information is available so to quantify its likelihood) or to leave implementation details open. A probabilistic behavior occurs whenever the successor state of the system is not uniquely determined by the current system and the performed action, but depends on a random choice; such a choice can be due to the design of the system, as it is required by the implementation of a distributed consensus protocol with faulty processes [3, 14], or by physical properties that need to be taken into account, like transmission errors.

Finding the exact probability values for the transitions is sometimes a difficult task: while probabilities introduced by design can be well known, probabilities modelling physical properties are usually estimated by observing the actual system. This means that the resulting *MDP* is a more or less appropriate abstraction of the real system, depending on how close the estimated probability values are to the actual values; as a consequence, the actual properties of the real system are more or less reflected by the satisfaction of the formulas by the model.

Interval Markov Decision Processes (*IMDP*s) extend the classical *MDP*s by including uncertainty over the transition probabilities. Instead of a single value for the probability of reaching a specific successor by taking a transition, *IMDP*s allow ranges of possible probability values given as closed intervals of the reals. Thereby, *IMDP*s provide a powerful modelling tool for probabilistic systems with an additional variation or uncertainty concerning the knowledge of exact transition probabilities. They are especially useful to represent realistic stochastic systems that, for instance, evolve in unknown environments with bounded behavior or do not preserve the Markov property.

Since their introduction (under the name of bounded-parameter *MDP*s) [16], *IMDP*s have been receiving a lot of attention in the formal verification community. They are particularly viewed as the appropriate abstraction model for uncertain systems with large state spaces, including continuous dynamical systems, for the purpose of analysis, verification, and control synthesis. Several model checking and control synthesis techniques have been developed [37, 38, 43] causing a boost in the applications of *IMDP*s, ranging from verification of continuous stochastic systems (e.g., [30]) to robust strategy synthesis for robotic systems (e.g., [32–34, 43]).

Bisimulation minimisation is a well-known technique that has been successfully used to reduce the size of a system while preserving the properties it satisfies [5, 8, 9, 23, 27]; this helps the task of the property solver, since it has to work on a smaller system. Compositional minimisation permits to minimise the single components of the system before combining them, thus making the task of the minimiser easier and extending its applicability to larger systems. In this paper, we show that this approach is suitable also for *IMDP*s. The contributions of the paper are as follows.

– We define alternating probabilistic bisimulations to compress the *IMDP* model size with respect to the controller synthesis semantics while preserving probabilistic CTL property satisfaction. We show that the compressed models can be computed in polynomial time.
– From the perspective of compositional reasoning, we show that alternating probabilistic bisimulations for *IMDP*s are congruences with respect to two facets of parallelism, namely synchronous product and interleaving.
– We show promising results on a variety of case studies, obtained by prototypical implementations of all algorithms.

Related work. Related work can be grouped into three categories: uncertain Markov model formalisms, bisimulation minimization, and compositional minimization.

Firstly, from the modelling viewpoint, various probabilistic modelling formalisms with uncertain transitions are studied in the literature. Interval Markov Chains (*IMC*s) [25,28] or abstract Markov chains [13] extend standard discrete-time Markov Chains (*MC*s) with interval uncertainties. They do not feature the non-deterministic choices of transitions. Uncertain *MDP*s [38] allow more general sets of distributions to be associated with each transition, not only those described by intervals. They usually are restricted to *rectangular uncertainty sets* requiring that the uncertainty is linear and independent for any two transitions of any two states. Parametric *MDP*s [17], to the contrary, allow such dependencies as every probability is described as a rational function of a finite set of global parameters. *IMDP*s extend *IMC*s by inclusion of nondeterminism and are a subset of uncertain *MDP*s and parametric *MDP*s.

Secondly, as regards to the bisimulation minimization for uncertain or parametric probabilistic models, works in [18, 20, 21] explored the computational complexity and approximability of deciding probabilistic bisimulation for *IMDP*s with respect to the cooperative resolution of nondeterminism. In this work, we show that *IMDP*s can be minimized efficiently with respect to the competitive resolution of nondeterminism.

Lastly, from the viewpoint of compositional minimization, *IMC*s [25] and abstract Probabilistic Automata (*PA*) [10, 11] serve as specification theories for *MC* and *PA*, featuring satisfaction relation and various refinement relations. In [22], the authors discuss the key ingredients to build up the operations of parallel composition for composing *IMDP* components at run-time. Our paper follows this spirit for alternating probabilistic bisimulation on *IMDP*s.

Structure of the paper. We start with necessary preliminaries in Section 2. In Section 3, we give the definitions of alternating probabilistic bisimulation for interval *MDP* and discuss their properties. A polynomial time decision algorithm to decide alternating probabilistic bisimulation for *IMDP*s and also compositional reasoning are discussed in Sections 4 and 5, respectively. In Section 6, we demonstrate our approach on some case studies and present promising experimental results. Finally, in Section 7 we conclude the paper.

2 Mathematical Preliminaries

For a set X, denote by $\mathrm{Disc}(X)$ the set of discrete probability distributions over X. Intuitively, a discrete probability distribution ρ is a function $\rho\colon X \to \mathbb{R}_{\geq 0}$ such that $\sum_{x\in X} \rho(x) = 1$; for $X' \subseteq X$, we write $\rho(X')$ for $\sum_{x\in X'} \rho(x)$. Given $\rho \in \mathrm{Disc}(X)$, we denote by $\mathrm{Supp}(\rho)$ the set $\{\, x \in X \mid \rho(x) > 0 \,\}$ and by δ_x, where $x \in X$, the *Dirac* distribution such that $\delta_x(y) = 1$ for $y = x$, 0 otherwise. For a probability distribution ρ, we also write $\rho = \{\, (x, p_x) \mid x \in X \,\}$ where p_x is the probability of x.

The lifting $\mathcal{L}(\mathcal{R})$ [31] of a relation $\mathcal{R} \subseteq X \times Y$ is defined as follows: for $\rho_X \in \mathrm{Disc}(X)$ and $\rho_Y \in \mathrm{Disc}(Y)$, $\rho_X\ \mathcal{L}(\mathcal{R})\ \rho_Y$ holds if there exists a *weighting function* $w\colon X \times Y \to [0,1]$ such that (1) $w(x,y) > 0$ implies $x\ \mathcal{R}\ y$, (2) $\sum_{y\in Y} w(x,y) = \rho_X(x)$, and (3) $\sum_{x\in X} w(x,y) = \rho_Y(y)$. When $\mathcal{R}$ is an equivalence relation on X, $\rho_1\ \mathcal{L}(\mathcal{R})\ \rho_2$ holds if for each $\mathcal{C} \in X/\mathcal{R}$, $\rho_1(\mathcal{C}) = \rho_2(\mathcal{C})$ where $X/\mathcal{R} = \{\, [x]_\mathcal{R} \mid x \in X \,\}$ and $[x]_\mathcal{R} = \{\, y \in X \mid y\ \mathcal{R}\ x \,\}$.

For a vector $\boldsymbol{x} \in \mathbb{R}^n$ we denote by $\boldsymbol{x}_i$, its i-th component, and we call $\boldsymbol{x}$ a *weight vector* if $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$ and $\sum_{i=1}^n \boldsymbol{x}_i = 1$. Given two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, their Euclidean inner product $\boldsymbol{x} \cdot \boldsymbol{y}$ is defined as $\boldsymbol{x} \cdot \boldsymbol{y} = \boldsymbol{x}^T\boldsymbol{y} = \sum_{i=1}^n \boldsymbol{x}_i \cdot \boldsymbol{y}_i$. We write $\boldsymbol{x} \leq \boldsymbol{y}$ if $\boldsymbol{x}_i \leq \boldsymbol{y}_i$ for each $1 \leq i \leq n$ and we denote by $\mathbf{1} \in \mathbb{R}^n$ the vector such that $\mathbf{1}_i = 1$ for each $1 \leq i \leq n$. For a set of vectors $S = \{\boldsymbol{s}^1, \ldots, \boldsymbol{s}^m\} \subseteq \mathbb{R}^n$, we say that $\boldsymbol{s}$ is a *convex combination* of elements of S, if $\boldsymbol{s} = \sum_{i=1}^m \boldsymbol{w}_i \cdot \boldsymbol{s}^i$ for some weight vector $\boldsymbol{w} \in \mathbb{R}^m_{\geq 0}$. For a given set $P \subseteq \mathbb{R}^n$, we denote by $\mathrm{conv}\, P$ the convex hull of P and by $\mathtt{Ext}(P)$ the set of extreme points of P. If P is a polytope in $\mathbb{R}^n$ then for each $1 \leq i \leq n$, the projection $\mathrm{proj}_{\mathbf{e^i}} P$ on the i-th dimension of P is defined as $\mathrm{proj}_{\mathbf{e^i}} P = [\min_i P, \max_i P]$ where $\mathbf{e^i} \in \mathbb{R}^n$ is such that $\mathbf{e}^\mathbf{i}_i = 1$ and $\mathbf{e}^\mathbf{i}_j = 0$ for each $j \neq i$, $\min_i P = \min\{\, \boldsymbol{x}_i \mid \boldsymbol{x} \in P \,\}$, and $\max_i P = \max\{\, \boldsymbol{x}_i \mid \boldsymbol{x} \in P \,\}$.

2.1 Interval Markov Decision Processes

We now define *Interval Markov Decision Processes* (*IMDP*s) as an extension of *MDP*s, which allows for the inclusion of transition probability uncertainties as *intervals*. *IMDP*s belong to the family of uncertain *MDP*s and allow to describe a set of *MDP*s with identical (graph) structures that differ in distributions associated with transitions.

Definition 1 (*IMDP*s). *An* Interval Markov Decision Process *(IMDP)* $\mathcal{M}$ *is a tuple* $(S, \bar{s}, \mathcal{A}, \mathtt{AP}, L, I)$*, where* S *is a finite set of* states*,* $\bar{s} \in S$ *is the initial state,* $\mathcal{A}$ *is a finite set of* actions*,* $\mathtt{AP}$ *is a finite set of* atomic propositions*,* $L\colon S \to 2^{\mathtt{AP}}$ *is a* labelling function*, and* $I\colon S \times \mathcal{A} \times S \to \mathbb{I} \cup \{[0,0]\}$ *is a total* interval transition probability function *with* $\mathbb{I} = \{\, [l,u] \subseteq \mathbb{R} \mid 0 < l \leq u \leq 1 \,\}$.

Given $s \in S$ and $a \in \mathcal{A}$, we call $\mathfrak{h}^a_s \in \mathrm{Disc}(S)$ a *feasible distribution* reachable from s by a, denoted by $s \xrightarrow{a} \mathfrak{h}^a_s$, if, for each state $s' \in S$, we have $\mathfrak{h}^a_s(s') \in I(s,a,s')$. We denote the set of feasible distributions for state s and action a by $\mathcal{H}^a_s$, i.e., $\mathcal{H}^a_s = \{\, \mathfrak{h}^a_s \in \mathrm{Disc}(S) \mid s \xrightarrow{a} \mathfrak{h}^a_s \,\}$ and we denote the set of available

actions at state $s \in S$ by $\mathcal{A}(s)$, i.e., $\mathcal{A}(s) = \{\, a \in \mathcal{A} \mid \mathcal{H}^a_s \neq \emptyset \,\}$. We assume that $\mathcal{A}(s) \neq \emptyset$ for all $s \in S$.

We define the *size* of $\mathcal{M}$, written $|\mathcal{M}|$, as the number of non-zero entries of I, i.e., $|\mathcal{M}| = |\{\, (s,a,s',\iota) \in S \times \mathcal{A} \times S \times \mathbb{I} \mid I(s,a,s') = \iota \,\}| \in \mathcal{O}(|S|^2 \cdot |\mathcal{A}|)$.

A *path* ξ in $\mathcal{M}$ is a finite or infinite sequence of states $\xi = s_0 s_1 \ldots$ such that for each $i \geq 0$ there exists $a_i \in \mathcal{A}(s_i)$ such that $I(s_i, a_i, s_{i+1}) \in \mathbb{I}$. The i-th state along the path ξ is denoted by $\xi[i]$ and, if the path is finite, we denote by $\mathit{last}(\xi)$ its last state. The sets of all finite and infinite paths in $\mathcal{M}$ are denoted by Paths^* and Paths, respectively.

The nondeterministic choices between available actions and feasible distributions present in an *IMDP* are resolved by strategies and natures, respectively.

Definition 2 (Strategy and Nature in *IMDP*s). *Given an IMDP $\mathcal{M}$, a* strategy *is a function $\sigma\colon \mathit{Paths}^* \to \mathrm{Disc}(\mathcal{A})$ such that for each path $\xi \in \mathit{Paths}^*$, $\sigma(\xi) \in \mathrm{Disc}(\mathcal{A}(\mathit{last}(\xi))$. A* nature *is a function $\pi\colon \mathit{Paths}^* \times \mathcal{A} \to \mathrm{Disc}(S)$ such that for each path $\xi \in \mathit{Paths}^*$ and action $a \in \mathcal{A}(s)$, $\pi(\xi, a) \in \mathcal{H}^a_s$ where $s = \mathit{last}(\xi)$.*

The sets of all strategies and all natures are denoted by Σ and Π, respectively.

Given a finite path ξ of an *IMDP*, a strategy σ, and a nature π, the system evolution proceeds as follows: let $s = \mathit{last}(\xi)$. First, an action $a \in \mathcal{A}(s)$ is chosen probabilistically by σ. Then, π resolves the uncertainties and chooses one feasible distribution $\mathfrak{h}^a_s \in \mathcal{H}^a_s$. Finally, the next state s' is chosen according to the distribution $\mathfrak{h}^a_s$, and the path ξ is extended by s'.

A strategy σ and a nature π induce a probability measure over paths as follows. The basic measurable events are the cylinder sets of finite paths, where the *cylinder set* of a finite path ξ is the set $\mathit{Cyl}_\xi = \{\, \xi' \in \mathit{Paths} \mid \xi \text{ is a prefix of } \xi' \,\}$. The probability $\Pr^{\sigma,\pi}_{\mathcal{M}}$ of a state s' is defined to be $\Pr^{\sigma,\pi}_{\mathcal{M}}[\mathit{Cyl}_{s'}] = \delta_{\bar{s}}(s')$ and the probability $\Pr^{\sigma,\pi}_{\mathcal{M}}[\mathit{Cyl}_{\xi s'}]$ of traversing a finite path $\xi s'$ is defined to be

$$\Pr^{\sigma,\pi}_{\mathcal{M}}[\mathit{Cyl}_{\xi s'}] = \Pr^{\sigma,\pi}_{\mathcal{M}}[\mathit{Cyl}_{\xi}] \cdot \sum_{a \in \mathcal{A}(\mathit{last}(\xi))} \sigma(\xi)(a) \cdot \pi(\xi, a)(s').$$

Standard measure theoretical arguments ensure that $\Pr^{\sigma,\pi}_{\mathcal{M}}$ extends uniquely to the σ-field generated by cylinder sets.

As an example of *IMDP*s, consider the one depicted in Figure 1. The set of states is $S = \{s, t, u\}$ with s being the initial one; the set of actions is $\mathcal{A} = \{a, b\}$ while the set of atomic propositions assigned to each state by the labelling function L is represented by the letters in curly brackets near each state. Finally, the transition probability intervals are $I(s,a,t) = [\frac{1}{3}, \frac{2}{3}]$, $I(s,a,u) = [\frac{1}{10}, 1]$, $I(s,b,t) = [\frac{2}{5}, \frac{3}{5}]$, $I(s,b,u) = [\frac{1}{4}, \frac{2}{3}]$, $I(t,a,t) = I(u,b,u) = [1,1]$, and $I(t,b,t) = I(u,a,u) = [0,0]$.

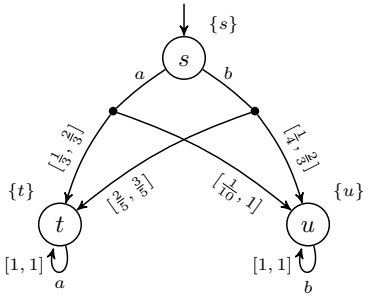


Fig. 1. An example of *IMDP*.

2.2 Probabilistic Computation Tree Logic (PCTL)

There are various ways how to describe properties of *IMDP*s. Here we focus on *probabilistic CTL* (PCTL) [19]. The syntax of PCTL state formulas φ and PCTL path formulas ψ is given by:

$$\varphi := a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathsf{P}_{\bowtie p}(\psi)$$
$$\psi := \mathsf{X}\varphi \mid \varphi_1 \,\mathsf{U}\, \varphi_2 \mid \varphi_1 \,\mathsf{U}^{\leq k}\, \varphi_2$$

where $a \in \mathtt{AP}$, $p \in [0,1]$ is a rational constant, $\bowtie \in \{\leq, <, \geq, >\}$, and $k \in \mathbb{N}$.

The semantics of a PCTL formula with respect to *IMDP*s is very similar to the classical PCTL semantics for *MDP*s: they coincide on all formulas except for $\mathsf{P}_{\bowtie p}(\psi)$, where they may differ depending on how the nondeterminism is resolved. Formally, for the formulas they agree on, given a state s and a state formula φ, the satisfaction relation $s \models \varphi$ is defined as follows:

$s \models a$	if $a \in L(s)$;
$s \models \neg\varphi$	if it is not the case that $s \models \varphi$, also written $s \not\models \varphi$;
$s \models \varphi_1 \wedge \varphi_2$	if $s \models \varphi_1$ and $s \models \varphi_2$.

Given an infinite path $\xi = s_1 s_2 \ldots$ and a path formula ψ, the satisfaction relation $\xi \models \psi$ is defined as follows:

$\xi \models \mathsf{X}\varphi$	if $s_2 \models \varphi$;
$\xi \models \varphi_1 \,\mathsf{U}^{\leq k}\, \varphi_2$	if there exists $i \leq k$ such that $s_i \models \varphi_2$ and $s_j \models \varphi_1$ for every $1 \leq j < i$;
$\xi \models \varphi_1 \,\mathsf{U}\, \varphi_2$	if there exists $k \in \mathbb{N}$ such that $\xi \models \varphi_1 \,\mathsf{U}^{\leq k}\, \varphi_2$.

Regarding the state formula $\mathsf{P}_{\bowtie p}(\psi)$, its semantics depends on the way the nondeterminism is resolved for the probabilistic operator $\mathsf{P}_{\bowtie p}(\psi)$. When quantifying both types of nondeterminism universally, the corresponding satisfaction relation $s \models \mathsf{P}_{\bowtie p}(\psi)$ is defined as follows:

$$s \models \mathsf{P}_{\bowtie p}(\psi) \text{ if } \forall \sigma \in \Sigma : \forall \pi \in \Pi : \Pr_s^{\sigma,\pi}[Paths_\psi] \bowtie p \qquad (\forall)$$

where $Paths_\psi = \{\, \xi \in Paths \mid \xi \models \psi \,\}$ denotes the set of infinite paths satisfying ψ. It is easy to show that the set $Paths_\psi$ is measurable for any path formula ψ, hence its probability can be computed and compared with p. When the *IMDP* is actually an *MDP*, i.e., all intervals are single values, then the satisfaction relation $s \models \mathsf{P}_{\bowtie p}(\psi)$ in Equation $(\forall)$ coincides with the corresponding definition for *MDP*s (cf. [2, Sect. 10.6.2]). We explain later how the semantics differs for a different resolution of nondeterminism for strategy and nature.

3 Alternating Probabilistic Bisimulation for *IMDP*s

This section revisits required main results on probabilistic bisimulation for *IMDP*s, as developed in [20]. In the setting of this paper, we consider alternating probabilistic bisimulation which stems from the competitive resolution

of nondeterminisms in *IMDP*s. In the competitive semantics, the strategy and nature are playing in a game *against* each other; therefore, they are resolved *competitively*. This semantics is very natural in the context of controller synthesis for systems with uncertain probabilities or in the context of parameter synthesis for parallel systems.

In this paper, in order to resolve the stochastic nondeterminism we focus on the dynamic approach [24, 42], i.e., independently at each computation step as it is easier to work with algorithmically and can be seen as a relaxation of the static approach that is often intractable [4, 7, 12, 16].

To this end, we consider the *controller synthesis* semantics to resolve the two sources of *IMDP* nondeterminisms and discuss the resultant alternating probabilistic bisimulation. Note that there is another variant of alternating probabilistic bisimulation based on the *parameter synthesis* semantics [20]. However, the alternating bisimulations relations resulting from these two semantics coincide [20, Theorem 4].

In the controller synthesis semantics, we search for a strategy σ such that for any nature π, a fixed property φ is satisfied. This corresponds to the satisfaction relation $\models_{(\exists\sigma\forall)}$ in PCTL, obtained from $\models$ by replacing the rule $(\forall)$ with

$$s \models_{(\exists\sigma\forall)} \mathsf{P}_{\bowtie p}(\psi) \text{ if } \exists\sigma \in \Sigma : \forall\pi \in \Pi : \Pr_s^{\sigma,\pi}[Paths_\psi] \bowtie p. \qquad (\exists\sigma\forall)$$

As regards to bisimulation, the competitive setting is not common. We define a bisimulation similar to the alternating bisimulation of [1] applied to non-stochastic two-player games. For a decision $\rho \in \mathrm{Disc}(\mathcal{A})$ of σ, let $s \xrightarrow{\rho} \mu$ denote that μ is a possible successor distribution, i.e., there are decisions μ_a of π for each $a \in \mathrm{Supp}(\rho)$ such that $\mu = \sum_{a\in\mathcal{A}} \rho(a) \cdot \mu_a$.

Definition 3. *Given an IMDP $\mathcal{M}$, let $\mathcal{R} \subseteq S \times S$ be an equivalence relation. We say that $\mathcal{R}$ is an* alternating probabilistic $(\exists\sigma\forall)$-bisimulation *if for any $(s,t) \in \mathcal{R}$ we have that $L(s) = L(t)$ and for each $\rho_s \in \mathrm{Disc}(\mathcal{A}(s))$ there exists $\rho_t \in \mathrm{Disc}(\mathcal{A}(t))$ such that for each $t \xrightarrow{\rho_t} \mu_t$ there exists $s \xrightarrow{\rho_s} \mu_s$ such that $\mu_s \, \mathcal{L}(\mathcal{R}) \, \mu_t$. We write $s \sim_{(\exists\sigma\forall)} t$ whenever $(s,t) \in \mathcal{R}$ for some alternating probabilistic $(\exists\sigma\forall)$-bisimulation $\mathcal{R}$.*

The exact alternation of quantifiers might be counter-intuitive at first sight. Note that it exactly corresponds to the situation in non-stochastic games [1]. The defined bisimulation preserves the PCTL logic with respect to the $\models_{(\exists\sigma\forall)}$ semantics.

Theorem 4. *For states $s \sim_{(\exists\sigma\forall)} t$ and any PCTL formula φ, we have $s \models_{(\exists\sigma\forall)} \varphi$ if and only if $t \models_{(\exists\sigma\forall)} \varphi$.*

As a concluding remark, it is worthwhile to note that Definition 3 can be seen as the conservative extension of probabilistic bisimulation for (state-labelled) MDPs. To see that, assume the set of uncertainty for every transition is a singleton. Since there is only one choice for the nature, the role of nature can be safely removed from the definitions.

4 A PTIME Decision Algorithm for Bisimulation Minimization

Computation of the alternating probabilistic bisimulation $\sim_{(\exists\sigma\forall)}$ for *IMDP*s follows the standard partition refinement approach [6,15,26,35]. However, the core part is finding out whether two states "violate the definition of bisimulation". This verification routine amounts to check that s and t have the same set of *strictly minimal polytopes* detailed as follows.

For $s \in S$ and $a \in \mathcal{A}(s)$, recall that $\mathcal{H}_s^a$ denotes the polytope of feasible successor distributions over *states* with respect to taking the action a in the state s. By $\mathcal{P}_{\mathcal{R}}^{s,a}$, we denote the polytope of feasible successor distributions over *equivalence classes* of $\mathcal{R}$ with respect to taking the action a in the state s. Given an interval $[l,u]$, let $\inf[l,u] = l$ and $\sup[l,u] = u$. For $\mu \in \mathrm{Disc}(S/\mathcal{R})$ we set $\mu \in \mathcal{P}_{\mathcal{R}}^{s,a}$ if, for each $\mathcal{C} \in S/\mathcal{R}$, we have $\mu(\mathcal{C}) \in I(s,a,\mathcal{C})$ where

$$I(s,a,\mathcal{C}) = \Big[\min\big\{1, \sum_{s'\in\mathcal{C}} \inf I(s,a,s')\big\}, \min\big\{1, \sum_{s'\in\mathcal{C}} \sup I(s,a,s')\big\}\Big].$$

It is not difficult to see that each $\mathcal{P}_{\mathcal{R}}^{s,a}$ can be represented as an $\mathcal{H}$-polytope. To simplify our presentation, we shall fix an order over the equivalence classes in $S/\mathcal{R}$. By doing so, any distribution $\rho \in \mathrm{Disc}(S/\mathcal{R})$ can be seen as a vector $\boldsymbol{v} \in \mathbb{R}^n_{\geq 0}$ such that $\boldsymbol{v}_i = \rho(\mathcal{C}_i)$ for each $1 \leq i \leq n$, where $n = |S/\mathcal{R}|$ and $\mathcal{C}_i$ is the i-th equivalence class in the order. For the above discussion, $\rho \in \mathcal{P}_{\mathcal{R}}^{s,a}$ if and only if $\rho(\mathcal{C}_i) \in [\boldsymbol{l}_i^{s,a}, \boldsymbol{u}_i^{s,a}]$ for any $1 \leq i \leq n$ and $\rho \in \mathrm{Disc}(S/\mathcal{R})$, where $\boldsymbol{l}^{s,a}$ and $\boldsymbol{u}^{s,a}$ are vectors such that $\boldsymbol{l}_i^{s,a} = \min\{1, \sum_{s'\in\mathcal{C}_i} \inf I(s,a,s')\}$ and $\boldsymbol{u}_i^{s,a} = \min\{1, \sum_{s'\in\mathcal{C}_i} \sup I(s,a,s')\}$ for each $1 \leq i \leq n$. Therefore, $\mathcal{P}_{\mathcal{R}}^{s,a}$ corresponds to an $\mathcal{H}$-polytope defined by $\{\, \boldsymbol{x}^{s,a} \in \mathbb{R}^n \mid \boldsymbol{l}^{s,a} \leq \boldsymbol{x}^{s,a} \leq \boldsymbol{u}^{s,a}, \mathbf{1}\cdot\boldsymbol{x}^{s,a} = 1 \,\}$.

Definition 5 (Strictly minimal polytopes). *Given an IMDP $\mathcal{M}$, a state s, an equivalence relation $\mathcal{R} \subseteq S \times S$, and a set $\{\, \mathcal{P}_{\mathcal{R}}^{s,a} \mid a \in \mathcal{A}(s) \,\}$ where for each $a \in \mathcal{A}(s)$, for given $\boldsymbol{l}^{s,a}, \boldsymbol{u}^{s,a} \in \mathbb{R}^n$, $\mathcal{P}_{\mathcal{R}}^{s,a}$ is the convex polytope $\mathcal{P}_{\mathcal{R}}^{s,a} = \{\, \boldsymbol{x}^{s,a} \in \mathbb{R}^n \mid \boldsymbol{l}^{s,a} \leq \boldsymbol{x}^{s,a} \leq \boldsymbol{u}^{s,a}, \mathbf{1}\cdot\boldsymbol{x}^{s,a} = 1 \,\}$, a polytope $\mathcal{P}_{\mathcal{R}}^{s,a}$ is called* strictly minimal*, if for no $\rho \in \mathrm{Disc}(\mathcal{A}(s)\setminus\{a\})$, we have $\mathcal{P}_{\mathcal{R}}^{s,\rho} \subseteq \mathcal{P}_{\mathcal{R}}^{s,a}$ where $\mathcal{P}_{\mathcal{R}}^{s,\rho}$ is defined as $\mathcal{P}_{\mathcal{R}}^{s,\rho} = \{\, \boldsymbol{x}^{s,\rho} \in \mathbb{R}^n \mid \boldsymbol{x}^{s,\rho} = \sum_{b\in\mathcal{A}(s)\setminus\{a\}} \rho(b)\cdot\boldsymbol{x}^{s,b} \wedge \boldsymbol{x}^{s,b} \in \mathcal{P}_{\mathcal{R}}^{s,b} \,\}$.*

Checking violation of a given pair of states amounts to check if the states have the same set of strictly minimal polytopes. Formally,

Lemma 6 (cf. [20]). *Given an IMDP $\mathcal{M}$ and $s,t \in S$, we have $s \sim_{(\exists\sigma\forall)} t$ if and only if $L(s) = L(t)$ and $\{\, \mathcal{P}_{\sim_{(\exists\sigma\forall)}}^{s,a} \mid a \in \mathcal{A}$ and $\mathcal{P}_{\sim_{(\exists\sigma\forall)}}^{s,a}$ is strictly minimal$\,\} = \{\, \mathcal{P}_{\sim_{(\exists\sigma\forall)}}^{t,a} \mid a \in \mathcal{A}$ and $\mathcal{P}_{\sim_{(\exists\sigma\forall)}}^{t,a}$ is strictly minimal$\,\}$.*

The expensive procedure in the analysis of the worst case time complexity of computing the coarsest alternating probabilistic bisimulation $\sim_{(\exists\sigma\forall)}$, as described in [20], is to check the strict minimality of a polytope $\mathcal{P}_{\mathcal{R}}^{s,a}$ for $a \in \mathcal{A}(s)$. This decision problem has been shown to be exponentially verifiable via a reduction

to a system of linear (in)equalities in EXPTIME. In this paper, we give a polynomial time routine to verify the strict minimality of a polytope which in turn enables a polynomial time decision algorithm to decide $\sim_{(\exists\sigma\forall)}$. To this aim, we use the following equivalent form of the Farkas' Lemma [39].

Lemma 7. *Let $A \in \mathbb{R}^{m\times n}$, $\boldsymbol{b} \in \mathbb{R}^m$ and $\boldsymbol{c} \in \mathbb{R}^n$. Then, $A\boldsymbol{x} \leq \boldsymbol{b}$ implies $\boldsymbol{c}\cdot\boldsymbol{x} \leq d$ if and only if there exists $\boldsymbol{y} \in \mathbb{R}^m_{\geq 0}$ such that $A^T\boldsymbol{y} = \boldsymbol{c}$ and $\boldsymbol{b}\cdot\boldsymbol{y} \leq d$.*

This variant of Farkas' Lemma leads us to establish the main result of the paper. Formally,

Theorem 8. *Given an IMDP $\mathcal{M}$, a state $s \in S$, an equivalence relation $\mathcal{R} \subseteq S \times S$ and a set $\{\, \mathcal{P}^{s,a}_{\mathcal{R}} \mid a \in \mathcal{A}(s) \,\}$ defined as in Definition 5, checking whether for each $a \in \mathcal{A}(s)$, the polytope $\mathcal{P}^{s,a}_{\mathcal{R}}$ is strictly minimal, is in $\boldsymbol{P}$.*

Proof. Let $\mathcal{A}(s) = \{a_0, a_1, \ldots, a_m\}$, $n = |S/\mathcal{R}|$, and $P_i = \mathcal{P}^{s,a_i}_{\mathcal{R}}$ for $0 \leq i \leq m$. We describe the verification routine to check the strict minimality of P_0; the same routine applies to the other polytopes. We consider the converse of the strict minimality problem which asks to decide whether there exist $\lambda_1, \lambda_2, \ldots, \lambda_m \in \mathbb{R}_{\geq 0}$ such that $\sum_{i=1}^m \lambda_i = 1$ and $\sum_{i=1}^m \lambda_i P_i \subseteq P_0$. We show that the latter problem can be casted as an LP via Farkas' Lemma 7. To this aim, we alternatively reformulate the converse problem as "*do there exist $\lambda_1, \lambda_2, \ldots, \lambda_m \in \mathbb{R}_{\geq 0}$ with $\sum_{i=1}^m \lambda_i = 1$, such that $\boldsymbol{x}^i \in P_i$ for each $1 \leq i \leq m$ implies $\sum_{i=1}^m \lambda_i \boldsymbol{x}^i \in P_0$?*".

For every fixed $\lambda_1, \lambda_2, \ldots, \lambda_m \in \mathbb{R}_{\geq 0}$ with $\sum_{i=1}^m \lambda_i = 1$, the implication "$(\forall 1 \leq i \leq m : \boldsymbol{x}^i \in P_i) \implies \sum_{i=1}^m \lambda_i \boldsymbol{x}^i \in P_0$" can be written as the conjunction of $2n$ conditions:

$$\bigwedge_{i=1}^m \boldsymbol{l}^i \leq \boldsymbol{x}^i \leq \boldsymbol{u}^i \wedge \bigwedge_{i=1}^m \mathbf{1}\cdot\boldsymbol{x}^i = 1 \implies \sum_{i=1}^m \lambda_i \boldsymbol{x}^i_k \geq \boldsymbol{l}^0_k \tag{1}$$

$$\bigwedge_{i=1}^m \boldsymbol{l}^i \leq \boldsymbol{x}^i \leq \boldsymbol{u}^i \wedge \bigwedge_{i=1}^m \mathbf{1}\cdot\boldsymbol{x}^i = 1 \implies \sum_{i=1}^m \lambda_i \boldsymbol{x}^i_k \leq \boldsymbol{u}^0_k \tag{2}$$

for all $1 \leq k \leq n$. (Note that the condition $\mathbf{1}\cdot\sum_{i=1}^m \lambda_i \boldsymbol{x}^i = 1$ is trivially satisfied if $\mathbf{1}\cdot\boldsymbol{x}^i = 1$ for all $1 \leq i \leq m$.) Each of the conditions (1) and (2), by Farkas' Lemma, is equivalent to the feasibility of a system of inequalities; for instance, for a given k, (1) is true if and only if there exist vectors $\boldsymbol{\mu}^{k,i}, \boldsymbol{\nu}^{k,i} \in \mathbb{R}^n_{\geq 0}$ and scalars $\theta^{k,i}, \eta^{k,i} \in \mathbb{R}_{\geq 0}$ for each $1 \leq i \leq m$ satisfying:

$$\boldsymbol{\mu}^{k,i} - \boldsymbol{\nu}^{k,i} + \theta^{k,i}\mathbf{1} - \eta^{k,i}\mathbf{1} = -\lambda_i \mathbf{e}^{\mathbf{k}} \qquad \forall 1 \leq i \leq m \tag{3}$$

$$\sum_{i=1}^m \left(\boldsymbol{u}^i\cdot\boldsymbol{\mu}^{k,i} - \boldsymbol{l}^i\cdot\boldsymbol{\nu}^{k,i} + \theta^{k,i} - \eta^{k,i}\right) \leq -\boldsymbol{l}^0_k \tag{4}$$

Similarly, for a given k, (2) is true if and only if there exist vectors $\widehat{\boldsymbol{\mu}}^{k,i}, \widehat{\boldsymbol{\nu}}^{k,i} \in \mathbb{R}^n_{\geq 0}$ and scalars $\widehat{\theta}^{k,i}, \widehat{\eta}^{k,i} \in \mathbb{R}_{\geq 0}$ for each $1 \leq i \leq m$ satisfying:

$$\widehat{\boldsymbol{\mu}}^{k,i} - \widehat{\boldsymbol{\nu}}^{k,i} + \widehat{\theta}^{k,i}\mathbf{1} - \widehat{\eta}^{k,i}\mathbf{1} = \lambda_i \mathbf{e}^{\mathbf{k}} \qquad \forall 1 \leq i \leq m \tag{5}$$

$$\sum_{i=1}^m (\boldsymbol{u}^i\cdot\widehat{\boldsymbol{\mu}}^{k,i} - \boldsymbol{l}^i\cdot\widehat{\boldsymbol{\nu}}^{k,i} + \widehat{\theta}^{k,i} - \widehat{\eta}^{k,i}) \leq \boldsymbol{u}^0_k \tag{6}$$

Algorithm 1: BISIMULATION($\mathcal{M}$)

Input: A relation $\mathcal{R}$ on $S \times S$
Output: A probabilistic bisimulation $\mathcal{R}$

begin
  $\mathcal{R} \leftarrow \{ (s,t) \in S \times S \mid L(s) = L(t) \}$;
  repeat
    $\mathcal{R}' \leftarrow \mathcal{R}$;
    forall $s \in S$ **do**
      $D \leftarrow \emptyset$;
      forall $t \in [s]_{\mathcal{R}}$ **do**
        if VIOLATE$(s,t,\mathcal{R})$ **then**
          $D \leftarrow D \cup \{t\}$;
      split $[s]_{\mathcal{R}}$ in $\mathcal{R}$ into D and $[s]_{\mathcal{R}} \setminus D$;
  until $\mathcal{R} = \mathcal{R}'$;
  return $\mathcal{R}$;

Procedure 2: VIOLATE$(s,t,\mathcal{R})$

Input: States s,t and relation $\mathcal{R}$
Output: Checks if $s \sim_{\mathcal{R}} t$

begin
  $S,T \leftarrow \emptyset$;
  forall $a \in \mathcal{A}$ **do**
    if $\mathcal{P}_{\mathcal{R}}^{s,a}$ *is strictly minimal* **then**
      $S \leftarrow S \cup \{\mathcal{P}_{\mathcal{R}}^{s,a}\}$;
    if $\mathcal{P}_{\mathcal{R}}^{t,a}$ *is strictly minimal* **then**
      $T \leftarrow T \cup \{\mathcal{P}_{\mathcal{R}}^{t,a}\}$;
  return $S \neq T$;

Fig. 2. Alternating probabilistic bisimulation algorithm for interval *MDP*s

Thus, the converse problem we are aiming to solve reduces to checking the existence of vectors $\boldsymbol{\mu}^{k,i}, \boldsymbol{\nu}^{k,i}, \widehat{\boldsymbol{\mu}}^{k,i}, \widehat{\boldsymbol{\nu}}^{k,i} \in \mathbb{R}^n_{\geq 0}$ and scalars $\lambda_i, \theta^{k,i}, \eta^{k,i}, \widehat{\theta}^{k,i}, \widehat{\eta}^{k,i} \in \mathbb{R}_{\geq 0}$ for each $1 \leq i \leq m$ satisfying (3)-(6) and $\sum_{i=1}^{m} \lambda_i = 1$. That amounts to solve an LP problem, which is known to be in **P**. □

As stated earlier, in order to compute $\sim_{(\exists\sigma\forall)}$ we follow the standard partition refinement approach formalized by the procedure BISIMULATION in Figure 2. Namely, we start with $\mathcal{R}$ being the complete relation and iteratively remove from $\mathcal{R}$ pairs of states that violate the definition of bisimulation with respect to $\mathcal{R}$. Clearly the core part of the algorithm is to check if two states "violate the definition of bisimulation". The violation of bisimilarity of s and t with respect to $\mathcal{R}$, which is addressed by the procedure VIOLATE, is checked by verifying if states s and t have the same set of strictly minimal polytopes. As a result of Theorem 8, this verification routine can be checked in polynomial time. As regards the computational complexity of Algorithm 1, let $|S| = n$ and $|\mathcal{A}| = m$. The procedure VIOLATE in Figure 2 is called at most n^3 times. The procedure VIOLATE is then linear in m and in the complexity of checking strict minimality of $\mathcal{P}_{\mathcal{R}}^{s,a}$ and $\mathcal{P}_{\mathcal{R}}^{t,a}$, which is in $\mathcal{O}(|\mathcal{M}|^{\mathcal{O}(1)})$. Putting all these together, we get the following result.

Theorem 9. *Given an IMDP $\mathcal{M}$, computing $\sim_{(\exists\sigma\forall)}$ belongs to $\mathcal{O}(|\mathcal{M}|^{\mathcal{O}(1)})$.*

5 Compositional Reasoning

In order to study the compositional minimization, that is, to split a complex *IMDP* as parallel composition of several simpler *IMDP*s and then to use the bisimulation as a means to reduce the size of each of these *IMDP*s before performing the model checking for a given PCTL formula φ, we have to extend the notion of bisimulation from one *IMDP* to a pair of *IMDP*s; we do this by

following the usual construction (see, e.g., [6, 40]). Given two *IMDP*s $\mathcal{M}_1$ and $\mathcal{M}_2$, we say that they are *alternating probabilistic* $(\exists\sigma\forall)$*-bisimilar*, denoted by $\mathcal{M}_1 \sim_{(\exists\sigma\forall)} \mathcal{M}_2$, if there exists an alternating probabilistic $(\exists\sigma\forall)$-bisimulation on the disjoint union of $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $\bar{s}_1 \sim_{(\exists\sigma\forall)} \bar{s}_2$. We can now establish the first property needed for the compositional minimization, that is, transitivity of $\sim_{(\exists\sigma\forall)}$:

Theorem 10. *Given three IMDPs $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$, whenever $\mathcal{M}_1 \sim_{(\exists\sigma\forall)} \mathcal{M}_2$ and $\mathcal{M}_2 \sim_{(\exists\sigma\forall)} \mathcal{M}_3$, then $\mathcal{M}_1 \sim_{(\exists\sigma\forall)} \mathcal{M}_3$.*

For the second property needed by the compositional minimization, that is, that $\sim_{(\exists\sigma\forall)}$ is preserved by the parallel composition operator, we first have to introduce such an operator; to this end, we consider a slight adaption of synchronous product of $\mathcal{M}_1$ and $\mathcal{M}_2$ as introduced in [22]. Such a synchronous product makes use of a subclass of the Segala's (simple) probabilistic automata [40, 41], called *action agnostic probabilistic automata* [22], where each automaton has as set of actions the same singleton set $\{f\}$, that is, all transitions are labelled by the same external action f: an *(action agnostic) probabilistic automaton* (*PA*) is a tuple $\mathcal{P} = (S, \bar{s}, \mathtt{AP}, L, D)$, where S is a set of *states*, $\bar{s} \in S$ is the *start* state, $\mathtt{AP}$ is a finite set of *atomic propositions*, $L\colon S \to 2^{\mathtt{AP}}$ is a *labelling function*, and $D \subseteq S \times \mathrm{Disc}(S)$ is a *probabilistic transition relation*.

Definition 11. *Given two IMDPs $\mathcal{M}_1$ and $\mathcal{M}_2$, we define the* synchronous product *of $\mathcal{M}_1$ and $\mathcal{M}_2$ as $\mathcal{M}_1 \otimes \mathcal{M}_2 := \mathtt{F}(\mathtt{UF}(\mathcal{M}_1) \otimes \mathtt{UF}(\mathcal{M}_2))$ where*

- *the unfolding mapping $\mathtt{UF}\colon$ IMDP $\to$ PA is a function that maps a given IMDP $\mathcal{M} = (S, \bar{s}, \mathcal{A}, \mathit{AP}, L, I)$ to the PA $\mathcal{P} = (S, \bar{s}, \mathit{AP}, L, D)$ where $D = \{\, (s, \mu) \mid s \in S, \exists a \in \mathcal{A}(s) : \mu \in \mathtt{Ext}(\mathcal{H}^a_s) \wedge \mathcal{H}^a_s$ is a strictly minimal polytope $\}$;*
- *the folding mapping $\mathtt{F}\colon$ PA $\to$ IMDP transforms a PA $\mathcal{P} = (S, \bar{s}, \mathit{AP}, L, D)$ into the IMDP $\mathcal{M} = (S, \bar{s}, \{f\}, \mathit{AP}, L, I)$ where, for each $s, t \in S$, $I(s, f, t) = \mathrm{proj}_{\mathbf{et}}\, \mathrm{conv}\, \{\, \mu \mid (s, \mu) \in D \,\}$;*
- *the* synchronous product *of two PAs $\mathcal{P}_1$ and $\mathcal{P}_2$, denoted by $\mathcal{P}_1 \otimes \mathcal{P}_2$, is the probabilistic automaton $\mathcal{P} = (S, \bar{s}, \mathit{AP}, L, D)$ where $S = S_1 \times S_2$, $\bar{s} = (\bar{s}_1, \bar{s}_2)$, $\mathit{AP} = \mathit{AP}_1 \cup \mathit{AP}_2$, for each $(s_1, s_2) \in S$, $L(s_1, s_2) = L_1(s_1) \cup L_2(s_2)$, and $D = \{\, ((s_1, s_2), \mu_1 \times \mu_2) \mid (s_1, \mu_1) \in D_1$ and $(s_2, \mu_2) \in D_2 \,\}$, where $\mu_1 \times \mu_2$ is defined for each $(t_1, t_2) \in S_1 \times S_2$ as $(\mu_1 \times \mu_2)(t_1, t_2) = \mu_1(t_1) \cdot \mu_2(t_2)$.*

As stated earlier, Definition 11 is slightly different from its counterpart in [22]. As a matter of fact, due to the competitive semantics for resolving the nondeterminism, only actions whose uncertainty set is a strictly minimal polytope play a role in deciding the alternating bisimulation relation $\sim_{(\exists\sigma\forall)}$. In particular, for the compositional reasoning keeping state actions whose uncertainty set is not strictly minimal induces spurious behaviors and therefore, influences on the soundness of the parallel operator definition. In order to avoid such redundancies, we can either preprocess the *IMDP*s before composing by removing state actions whose uncertainty set is not strictly minimal or restricting the unfolding mapping $\mathtt{UF}$ to unfold a given *IMDP* while ensuring that all extreme transitions in the resultant probabilistic automaton correspond to extreme points of strictly

minimal polytopes in the original *IMDP*. For the sake of simplicity, we choose the latter.

Theorem 12. *Given three IMDPs $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$, if $\mathcal{M}_1 \sim_{(\exists\sigma\forall)} \mathcal{M}_2$, then $\mathcal{M}_1 \otimes \mathcal{M}_3 \sim_{(\exists\sigma\forall)} \mathcal{M}_2 \otimes \mathcal{M}_3$.*

We have considered so far the parallel composition via synchronous production, which is working by the definition of folding collapsing all labels to a single transition. Here we consider the other extreme of the parallel composition: interleaving only.

Definition 13. *Given two IMDPs $\mathcal{M}_l$ and $\mathcal{M}_r$, we define the* interleaved composition *$\mathcal{M}_l \between \mathcal{M}_r$ of $\mathcal{M}_l$ and $\mathcal{M}_r$ as the IMDP $\mathcal{M} = (S, \bar{s}, \mathcal{A}, \mathtt{AP}, L, I)$ where*

- $S = S_l \times S_r$;
- $\bar{s} = (\bar{s}_l, \bar{s}_r)$;
- $\mathcal{A} = (\mathcal{A}_l \times \{l\}) \cup (\mathcal{A}_r \times \{r\})$;
- $\mathtt{AP} = \mathtt{AP}_l \cup \mathtt{AP}_r$;
- *for each $(s_l, s_r) \in S$, $L(s_l, s_r) = L_l(s_l) \cup L_r(s_r)$; and*
- $$I((s_l, s_r), (a, i), (t_l, t_r)) = \begin{cases} I_l(s_l, a, t_l) & \text{if } i = l \text{ and } t_r = s_r, \\ I_r(s_r, a, t_r) & \text{if } i = r \text{ and } t_l = s_l, \\ [0, 0] & \text{otherwise.} \end{cases}$$

Theorem 14. *Given three IMDPs $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$, if $\mathcal{M}_1 \sim_{(\exists\sigma\forall)} \mathcal{M}_2$, then $\mathcal{M}_1 \between \mathcal{M}_3 \sim_{(\exists\sigma\forall)} \mathcal{M}_2 \between \mathcal{M}_3$.*

6 Case Studies

We implemented in a prototypical tool the proposed bisimulation minimization algorithm and applied it to several case studies. The bisimulation algorithm is tested on several PRISM [29] benchmarks extended to support also intervals in the transitions. For the evaluation, we have used a machine with a 3.6 GHz Intel i7-4790 with 16 GB of RAM of which 12 assigned to the tool; the timeout has been set to 30 minutes. Our tool reads a model specification in the PRISM input language and constructs an explicit-state representation of the state space. Afterwards, it computes the quotient using the algorithm in Figure 2.

Table 1 shows the performance of our prototype on a number of case studies taken from the PRISM website [36], where we have replaced some of the probabilistic choices with intervals. Despite using an explicit representation for the model, the prototype is able to manage cases studies in the order of millions of states and transitions (columns "Model", "$|S|$", and "$|I|$"). The time in seconds required to compute the bisimulation relation and the size of the corresponding quotient *IMDP*are shown in columns "$t_\sim$", "$|S_\sim|$", and "$|I_\sim|$". In order to improve the performance of the tool, we have implemented optimizations, such as caching equivalent LP problems, which improve the runtime of our prototype.

Table 1. Experimental evaluation of the bisimulation computation

Model	$\|S\|$	$\|I\|$	$t_\sim$ (s)	$\|S_\sim\|$	$\|I_\sim\|$
Consensus-Shared-Coin-3	5 216	13 380	1	787	1 770
Consensus-Shared-Coin-4	43 136	144 352	3	2 189	5 621
Consensus-Shared-Coin-5	327 936	1 363 120	26	5 025	14 192
Consensus-Shared-Coin-6	2 376 448	11 835 456	238	10 173	30 861
Crowds-5-10	111 294	261 444	1	107	153
Crowds-5-20	2 061 951	7 374 951	20	107	153
Crowds-5-30	12 816 233	61 511 033	149	107	153
Crowds-5-40		–MO–			
Mutual-Exclusion-PZ-3	2 368	8 724	4	475	1 632
Mutual-Exclusion-PZ-4	27 600	136 992	70	3 061	13 411
Mutual-Exclusion-PZ-5	308 800	1 930 160	534	12 732	65 661
Mutual-Exclusion-PZ-6	3 377 344	25 470 144		–TO–	
Dining-Phils-LR-nofair-3	956	3 048	1	172	509
Dining-Phils-LR-nofair-4	9 440	40 120	14	822	3 285
Dining-Phils-LR-nofair-5	93 068	494 420	622	5 747	29 279
Dining-Phils-LR-nofair-6	917 424	5 848 524		–TO–	

Because of this, we saved to solve several LP problems in each tool run, thereby avoiding the potentially costly solution of LP problems from becoming a bottleneck. However, the more refinements are needed, the more time is required to complete the minimization, since several new LP problems need to be solved. The plots in Figure 3 show graphically the number of states and transitions for the Consensus and Crowds experiments, where for the latter we have considered more instances than the ones reported in Table 1. As we can see, the bisimulation minimization is able to reduce considerably the size of the *IMDP*, by several orders of magnitude. Additionally, this reduction correlates positively with the number of model parameters as depicted in Figure 4.

7 Concluding Remarks

In this paper, we have analyzed interval Markov decision processes under controller synthesis semantics in a dynamic setting. In particular, we provided an efficient compositional bisimulation minimization approach for *IMDP*s with respect to the competitive semantics, encompassing both the controller and parameter synthesis semantics. In this regard, we proved that alternating probabilistic bisimulation for *IMDP*s with respect to the competitive semantics can be decided in polynomial time. From perspective of compositional reasoning, we showed that alternating probabilistic bisimulations for *IMDP*s are congruences with respect to synchronous product and interleaving. Finally, we presented results obtained with a prototype tool on several case studies to show the effectiveness of the developed algorithm.

The core part of this algorithm relies on verifying strictly minimal polytopes in polynomial time, which depends on the special structure of the uncertainty

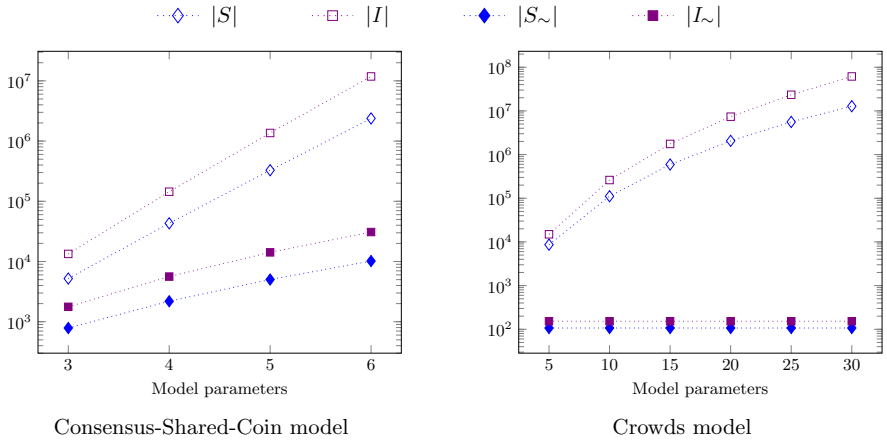


Fig. 3. Effectiveness of bisimulation minimization on model reduction

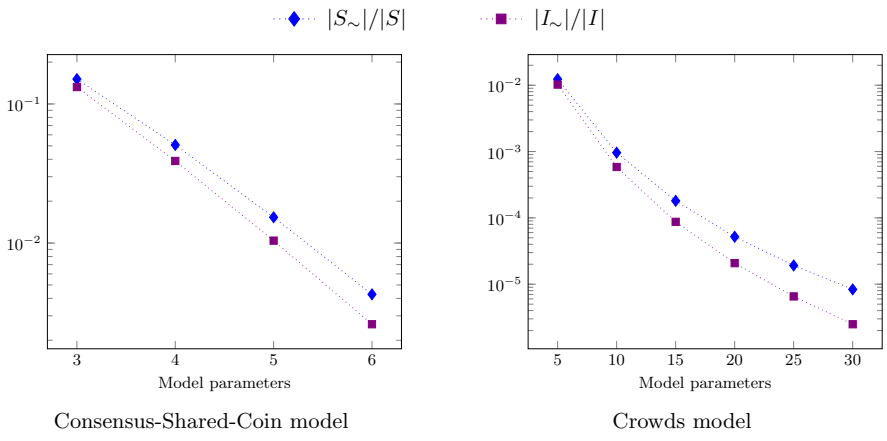


Fig. 4. State and transition reduction ratio by bisimulation minimization

polytopes. For future work, we aim to explore the possibility of preserving this computational efficiency for *MDP*s with richer formalisms for uncertainties such as likelihood or ellipsoidal uncertainties.

References

1. R. Alur, T. A. Henzinger, O. Kupferman, and M. Y. Vardi. Alternating refinement relations. In *CONCUR*, volume 1466 of *LNCS*, pages 163–178, 1998.
2. C. Baier and J.-P. Katoen. *Principles of Model Checking*. The MIT Press, 2008.
3. M. Ben-Or. Another advantage of free choice: Completely asynchronous agreement protocols (extended abstract). In *PODC*, pages 27–30, 1983.

4. M. Benedikt, R. Lenhardt, and J. Worrell. LTL model checking of interval Markov chains. In *TACAS*, volume 7795 of *LNCS*, pages 32–46, 2013.
5. E. Böde, M. Herbstritt, H. Hermanns, S. Johr, T. Peikenkamp, R. Pulungan, J. Rakow, R. Wimmer, and B. Becker. Compositional dependability evaluation for STATEMATE. *ITSE*, 35(2):274–292, 2009.
6. S. Cattani and R. Segala. Decision algorithms for probabilistic bisimulation. In *CONCUR*, volume 2421 of *LNCS*, pages 371–385, 2002.
7. K. Chatterjee, K. Sen, and T. A. Henzinger. Model-checking ω-regular properties of interval Markov chains. In *FoSSaCS*, volume 4962 of *LNCS*, pages 302–317, 2008.
8. G. Chehaibar, H. Garavel, L. Mounier, N. Tawbi, and F. Zulian. Specification and verification of the PowerScale® bus arbitration protocol: An industrial experiment with LOTOS. In *FORTE*, pages 435–450, 1996.
9. N. Coste, H. Hermanns, E. Lantreibecq, and W. Serwe. Towards performance prediction of compositional models in industrial GALS designs. In *CAV*, volume 5643 of *LNCS*, pages 204–218, 2009.
10. B. Delahaye, J.-P. Katoen, K. G. Larsen, A. Legay, M. L. Pedersen, F. Sher, and A. Wasowski. Abstract probabilistic automata. In *VMCAI*, volume 6538 of *LNCS*, pages 324–339, 2011.
11. B. Delahaye, J.-P. Katoen, K. G. Larsen, A. Legay, M. L. Pedersen, F. Sher, and A. Wasowski. New results on abstract probabilistic automata. In *ACSD*, pages 118–127, 2011.
12. B. Delahaye, K. G. Larsen, A. Legay, M. L. Pedersen, and A. Wasowski. Decision problems for interval Markov chains. In *LATA*, volume 6638 of *LNCS*, pages 274–285, 2011.
13. H. Fecher, M. Leucker, and V. Wolf. Don't know in probabilistic systems. In *SPIN*, volume 3925 of *LNCS*, pages 71–88, 2006.
14. M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, 1985.
15. D. Gebler, V. Hashemi, and A. Turrini. Computing behavioral relations for probabilistic concurrent systems. In *Stochastic Model Checking*, volume 8453 of *LNCS*, pages 117–155, 2014.
16. R. Givan, S. M. Leach, and T. L. Dean. Bounded-parameter Markov decision processes. *Artif. Intell.*, 122(1-2):71–109, 2000.
17. E. M. Hahn, T. Han, and L. Zhang. Synthesis for PCTL in parametric Markov decision processes. In *NFM*, volume 6617 of *LNCS*, pages 146–161, 2011.
18. E. M. Hahn, V. Hashemi, H. Hermanns, and A. Turrini. Exploiting robust optimization for interval probabilistic bisimulation. In *QEST*, volume 9826 of *LNCS*, pages 55–71, 2016.
19. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Asp. Comput.*, 6(5):512–535, 1994.
20. V. Hashemi, H. Hatefi, and J. Krčál. Probabilistic bisimulations for PCTL model checking of interval MDPs. In *SynCoP*, pages 19–33. EPTCS, 2014.
21. V. Hashemi, H. Hermanns, L. Song, K. Subramani, A. Turrini, and P. Wojciechowski. Compositional bisimulation minimization for interval Markov decision processes. In *LATA*, volume 9618 of *LNCS*, pages 114–126, 2016.
22. V. Hashemi, H. Hermanns, and A. Turrini. Compositional reasoning for interval Markov decision processes. Available at `http://arxiv.org/abs/1607.08484`.
23. H. Hermanns and J.-P. Katoen. Automated compositional Markov chain generation for a plain-old telephone system. *SCP*, 36(1):97–127, 2000.

24. G. N. Iyengar. Robust dynamic programming. *Math. Oper. Res.*, 30(2):257–280, 2005.
25. B. Jonsson and K. G. Larsen. Specification and refinement of probabilistic processes. In *LICS*, pages 266–277, 1991.
26. P. C. Kanellakis and S. A. Smolka. CCS expressions, finite state processes, and three problems of equivalence. *I&C*, pages 43–68, 1990.
27. J.-P. Katoen, T. Kemna, I. S. Zapreev, and D. N. Jansen. Bisimulation minimisation mostly speeds up probabilistic model checking. In *TACAS*, volume 4424 of *LNCS*, pages 76–92, 2007.
28. I. Kozine and L. V. Utkin. Interval-valued finite Markov chains. *Reliable Computing*, 8(2):97–113, 2002.
29. M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV*, volume 6806 of *LNCS*, pages 585–591, 2011.
30. M. Lahijanian, S. B. Andersson, and C. Belta. Formal verification and synthesis for discrete-time stochastic systems. *IEEE Tr. Autom. Contr.*, 60(8):2031–2045, 2015.
31. K. G. Larsen and A. Skou. Bisimulation through probabilistic testing (preliminary report). In *POPL*, pages 344–352, 1989.
32. R. Luna, M. Lahijanian, M. Moll, and L. E. Kavraki. Asymptotically optimal stochastic motion planning with temporal goals. In *WAFR*, pages 335–352, 2014.
33. R. Luna, M. Lahijanian, M. Moll, and L. E. Kavraki. Fast stochastic motion planning with optimality guarantees using local policy reconfiguration. In *ICRA*, pages 3013–3019, 2014.
34. R. Luna, M. Lahijanian, M. Moll, and L. E. Kavraki. Optimal and efficient stochastic motion planning in partially-known environments. In *AAAI*, pages 2549–2555, 2014.
35. R. Paige and R. E. Tarjan. Three partition refinement algorithms. *SIAM J. on Computing*, 16(6):973–989, 1987.
36. PRISM model checker. `http://www.prismmodelchecker.org/`.
37. A. Puggelli. *Formal Techniques for the Verification and Optimal Control of Probabilistic Systems in the Presence of Modeling Uncertainties*. PhD thesis, EECS Department, University of California, Berkeley, 2014.
38. A. Puggelli, W. Li, A. L. Sangiovanni-Vincentelli, and S. A. Seshia. Polynomial-time verification of PCTL properties of MDPs with convex uncertainties. In *CAV*, volume 8044 of *LNCS*, pages 527–542, 2013.
39. A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.
40. R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, MIT, 1995.
41. R. Segala. Probability and nondeterminism in operational models of concurrency. In *CONCUR*, volume 4137 of *LNCS*, pages 64–78, 2006.
42. K. Sen, M. Viswanathan, and G. Agha. Model-checking Markov chains in the presence of uncertainties. In *TACAS*, volume 3920 of *LNCS*, pages 394–410, 2006.
43. E. M. Wolff, U. Topcu, and R. M. Murray. Robust control of uncertain Markov decision processes with temporal logic specifications. In *CDC*, pages 3372–3379, 2012.