Compositional Bisimulation Minimization for Interval Markov Decision Processes

Vahid Hashemi[1,2(✉)], Holger Hermanns[2], Lei Song[3,4], K. Subramani[5], Andrea Turrini[4], and Piotr Wojciechowski[5]

[1] Max Planck Institute for Informatics, Saarbrücken, Germany
hashemi@mpi-inf.mpg.de
[2] Department of Computer Science, Saarland University, Saarbrücken, Germany
[3] University of Technology Sydney, Sydney, Australia
[4] State Key Laboratory of Computer Science, Institute of Software, CAS, Beijing, China
[5] LDCSEE, West Virginia University, Morgantown, WV, USA

Abstract. Formal verification of PCTL properties of MDPs with convex uncertainties has been recently investigated by Puggelli et al. However, model checking algorithms typically suffer from state space explosion. In this paper, we address probabilistic bisimulation to reduce the size of such an MDP while preserving PCTL properties it satisfies. We give a compositional reasoning over interval models to understand better the ways how large models with interval uncertainties can be composed. Afterwards, we discuss computational complexity of the bisimulation minimization and show that the problem is coNP-complete. Finally, we show that, under a mild condition, bisimulation can be computed in polynomial time.

Keywords: Markov decision process · Interval MDP · Compositionality · Bisimulation · Complexity

1 Introduction

Probability, nondeterminism, and uncertainty are three core aspects of real systems. *Probability* arises when a system, performing an action, is able to reach more than one state and we can estimate the proportion between reaching each of such states: probability can model both specific system choices (such as flipping a coin, commonly used in randomized distributed algorithms) and general system properties (such as message loss probabilities when sending a message over a wireless medium). *Nondeterminism* represents behaviors that we can not or we do not want to attach a precise (possibly probabilistic) outcome to. This might reflect the concurrent execution of several components at unknown (relative) speeds or behaviors we keep undetermined for simplifying the system or allowing for different implementations. *Uncertainty* relates to the fact that not all system parameters may be known exactly, including exact probability values.

Probabilistic automata (*PA*s) [32] extend classical concurrency models in a simple yet conservative fashion. In probabilistic automata, concurrent processes

A.-H. Dediu et al. (Eds.): LATA 2016, LNCS 9618, pp. 114–126, 2016.
DOI: 10.1007/978-3-319-30000-9_9

may perform probabilistic experiments inside a transition. Labeled transition systems are instances of this model family, obtained by restricting to Dirac distributions (assigning full probability to single states). Thus, foundational concepts and results of standard concurrency theory are retained in full and extend smoothly to the *PA* model. *PA*s are akin to *Markov decision processes* (*MDP*s), their fundamental beauty can be paired with powerful model checking techniques, as implemented for instance in the PRISM tool [27].

In *PA*s and *MDP*s, probability values need to be specified precisely. This is often an impediment to their applicability to real systems. Instead it appears more viable to specify ranges of probabilities, so as to reflect the uncertainty in these values. This leads to a model where intervals of probability values replace probabilities. This is the model studied in this paper, we call it *interval Markov decision processes*, *IMDP*s.

In standard concurrency theory, *bisimulation* plays a central role as the undisputed reference for distinguishing the behaviour of systems. Besides for distinguishing systems, bisimulation relations conceptually allow us to reduce the size of a behaviour representation without changing its properties (i.e., with respect to logic formulae the representation satisfies). This is particularly useful to alleviate the state explosion problem notoriously encountered in model checking. If the bisimulation is a congruence with respect to a parallel composition operator used to build up the model out of smaller ones, this can give rise to a compositional strategy to associate a small model to a large system without intermediate state space explosion. In several related settings, this strategy has been proven very effective [9,18]. In order to be of practical use also for *IMDP*s, efficient bisimulation decision procedures are required.

This paper discusses the key ingredients for this to work for bisimulation on *IMDP*s. On the one hand, we discuss congruence properties, on the other hand we discuss the complexity of deciding bisimulation on *IMDP*s. More precisely, on the one hand we show that the probabilistic bisimulation for *IMDP*s proposed in [16] for the cooperative resolution of the nondeterminism in a dynamic setting is preserved by parallel composition and that it is transitive, so it is indeed a congruence. On the other hand, we show that deciding probabilistic bisimulation for *IMDP*s is in general coNP-complete; under a mild restriction on the amount of nondeterminism in the *IMDP*, the bisimulation becomes polynomial.

Related Work. Various probabilistic formalisms with uncertain transitions are studied in the literature. Uncertain *MDP*s [28,30,37] allow more general sets of distributions to be associated with each transition, not only those described by intervals. Usually, they are restricted to *rectangular uncertainty sets* requiring that the uncertainty is linear and independent for any two transitions of any two states. Our general algorithm working with polytopes can be easily adapted to this setting. Parametric *MDP*s [14] instead allow such dependencies as every probability is described as a rational function of a finite set of global parameters.

From the compositional specification point of view, Interval *MC*s [21,25] and Abstract *PA*s [10] serve as specification theories for *MC* and *PA*s featuring satisfaction relation, and various refinement relations. In order to be closed under

parallel composition, Abstract *PA*s allow general polynomial constraints on probabilities instead of interval bounds. Since for Interval *MC*s it is not possible to explicitly construct parallel composition, the problem of whether there is a common implementation of a set of Interval *MC*s is addressed instead [11]. To the contrary, interval bounds on *rates* of outgoing transitions work well with parallel composition in the continuous-time setting of Abstract Interactive *MC*s [24]. The reason is that unlike probabilities, rates do not need to sum up to 1. Authors of [39] successfully define parallel composition for interval models by separating synchronizing transitions from the transitions with uncertain probabilities.

Probabilistic bisimulation for uncertain probabilistic models has been studied quite recently in [16]. To the best of our knowledge, we are not aware of any other existing results on probabilistic bisimulations for uncertain or parametric models. Among similar concepts studied in the literature are simulation [39] and refinement [10,21] relations for the previously mentioned models.

Many new verification algorithms for interval models appeared in last few years. Reachability and expected total reward is addressed for Interval *MC*s [8] as well as *IMDP*s [38]. PCTL model checking and PLTL model checking are studied for Interval *MC*s [6,8] and also for *IMDP*s [30,37]. Among other technical tools, all these approaches make use of (robust) dynamic programming relying on the fact that transition probability distributions are resolved dynamically. For the static resolution of distributions, adaptive discretization technique for PCTL parameter synthesis is given in [14]. Uncertain models are also widely studied in the control community (see, e.g. [28,38]), mainly interested in maximal expected finite-horizon/discounted reward.

Organization of the paper. We start with necessary preliminaries in Sect. 2. In Sect. 3, we give the definition of probabilistic bisimulation for *IMDP*s and discuss the main results of [16]. Furthermore, we show that the probabilistic bisimulation over *IMDP*s is compositional and transitive. In Sect. 4, we discuss the hardness of deciding probabilistic bisimulation and also show that polynomiality is achievable under a mild condition. Finally, in Sect. 5 we conclude the paper.

2 Preliminaries

Given $n \in \mathbb{N}$, we denote by $\mathbf{1} \in \mathbb{R}^n$ the unit vector and by $\mathbf{1}^T$ its transpose. In the sequel, the comparison between vectors is element-wise and all vectors are column ones unless otherwise stated. For a given set $P \subseteq \mathbb{R}^n$, we denote by $\mathrm{CH}(P)$ the convex hull of P. We denote by $\mathbb{I}$ is a set of closed subintervals of $[0,1]$ and, for a given $[a,b] \in \mathbb{I}$, we let $\inf[a,b] = a$ and $\sup[a,b] = b$.

For a given set X, we denote by $\Delta(X)$ the set of discrete probability distributions over X. For an equivalence relation $\mathcal{R}$ on X and $\rho_1, \rho_2 \in \Delta(X)$, we write $\rho_1 \; \mathcal{L}(\mathcal{R}) \; \rho_2$ if for each $\mathcal{C} \in X/\mathcal{R}$, it holds that $\rho_1(\mathcal{C}) = \rho_2(\mathcal{C})$. By abuse of notation, we extend $\mathcal{L}(\mathcal{R})$ to distributions over $X/\mathcal{R}$, i.e., for $\rho_1, \rho_2 \in \Delta(X/\mathcal{R})$, we write $\rho_1 \; \mathcal{L}(\mathcal{R}) \; \rho_2$ if for each $\mathcal{C} \in X/\mathcal{R}$, it holds that $\rho_1(\mathcal{C}) = \rho_2(\mathcal{C})$.

2.1 Interval Markov Decision Processes

Let us formally define Interval Markov Decision Processes.

Definition 1. *An* Interval Markov Decision Process *(IMDP)* $\mathcal{M}$ *is a tuple* $\mathcal{M} = (S, \bar{s}, \mathcal{A}, \mathtt{AP}, L, I)$*, 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* $I \colon S \times \mathcal{A} \times S \to \mathbb{I}$ *is an* interval transition probability function *such that for each* s*, there exist* a *and* s' *such that* $I(s, a, s') \neq [0, 0]$*.*

We denote by $\mathcal{A}(s)$ the set of actions that are enabled from state s, i.e., $\mathcal{A}(s) = \{\, a \in \mathcal{A} \mid \exists s' \in S. I(s, a, s') \neq [0, 0] \,\}$. Furthermore, for each state s and action $a \in \mathcal{A}(s)$, we denote by $s \xrightarrow{a} \mu_s$ that $\mu_s \in \Delta(S)$ is a *feasible distribution*, i.e., for each state s' we have $\mu_s(s') \in I(s, a, s')$. We require that the set $\{\, \mu_s \mid s \xrightarrow{a} \mu_s \,\}$, also denoted by $\mathcal{P}^{s,a}$, is non-empty for each state s and action $a \in \mathcal{A}(s)$. We denote by $b_{\mathcal{M}}$ the *branching* of $\mathcal{M}$, where $b_{\mathcal{M}} = \max_{s \in S}\{|\mathcal{A}(s)|\}$.

An *IMDP* is initiated in some state s_1 and then moves in discrete steps from state to state forming an infinite path $s_1\, s_2\, s_3 \ldots$. One step, say from state s_i, is performed as follows. First, an action $a \in \mathcal{A}(s)$ is chosen nondeterministically by *scheduler*. Then, *nature* resolves the uncertainty and chooses nondeterministically one corresponding feasible distribution $\mu_{s_i} \in \mathcal{P}^{s_i,a}$. Finally, the next state s_{i+1} is chosen randomly according to the distribution μ_{s_i}.

Let us define the semantics of an *IMDP* formally. A *path* is a finite or infinite sequence of states $\omega = s_1\, s_2 \cdots$. For a finite path ω, we denote by $last(\omega)$ the last state of ω. The set of all finite paths and the set of all infinite paths are denoted by $Paths_{fin}$ and $Paths_{inf}$, respectively. Furthermore, let $Paths_{\omega} = \{\omega\omega' \mid \omega' \in Paths_{inf}\}$ denote the set of paths that have the finite prefix $\omega \in Paths_{fin}$.

Definition 2. *A* scheduler *is a function* $\sigma \colon Paths_{fin} \to \Delta(\mathcal{A})$ *that to each finite path* ω *assigns a distribution over the set of actions. A* nature *is a function* $\pi \colon Paths_{fin} \times \mathcal{A} \to \Delta(S)$ *that to each finite path* ω *and action* $a \in \mathcal{A}(last(\omega))$ *assigns a feasible distribution, i.e., an element of* $\mathcal{P}^{s,a}$ *where* $s = last(\omega)$*. We denote by* Σ *the set of all schedulers and by* Π *the set of all natures.*

For a state s, a scheduler σ, and a nature π, let $\Pr_s^{\sigma,\pi}$ denote the unique probability measure over $(Paths_{inf}, \mathcal{B})$[1] such that the probability $\Pr_s^{\sigma,\pi}[Paths_{s'}]$ of starting in s' equals 1 if $s' = s$ and 0, otherwise; and the probability $\Pr_s^{\sigma,\pi}[Paths_{\omega s'}]$ of traversing a finite path $\omega s'$ equals $\Pr_s^{\sigma,\pi}[Paths_{\omega}] \cdot \sum_{a \in \mathcal{A}} \sigma(\omega)(a) \cdot \pi(\omega, a)(s')$.

Observe that the scheduler does not choose an action but a *distribution* over actions. It is well-known [32] that such randomization brings more power in the context of bisimulations. Note that for nature this is not the case, since $\mathcal{P}^{s,a}$ is closed under convex combinations, thus nature can choose all distributions.

[1] Here, $\mathcal{B}$ is the standard σ-algebra over $Paths_{inf}$ generated from the set of all cylinder sets $\{Paths_{\omega} \mid \omega \in Paths_{fin}\}$. The unique probability measure is obtained by the application of the extension theorem (see, e.g. [3]).

2.2 Probabilistic Computation Tree Logic (PCTL)

There are various ways how to describe properties of interval *MDP*s. Here we focus on *probabilistic CTL* (PCTL) [15]. The syntax of PCTL state formulas φ and PCTL path formulas ψ is given by:

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

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

The satisfaction relation for PCTL formulae depends on the way how nondeterminism is resolved for the probabilistic operator $\mathsf{P}_{\bowtie p}(\psi)$. When quantifying both the nondeterminisms universally, we define the satisfaction relation $s \models_c \varphi$ as follows: $s \models_c x$ if $x \in L(s)$; $s \models_c \neg\varphi$ if not $s \models_c \varphi$; $s \models_c \varphi_1 \wedge \varphi_2$ if both $s \models_c \varphi_1$ and $s \models_c \varphi_2$; and

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

where the satisfaction relation $\omega \models_c \psi$ for an infinite path $\omega = s_1 s_2 \cdots$ and a path formula ψ is given by:

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

It is easy to show that the set $\models_c \psi$ is measurable for any path formula ψ, hence the definition is mathematically well defined.

3 Probabilistic Bisimulation for Interval MDPs

We now recall the main results on probabilistic bisimulation for *IMDP*s, as developed in [16]. In this work, we consider the notion of probabilistic bisimulation for the cooperative resolution of nondeterminism. This semantics is very natural in the context of verification of parallel systems with uncertain transition probabilities in which we assume that scheduler and nature are resolved *cooperatively* in the most *adversarial* way. Moreover, resolution of a feasible probability distribution respecting the interval constraints can be either done *statically* [21], i.e., at the beginning once for all, or *dynamically* [20,33], i.e., independently for each computation step. In this paper, we focus on dynamic approach in resolving the stochastic nondeterminism that is easier to work with algorithmically and can be seen as a relaxation of the static approach that is often intractable [2,6].

Let $s \longrightarrow \mu_s$ denote that a transition from s to μ_s can be taken cooperatively, i.e., that there is a scheduler $\sigma \in \Sigma$ and a nature $\pi \in \Pi$ such that $\mu_s = \sum_{a \in \mathcal{A}(s)} \sigma(s)(a) \cdot \pi(s,a)$. In other words, $s \longrightarrow \mu_s$ if $\mu_s \in \mathrm{CH}(\bigcup_{a \in \mathcal{A}(s)} \mathcal{P}^{s,a})$.

Definition 3 (cf. [16]). *Given an IMDP $\mathcal{M}$, let $\mathcal{R} \subseteq S \times S$ be an equivalence relation. We say that $\mathcal{R}$ is a* probabilistic bisimulation *if for each $(s,t) \in \mathcal{R}$ we have that $L(s) = L(t)$ and for each $s \longrightarrow \mu_s$ there exists $t \longrightarrow \mu_t$ such that $\mu_s \, \mathcal{L}(\mathcal{R}) \, \mu_t$. Furthermore, we write $s \sim_c t$ if there is a probabilistic bisimulation $\mathcal{R}$ such that $(s,t) \in \mathcal{R}$.*

Intuitively, each (cooperative) step of scheduler and nature from state s needs to be matched by a (cooperative) step of scheduler and nature from state t; symmetrically, s also needs to match t. It is shown in [16] that $\sim_c$ preserves the (cooperative) universally quantified PCTL satisfaction $\models_c$. More precisely,

Theorem 4 (cf. [16]). *For states $s \sim_c t$ and any PCTL formula φ, we have $s \models_c \varphi$ if and only if $t \models_c \varphi$.*

Computation of probabilistic bisimulation for *IMDP*s follows the standard partition refinement approach [22,29]. However, the core part of the algorithm is to find out whether two states "violate the definition of bisimulation". Verification of this violation, however, amounts to check the inclusion of polytopes defined as follows. For $s \in S$ and an action $a \in \mathcal{A}$, recall that $\mathcal{P}^{s,a}$ denotes the polytope of feasible successor distributions over *states* with respect to taking the action a in the state s. By $\mathcal{P}^{s,a}_{\mathcal{R}}$, 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. Formally, for $\mu \in \Delta(S/\mathcal{R})$ we set $\mu \in \mathcal{P}^{s,a}_{\mathcal{R}}$ if

$$\mu(\mathcal{C}) \in \left[\sum_{s' \in \mathcal{C}} \inf I(s,a,s'), \sum_{s' \in \mathcal{C}} \sup I(s,a,s') \right] \qquad \text{for each } \mathcal{C} \in S/\mathcal{R} \, .$$

Furthermore, we define $\mathcal{P}^{s}_{\mathcal{R}} = \mathrm{CH}(\bigcup_{a \in \mathcal{A}(s)} \mathcal{P}^{s,a}_{\mathcal{R}})$, the set of feasible successor distributions over $S/\mathcal{R}$ with respect to taking an *arbitrary* distribution over enabled actions in state s. As specified in [16], checking violation of a given pair of states amounts to check equality of the corresponding constructed polytopes for the states. As regards the computational complexity of the proposed algorithm, the following theorem indicates that it is fixed parameter tractable. Formally,

Theorem 5 (cf. [16]). *Computing $\sim_c$ on an IMDP $\mathcal{M}$ is in $|\mathcal{M}|^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(f)}$ where $|\mathcal{M}| = |S|^2 \cdot |\mathcal{A}|$ and $f = \max_{s \in S, a \in \mathcal{A}(s)} |\{\, s' \in S \mid I(s,a,s') \neq [0,0] \,\}|$.*

3.1 Compositional Reasoning

The compositional reasoning is a widely used technique (see, e.g., [7,18,23]) that permits to deal with large systems. In particular, a large system is decomposed into multiple components running in parallel; such components are then minimized by replacing each of them by a bisimilar but smaller one so that the overall behaviour remains unchanged. In order to apply this technique, bisimulation has first to be extended to pairs of components and then to be shown to be transitive and preserved by the parallel composition operator. The extension to a pair of components is trivial and commonly done (see, e.g., [4,32]):

Definition 6. *Given two IMDPs $\mathcal{M}_1$ and $\mathcal{M}_2$, we say that they are* probabilistic bisimilar*, denoted by $\mathcal{M}_1 \sim_c \mathcal{M}_2$, if there exists a probabilistic bisimulation on the disjoint union of $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $\bar{s}_1 \sim_c \bar{s}_2$.*

The next step is to define the parallel composition for *IMDP*s:

Definition 7. *Given two IMDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ with $\mathcal{M}_i = (S_i, \bar{s}_i, \mathcal{A}_i, \mathtt{AP}_i, L_i, I_i)$, $i = 1, 2$, we define the* parallel composition *of $\mathcal{M}_1$ and $\mathcal{M}_2$, denoted by $\mathcal{M}_1 \parallel \mathcal{M}_2$ as the IMDP $(S, \bar{s}, \mathcal{A}, \mathtt{AP}, L, I)$ where $S = S_1 \times S_2$, $\bar{s} = (\bar{s}_1, \bar{s}_2)$, $\mathcal{A} = \mathcal{A}_1 \times \mathcal{A}_2$, $\mathtt{AP} = \mathtt{AP}_1 \cup \mathtt{AP}_2$, for each $(s_1, s_2) \in S$, $L(s_1, s_2) = L_1(s_1) \cup L_2(s_2)$, and for each $(s_1, s_2) \in S$, $(a_1, a_2) \in \mathcal{A}$, and $(t_1, t_2) \in S$, $I((s_1, s_2), (a_1, a_2), (t_1, t_2)) = I_1(s_1, a_1, t_1) \times I_2(s_2, a_2, t_2) = [l_1 \cdot l_2, u_1 \cdot u_2]$ where $[l_1, u_1] = I_1(s_1, a_1, t_1)$ and $[l_2, u_2] = I_2(s_2, a_2, t_2)$.*

Proposition 8. *For each IMDP $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$, if $\mathcal{M}_1 \sim_c \mathcal{M}_2$, then $\mathcal{M}_1 \parallel \mathcal{M}_3 \sim_c \mathcal{M}_2 \parallel \mathcal{M}_3$.*

Another property that is needed for $\sim_c$ in order to support the compositional reasoning is that it has to be transitive. This is indeed a property of $\sim_c$, as stated by the following proposition:

Proposition 9. *For each IMDP $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$, if $\mathcal{M}_1 \sim_c \mathcal{M}_2$ and $\mathcal{M}_2 \sim_c \mathcal{M}_3$, then $\mathcal{M}_1 \sim_c \mathcal{M}_3$.*

By being transitive and preserved by parallel composition, $\sim_c$ fully supports the compositional verification of complex systems.

4 Hardness and Tractability

Definition 3 is the central definition around which the paper revolves. Given an *IMDP* $\mathcal{M}$, the complexity of computing $\sim_c$ strictly depends on checking bisimilarity of a pair of states as a core part: in this section we will show that this verification routine is coNP-complete and therefore, the computation of $\sim_c$ as a whole is coNP-complete. Later on, we show how an equivalence relation can be computed in polynomial time under a mild condition. The definition of bisimulation can be reformulated equivalently as follows:

Definition 10. *Let $\mathcal{R} \subseteq S \times S$ be an equivalence relation. We say that $\mathcal{R}$ is a* probabilistic bisimulation *if $(s, t) \in \mathcal{R}$ implies that $L(s) = L(t)$ and $\mathcal{P}^s_{\mathcal{R}} = \mathcal{P}^t_{\mathcal{R}}$.*

As it is clear from Definition 10, the complexity of verifying bisimilarity of a pair of states s and t strictly depends on the complexity of the *Convex Hull Equivalence (CHE) problem* stated as follows:

Definition 11. *Given an IMDP $\mathcal{M}$, a pair of states s and t, $n, n_s, n_t \in \mathbb{N}$, two sets $\{ P^{s,i} \mid i \in \{1, \ldots, n_s\} \}$ and $\{ P^{t,i} \mid i \in \{1, \ldots, n_t\} \}$ where for each $r \in \{s, t\}$ and $i \in \{1, \ldots, n_r\}$, for given $\boldsymbol{l}^{r,i}, \boldsymbol{u}^{r,i} \in \mathbb{R}^n$, $P^{r,i}$ is the convex polyhedron*

$$P^{r,i} = \left\{ \boldsymbol{x}^{r,i} \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \boldsymbol{l}^{r,i} \leq \boldsymbol{x}^{r,i} \leq \boldsymbol{u}^{r,i} \\ \mathbf{1}^T \boldsymbol{x}^{r,i} = 1 \end{array} \right\},$$

the CHE problem *asks to determine whether* $\mathrm{CH}(\bigcup_{i=1}^{n_s} P^{s,i}) = \mathrm{CH}(\bigcup_{i=1}^{n_t} P^{t,i})$.

Theorem 12. *The CHE problem is coNP-complete.*

Proof (Sketch). We show that CHE problem is in coNP by reducing it to the coNP quantified linear implication problem [12] and that it is coNP-hard by reducing the coNP-hard tautology problem for 3DNF [13] to CHE. □

With this result at hand, together with Definition 10, we know that checking the bisimilarity of two states of an *IMDP* $\mathcal{M}$ is coNP-complete. Since the standard partition refinement algorithm performs this check a polynomial number of times (see, e.g., [4,5,22,29,35]), it follows that also computing $\sim_c$ is coNP-complete.

Theorem 13. *Given an IMDP* $\mathcal{M}$*, computing* $\sim_c$ *is coNP-complete.*

4.1 Decision Algorithm

In the previous section we have shown that the problem of deciding whether two states of an *IMDP* are bisimilar is coNP-complete. In this section we show that, under a mild restriction on *IMDP*s, the problem is solvable in polynomial time.

In the following we present a polynomial algorithm for deciding $\sim_c$ on *IMDP*s, provided its branching is considered as a constant. Given an *IMDP* $\mathcal{M}$ as in Definition 1 and an equivalence relation $\mathcal{R} \subseteq S \times S$, according to Definition 10, $\mathcal{R}$ is a probabilistic bisimulation iff for any $(s,t) \in \mathcal{R}$, $\mathcal{P}^s_{\mathcal{R}} = \mathcal{P}^t_{\mathcal{R}}$. Our algorithm relies on an algorithm in [34] to check efficiently whether $\mathcal{P}^s_{\mathcal{R}} = \mathcal{P}^t_{\mathcal{R}}$. However, the algorithm in [34] requires that both $\mathcal{P}^s_{\mathcal{R}}$ and $\mathcal{P}^t_{\mathcal{R}}$ are represented in form of $\mathcal{H}$-polytopes, i.e., defined as intersection of a finite number of closed half-spaces. Note $\mathcal{P}^s_{\mathcal{R}} = \mathrm{CH}(\bigcup_{a \in \mathcal{A}(s)} \mathcal{P}^{s,a}_{\mathcal{R}})$. Therefore, we focus on the procedure of obtaining an equivalent $\mathcal{H}$-polytope of $\mathcal{P}^s_{\mathcal{R}}$ given all $\mathcal{P}^{s,a}_{\mathcal{R}}$ being represented as $\mathcal{H}$-polytopes.

Now we show how to represent each $\mathcal{P}^{s,a}_{\mathcal{R}}$ as an $\mathcal{H}$-polytope. To simplify our presentation, we shall fix an order over all equivalence classes in $S/\mathcal{R}$. By doing so, any distribution $\rho \in \Delta(S/\mathcal{R})$ can be seen as a vector $\boldsymbol{v}$ such that $\boldsymbol{v}_i = \rho(C_i)$ for each $1 \leq i \leq n$, where $n = |S/\mathcal{R}|$, C_i is the i-th equivalence class, and $\boldsymbol{v}_i$ the i-th element in $\boldsymbol{v}$. For the above discussion, $\rho \in \mathcal{P}^{s,a}_{\mathcal{R}}$ iff $\rho(\mathcal{C}_i) \in [\boldsymbol{l}^a_i, \boldsymbol{u}^a_i]$ for any $1 \leq i \leq n$ and $\rho \in \Delta(S/\mathcal{R})$, where $\boldsymbol{l}^a$ and $\boldsymbol{u}^a$ are vectors whose components are $\boldsymbol{l}^a_i = \sum_{s' \in \mathcal{C}_i} \inf I(s,a,s')$ and $\boldsymbol{u}^a_i = \sum_{s' \in \mathcal{C}_i} \sup I(s,a,s')$ for each $1 \leq i \leq n$, respectively. Therefore, $\mathcal{P}^{s,a}_{\mathcal{R}}$ corresponds to an $\mathcal{H}$-polytope defined by:

$$\left\{ \boldsymbol{x}^a \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \boldsymbol{l}^a \leq \boldsymbol{x}^a \leq \boldsymbol{u}^a \\ \mathbf{1}^T \boldsymbol{x}^a = 1 \end{array} \right\}. \tag{1}$$

We have represented each $\mathcal{P}^{s,a}_{\mathcal{R}}$ for $a \in \mathcal{A}(s)$ as an $\mathcal{H}$-polytope. It is remained to show that $\mathcal{P}^s_{\mathcal{R}} = \mathrm{CH}(\bigcup_{a \in \mathcal{A}(s)} \mathcal{P}^{s,a}_{\mathcal{R}})$ can be represented by an $\mathcal{H}$-polytope as well. For this, we use the Balas Extension Theorem [1] in order to compute the $\mathcal{H}$-representation of the convex hull in polynomial time. The theorem allows for additional variables and thus maps the original space of variables to a higher dimensional space. By definition, $\boldsymbol{x} \in \mathcal{P}^s_{\mathcal{R}}$ iff there exist $\lambda^a \in \mathbb{R}$ and $\boldsymbol{x}^a \in \mathcal{P}^{s,a}_{\mathcal{R}}$

for each $a \in \mathcal{A}(s)$ such that $\boldsymbol{x} = \sum_{a\in\mathcal{A}(s)} \lambda^a \boldsymbol{x}^a$, $\lambda^a \geq 0$, and $\sum_{a\in\mathcal{A}(s)} \lambda^a = 1$. Therefore, together with Eq. (1), we have

$$\mathcal{P}^s_{\mathcal{R}} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} \exists \lambda^a \geq 0. \exists \boldsymbol{x}^a. & \forall a \in \mathcal{A}(s) \\ \boldsymbol{x} = \sum_{a\in\mathcal{A}(s)} \lambda^a \boldsymbol{x}^a & \\ \mathbf{1}^T \boldsymbol{x}^a = 1 & \forall a \in \mathcal{A}(s) \\ \boldsymbol{l}^a \leq \boldsymbol{x}^a \leq \boldsymbol{u}^a & \forall a \in \mathcal{A}(s) \\ \sum_{a\in\mathcal{A}(s)} \lambda^a = 1 & \end{array} \right\}. \tag{2}$$

Unfortunately, the constraint $\boldsymbol{x} = \sum_{a\in\mathcal{A}(s)} \lambda^a \boldsymbol{x}^a$ of Eq. (2) is not linear, hence is not a desirable $\mathcal{H}$-representation of $\mathcal{P}^s_{\mathcal{R}}$. In order to avoid non-linearity in the representation, we let $\boldsymbol{t}^a = \lambda^a \boldsymbol{x}^a$ for each $a \in \mathcal{A}(s)$. Constraints in Eq. (2) can be rewritten to an equivalent form as follows:

$$\mathcal{P}^s_{\mathcal{R}} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} \exists \lambda^a \geq 0. \exists \boldsymbol{t}^a. & \forall a \in \mathcal{A}(s) \\ \boldsymbol{x} = \sum_{a\in\mathcal{A}(s)} \boldsymbol{t}^a & \\ \mathbf{1}^T \boldsymbol{t}^a = \lambda^a & \forall a \in \mathcal{A}(s) \\ \lambda^a \boldsymbol{l}^a \leq \boldsymbol{t}^a \leq \lambda^a \boldsymbol{u}^a & \forall a \in \mathcal{A}(s) \\ \sum_{a\in\mathcal{A}(s)} \lambda^a = 1 & \end{array} \right\}. \tag{3}$$

Alternatively, Eq. (3) can be seen as an $\mathcal{H}$-polytope over variables $\boldsymbol{x}$, $\boldsymbol{t}^a$, and λ^a (where $a \in \mathcal{A}(s)$), whose projection on the space of variables $\boldsymbol{x}$ gives exactly the $\mathcal{H}$-polytope corresponding to $\mathcal{P}^s_{\mathcal{R}}$. We first try to project out variables $\boldsymbol{t}^a$ for all $a \in \mathcal{A}(s)$. According to Eq. (3), $\boldsymbol{x} = \sum_{a\in\mathcal{A}(s)} \boldsymbol{t}^a$, so all constraints related to $\boldsymbol{t}^a$ can be combined. As a result, we obtain:

$$\mathcal{P}^s_{\mathcal{R}} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} \exists \lambda^a \geq 0. & \forall a \in \mathcal{A}(s) \\ \mathbf{1}^T \boldsymbol{x} = 1 & \\ \sum_{a\in\mathcal{A}(s)} \lambda^a \boldsymbol{l}^a \leq \boldsymbol{x} \leq \sum_{a\in\mathcal{A}(s)} \lambda^a \boldsymbol{u}^a & \\ \sum_{a\in\mathcal{A}(s)} \lambda^a = 1 & \end{array} \right\}. \tag{4}$$

In Eq. (4), we still have extra variables λ^a, which should be eliminated in order to obtain an $\mathcal{H}$-polytope representing $\mathcal{P}^s_{\mathcal{R}}$. For this, we apply the well-known Fourier-Motzkin (FM) elimination method. The main idea of the FM elimination is to partition all inequalities relevant to y into two sets: $\{\sum_{1\leq j\leq n} e_{ij} x_j \leq y\}_{1\leq i\leq m_1}$ and $\{\sum_{1\leq j\leq n} e_{ij} x_j \geq y\}_{m_1<i\leq m}$, where e_{ij} $(1 \leq i \leq m, 1 \leq j \leq n)$ are coefficients and y is the variable to be eliminated. The resultant set of inequalities will contain those in form of $\sum_{1\leq j\leq n} e_{ij} x_j \leq \sum_{1\leq j\leq n} e_{i'j} x_j$ for each $1 \leq i \leq m_1$ and $m_1 < i' \leq m$, which defines a projection of the original $\mathcal{H}$-polytope on variables $\{x_j\}_{1\leq j\leq n} \cup \{y\}$ to an $\mathcal{H}$-polytope on $\{x_j\}_{1\leq j\leq n}$. More details can be found in e.g. [26].

However, the FM elimination causes an exponential blow-up and results in $4(\frac{m}{4})^{2^d}$ inequalities in the worst case, where m is the number of inequalities in the original representation and d the number of variables having been eliminated [31]. Given an *IMDP* $\mathcal{M}$, the number of inequalities in Eq. (4) is in $\mathcal{O}(S)$ and the number of λ^a is upper-bounded by the branching of $\mathcal{M}$. Therefore, if we assume

$b_{\mathcal{M}}$ to be a constant, FM elimination will not cause exponentially blow-up after removing all λ^a in Eq. (4) and we obtain an $\mathcal{H}$-polytope representing $\mathcal{P}^s_{\mathcal{R}}$, the number of inequalities in which is at most $4(\frac{|S|}{4})^{2^d}$ with $d = b_{\mathcal{M}}$.

Given we have obtained $\mathcal{H}$-polytopes of both $\mathcal{P}^s_{\mathcal{R}}$ and $\mathcal{P}^t_{\mathcal{R}}$, to check whether $\mathcal{P}^s_{\mathcal{R}} = \mathcal{P}^t_{\mathcal{R}}$, the convex set inclusion checking algorithm in [34, Algorithm 1] can be applied to decide whether $\mathcal{P}^s_{\mathcal{R}} \subseteq \mathcal{P}^t_{\mathcal{R}}$ and $\mathcal{P}^t_{\mathcal{R}} \subseteq \mathcal{P}^s_{\mathcal{R}}$ hold, which can be done in polynomial time. Our algorithm for deciding $\sim_c$ mainly follows the partition-refinement approach [4,17,22,35], the key ingredient of which is a refinement procedure that keeps refining a relation $\mathcal{R}$ until for every pair of states s and t in a same equivalence class, $\mathcal{P}^s_{\mathcal{R}} = \mathcal{P}^t_{\mathcal{R}}$. Let $L(n, m)$ denote the polynomial running time of an LP algorithm on n inequalities and m variables [36]. The complexity of the algorithm for deciding $\sim_c$ is shown in the following theorem.

Theorem 14. *Given an IMDP $\mathcal{M}$, $\sim_c$ on $\mathcal{M}$ can be decided in time $\mathcal{O}(|S|^3 \times \eta \times L(\eta, |S|))$, where $\eta = 4(\frac{|S|}{4})^{2^d}$ with $d = b_{\mathcal{M}}$.*

In practice it often holds that the branching of an *IMDP* is constant. As an evidence, all *MDP*s in PRISM [27] benchmarks have constant branching. Thus, our algorithm for computing $\sim_c$ will often terminate in polynomial time for practical models. It is worthwhile to mention that even in the case where $b_{\mathcal{M}}$ is a constant for an *IMDP* $\mathcal{M}$, its equivalent *MDP* can be exponentially larger than $\mathcal{M}$. Consequently, for such models our algorithm can avoid an exponential blow-up comparing to the bisimulation decision algorithm for *MDP*s in [4].

5 Concluding Remarks

In this paper, we have studied the probabilistic bisimulation problem for interval *MDP*s in order to speed up the run time of model checking algorithms that often suffer from the state space explosion. Interval MDPs include two sources of nondeterminism for which we have considered the cooperative resolution in a dynamic setting. We have extended the results in [16] by further investigating two core aspects of the defined bisimulation relation: compositional reasoning and complexity analysis of the decision algorithm. As regards the former, we have established a framework for compositional verification of complex systems with interval uncertainty. As regards the latter, we have shown that deciding probabilistic bisimulation for *IMDP*s is coNP-complete and also shown that tractability is guaranteed under a mild restriction on the class of *IMDP* models. There are various promising directions for future work. From modelling viewpoint, it is worthwhile to address more expressive formalisms to encode uncertainties (such as polynomial constraints or even parameters appearing in multiple states/actions). Moreover, from semantics viewpoint, it would be interesting to extend the current results for the competitive semantics of resolving nondeterminism. Finally, from algorithmic viewpoint, we conjecture that probabilistic bisimulation problem can be decided in polynomial time for *IMDP*s if a proper model of uncertainty is considered. More precisely, the probabilistic bisimulation

problem can be modeled as an instance of uncertain LPs based on the techniques in [17,35]. This class of uncertain LPs is computationally tractable [19], under some technical assumption regarding a proper model of uncertainty.

Acknowledgments. This work is supported by the EU 7th Framework Programme under grant agreements 295261 (MEALS) and 318490 (SENSATION), by the DFG as part of SFB/TR 14 AVACS, by the CAS/SAFEA International Partnership Program for Creative Research Teams, by the National Natural Science Foundation of China (Grants 61472473 and 61550110249), by the Chinese Academy of Sciences Fellowship for International Young Scientists (Grant 2015VTC029), and by the CDZ project CAP (GZ 1023). This research is supported in part by the National Science Foundation through Award CCF-1305054.

References

1. Balas, E.: Disjunctive programming: properties of the convex hull of feasible points. Discrete Appl. Math. **89**(1), 3–44 (1998)
2. Benedikt, M., Lenhardt, R., Worrell, J.: LTL model checking of interval Markov chains. In: Piterman, N., Smolka, S.A. (eds.) TACAS 2013 (ETAPS 2013). LNCS, vol. 7795, pp. 32–46. Springer, Heidelberg (2013)
3. Billingsley, P.: Probability and Measure. John Wiley and Sons, New York (1979)
4. Cattani, S., Segala, R.: Decision algorithms for probabilistic bisimulation. In: Brim, L., Jančar, P., Křetínský, M., Kučera, A. (eds.) CONCUR 2002. LNCS, vol. 2421, pp. 371–385. Springer, Heidelberg (2002)
5. Cattani, S., Segala, R., Kwiatkowska, M., Norman, G.: Stochastic transition systems for continuous state spaces and non-determinism. In: Sassone, V. (ed.) FOSSACS 2005. LNCS, vol. 3441, pp. 125–139. Springer, Heidelberg (2005)
6. Chatterjee, K., Sen, K., Henzinger, T.A.: Model-checking ω-regular properties of interval Markov chains. In: Amadio, R.M. (ed.) FOSSACS 2008. LNCS, vol. 4962, pp. 302–317. Springer, Heidelberg (2008)
7. Chehaibar, G., Garavel, H., Mounier, L., Tawbi, N., Zulian, F.: Specification and verification of the PowerScale® busarbitration protocol: an industrial experiment with LOTOS. In: FORTE, pp. 435–450 (1996)
8. Chen, T., Han, T., Kwiatkowska, M.: On the complexity of model checking interval-valued discrete time Markov chains. Inf. Process. Lett. **113**(7), 210–216 (2013)
9. Coste, N., Hermanns, H., Lantreibecq, E., Serwe, W.: Towards performance prediction of compositional models in industrial GALS designs. In: Bouajjani, A., Maler, O. (eds.) CAV 2009. LNCS, vol. 5643, pp. 204–218. Springer, Heidelberg (2009)
10. Delahaye, B., Katoen, J.-P., Larsen, K.G., Legay, A., Pedersen, M.L., Sher, F., Wąsowski, A.: Abstract probabilistic automata. In: Jhala, R., Schmidt, D. (eds.) VMCAI 2011. LNCS, vol. 6538, pp. 324–339. Springer, Heidelberg (2011)
11. Delahaye, B., Larsen, K.G., Legay, A., Pedersen, M.L., Wąsowski, A.: Decision problems for interval Markov chains. In: Dediu, A.-H., Inenaga, S., Martín-Vide, C. (eds.) LATA 2011. LNCS, vol. 6638, pp. 274–285. Springer, Heidelberg (2011)
12. Eirinakis, P., Ruggieri, S., Subramani, K., Wojciechowski, P.: On quantified linear implications. AMAI **71**(4), 301–325 (2014)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1990)

14. Hahn, E.M., Han, T., Zhang, L.: Synthesis for PCTL in parametric Markov decision processes. In: Bobaru, M., Havelund, K., Holzmann, G.J., Joshi, R. (eds.) NFM 2011. LNCS, vol. 6617, pp. 146–161. Springer, Heidelberg (2011)
15. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects Comput. **6**(5), 512–535 (1994)
16. Hashemi, V., Hatefi, H., Krčál, J.: Probabilistic bisimulations for PCTL model checking of interval MDPs. SynCoP EPTCS **145**, 19–33 (2014)
17. Hashemi, V., Hermanns, H., Turrini, A.: On the efficiency of deciding probabilistic automata weak bisimulation. ECEASST, 66 (2013)
18. Hermanns, H., Katoen, J.-P.: Automated compositional Markov chain generation for a plain-old telephone system. Sci. Comp. Progr. **36**(1), 97–127 (2000)
19. Iancu, D.A., Sharma, M., Sviridenko, M.: Supermodularity and affine policies in dynamic robust optimization. Oper. Res. **61**(4), 941–956 (2013)
20. Iyengar, G.N.: Robust dynamic programming. Math. Oper. Res. **30**(2), 257–280 (2005)
21. Jonsson, B., Larsen, K.G.: Specification and refinement of probabilistic processes. In: LICS, pp. 266–277 (1991)
22. Kanellakis, P.C., Smolka, S.A.: CCS expressions, finite state processes, and three problems of equivalence. Inf. Comput. **86**(1), 43–68 (1990)
23. Katoen, J.-P., Kemna, T., Zapreev, I., Jansen, D.N.: Bisimulation minimisation mostly speeds up probabilistic model checking. In: Grumberg, O., Huth, M. (eds.) TACAS 2007. LNCS, vol. 4424, pp. 87–101. Springer, Heidelberg (2007)
24. Katoen, J.-P., Klink, D., Neuhäußer, M.R.: Compositional abstraction for stochastic systems. In: Ouaknine, J., Vaandrager, F.W. (eds.) FORMATS 2009. LNCS, vol. 5813, pp. 195–211. Springer, Heidelberg (2009)
25. Kozine, I., Utkin, L.V.: Interval-valued finite Markov chains. Reliable Comput. **8**(2), 97–113 (2002)
26. Kroening, D., Strichman, O.: Decision Procedures: An Algorithmic Point of View, 1st edn. Springer, Heidelberg (2008)
27. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) CAV 2011. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011)
28. Nilim, A., Ghaoui, L.E.: Robust control of Markov decision processes with uncertain transition matrices. Oper. Res. **53**(5), 780–798 (2005)
29. Paige, R., Tarjan, R.E.: Three partition refinement algorithms. SIAM J. Comput. **16**(6), 973–989 (1987)
30. Puggelli, A., Li, W., Sangiovanni-Vincentelli, A.L., Seshia, S.A.: Polynomial-time verification of PCTL properties of MDPs with convex uncertainties. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 527–542. Springer, Heidelberg (2013)
31. Schrijver, A.: Theory of Linear and Integer Programming. John Wiley & Sons, Amsterdam (1998)
32. Segala, R.: Modeling and verification of randomized distributed real-time systems. Ph.D. thesis, MIT (1995)
33. Sen, K., Viswanathan, M., Agha, G.: Model-checking Markov chains in the presence of uncertainties. In: Hermanns, H., Palsberg, J. (eds.) TACAS 2006. LNCS, vol. 3920, pp. 394–410. Springer, Heidelberg (2006)
34. Subramani, K.: On the complexities of selected satisfiability and equivalence queries over boolean formulas and inclusion queries over hulls. JAMDS, 2009 (2009)
35. Turrini, A., Hermanns, H.: Polynomial time decision algorithms for probabilistic automata. Inf. Comput. **244**, 134–171 (2015)

36. Vaidya, P.M.: An algorithm for linear programming which requires $O(((m+n)n^2)+(m+n)^{1.5})n)L)$ arithmetic operations. Math. Program. **47**, 175–201 (1990)
37. Wolff, E.M., Topcu, U., Murray, R.M.: Robust control of uncertain Markov decision processes with temporal logic specifications. In: CDC, pp. 3372–3379 (2012)
38. Wu, D., Koutsoukos, X.D.: Reachability analysis of uncertain systems using bounded-parameter Markov decision processes. AI **172**(8–9), 945–954 (2008)
39. Yi, W.: Algebraic reasoning for real-time probabilistic processes with uncertain information. In: Langmaack, H., de Roever, W.-P., Vytopil, J. (eds.) FTRTFT 1994 and ProCoS 1994. LNCS, vol. 863, pp. 680–693. Springer, Heidelberg (1994)