CTL Model Checking of MDPs over Distribution Spaces: Algorithms and Sampling-based Computations

Yulong Gao
yulong.gao@imperial.ac.uk
Imperial College London
London, United Kingdom

Karl H. Johansson
kallej@kth.se
KTH Royal Institute of Technology
Stockholm, Sweden

Alessandro Abate
alessandro.abate@cs.ox.ac.uk
University of Oxford
Oxford, United Kingdom

ABSTRACT

This work studies computation tree logic (CTL) model checking for finite-state Markov decision processes (MDPs) over the space of their distributions. Instead of investigating properties over states of the MDP, as encoded by formulae in standard probabilistic CTL (PCTL), the focus of this work is on the associated transition system, which is induced by the MDP, and on its dynamics over the (transient) MDP distributions. CTL is thus used to specify properties over the space of distributions, and is shown to provide an alternative way to express probabilistic specifications or requirements over the given MDP. We discuss the distinctive semantics of CTL formulae over distribution spaces, compare them to existing non-branching logics that reason on probability distributions, and juxtapose them to traditional PCTL specifications. We then propose reachability-based CTL model checking algorithms over distribution spaces, as well as computationally tractable, sampling-based procedures for computing the relevant reachable sets: it is in particular shown that the satisfaction set of the CTL specification can be soundly under-approximated by the union of convex polytopes. Case studies display the scalability of these procedures to large MDPs.

KEYWORDS

Markov decision processes; transient probability distributions; computation tree logic; reachability analysis

ACM Reference Format:
Yulong Gao, Karl H. Johansson, and Alessandro Abate. 2024. CTL Model Checking of MDPs over Distribution Spaces: Algorithms and Sampling-based Computations. In *27th ACM International Conference on Hybrid Systems: Computation and Control (HSCC '24), May 14–16, 2024, Hong Kong SAR, China.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3641513.3651397

1 INTRODUCTION

Probabilistic logics offer a framework for specifying properties and/or requirements of stochastic models and probabilistic verification provides algorithms to calculate the likelihood of such specifications over Markov decision processes (MDPs), which are stochastic models that formalise decision making under uncertainty [25, 27, 29].

In this work, we study the use of computation tree logic (CTL) to express specifications for finite-state MDPs over their distribution space, and we correspondingly develop new CTL model checking algorithms. Unlike the standard body of work on verification of MDPs over specifications in probabilistic computation tree logic (PCTL), which concern requirements over states and trajectories of an MDP, we investigate the behavior of transient state distributions, which is governed by a discrete-time transition system evolving over the continuous space of the MDP distributions. This transition system is endowed with non-determinism, which is inherited from the actions of the MDP: this feature makes it natural to use CTL formulae to specify temporal properties and/or requirements over transient state distributions. Whilst admittedly less in use than state- or trajectory-specified temporal requirements, distribution-based specifications can express richer temporal properties of models of practical use, e.g., mobile robots in uncertain environment [24] and pharmacokinetics systems in [11, 26]. Recall that knowledge of transient distributions allows to fully characterise probability measures over product spaces, where temporal requirements, such as PCTL formulae, are defined [7]. It thus appears natural and meaningful to study distribution-specified requirements.

As a first contribution of this work, we show that the distribution-specified CTL formulae are semantically different from the traditional PCTL formulae, and thus can be employed to encode alternative probabilistic specifications. More specifically, the proposed CTL framework allows not only to express similar quantitative temporal logic specifications over the state space through marginalisations over state distributions, but also to encode other interesting requirements. In addition, in Section 3.3 we show that the new distribution-specified CTL formulae are largely different from the few existing logics that reason about probability distributions over MDPs, e.g., [17, 22]: indeed, these logics do not take into account the branching structures of the models, which instead can be naturally encoded by (universal and existential) quantifiers in CTL formulae (cf. also Section 1). Here, the use of CTL translates into broader objectives that comprise both model checking and policy synthesis.

The second major contribution of this work is the characterisation of satisfaction sets of CTL specifications over continuous distributions space. This is attained by a new algorithm that computes backward reachable sets for both existentially and universally quantified formulae. We should remark that the MDP model checking problem over continuous distribution spaces, including the CTL model checking problem in this work, is in general undecidable [5, 26], and additionally that the precise computation of backward reachable sets in continuous-space models is in general quite intractable. In order to practically and scalably compute such

HSCC '24, May 14–16, 2024, Hong Kong SAR, China

ACM ISBN 979-8-4007-0522-9/24/05
https://doi.org/10.1145/3641513.3651397

satisfaction sets, in this work we present new sampling-based algorithms, showing that (1) they compute under-approximations of the backward reachable sets, which are tight when the number of samples tends to infinity; and that (2) the satisfaction set of the CTL formula can be soundly under-approximated by the union of convex polytopes computed from the backward reachability.

2 RELATED WORK

Unlike LTL, CTL [12] formulae naturally account for non-determinism in the model of interest, for instance the presence of an external input, by using existential and universal quantification. The CTL model checking boils down to the recursive computation of satisfaction sets of sub-formulae, which relies on backward reachability analysis [13]: this is in general a difficult problem when models have continuous state spaces, as in our work. There are a few CTL model checkers, such as EMC [12] and SMV [34]: however, in this work we consider CTL model checking over continuous spaces (of distributions), which prevents the usability of these existing model checkers. It is remarkable that the control literature has so far *not* developed bespoke algorithms for CTL model checking over continuous-space models [10].

There is a large body of literature on probabilistic model checking, and we restrict our attention to PCTL model checking - see for example [27] for discrete-time Markov chains and [18] for MDPs. PCTL was first proposed in [20] as an extension of CTL, and its semantics involve (probabilistic) executions over the state space of an MDP. PCTL model checking classically reduces to the computation of maximal or minimal probabilities over sets of paths satisfying temporal requirements [7]. This can be solved via recursive equation for finite-horizon problem, or via solution to linear programming for infinite-horizon problem [18]. Notable software tools for probabilistic model checking are PRISM [28] and Storm [15].

Instead of considering probabilistic executions of an MDP over its state space, we study the MDP from the perspective of its transient probability distributions. Earlier work has introduced notions of distribution-based (bi)simulation relations for MDPs [17, 22]. In [9], a probabilistic logic was proposed to enable the comparison of the probability associated to different states of the model. Probability distributions have also been used to specify linear-time requirements: the iLTL for DTMCs in [30, 31], the LTL-like specifications for DTMCs in [2], the distribution-based ω-regular properties for MDPs in [11, 26], and the linear distribution temporal logic for partially observable MDPs (POMDPs) in [24]. Note that these linear-time properties are different from the distribution-specified CTL formulae in our work, which distinctively are in branching-time logic and thus entail different semantics. Comparisons amongst these probabilistic logics will be discussed in Section 3.3. A recent contribution [5] has considered the special case of existential and universal *safety* in the distributions space, which can be easily rewritten as CTL formulae in this work.

The decidability of model checking problems over distributions space has been studied in many works. It was proven in [3] that the reachability problem is as hard as the so-called SKOLEM problem and safety is as hard as the POSITIVITY problem, both of which boils down to a number-theoretical hardness result. In [2], an approximate model checking algorithm for the distributions dynamics of DTMCs under LTL-like specifications was developed. In [4], the authors developed an over-approximation approach to synthesize an affine invariant set in distributions space under safety constraint. The approaches put forward by these works however lack computational scalability: a key contribution of work is to provide sound, efficient, and more scalable computation algorithm for CTL model checking over probability distributions.

3 PRELIMINARIES AND PROBLEM STATEMENT

Notations. Let $\mathbb{N}$ denote the set of non-negative integers and $\mathbb{R}$ the set of real numbers. For some $q, s \in \mathbb{N}$ and $q < s$, let $\mathbb{N}_{[q,s]} = \{r \in \mathbb{N} \mid q \leq r \leq s\}$. For two sets $\mathbb{X}$ and $\mathbb{Y}$, $\mathbb{X} \oplus \mathbb{Y} = \{x+y \mid x \in \mathbb{X}, y \in \mathbb{Y}\}$. When $\leq$, $\geq$, $<$, and $>$ are applied to vectors, they are interpreted element-wise. Matrices of appropriate dimension with all elements equal to 1 and 0 are denoted by $\mathbf{1}$ and $\mathbf{0}$, respectively. The number of elements of a set $\mathbb{X}$ is denoted by $|\mathbb{X}|$.

3.1 Models - Markov Decision Processes (MDP)

Definition 1. *An MDP is a tuple* $\mathsf{M} = (\mathbb{X}, \mathbb{U}, T, \mathcal{AP}_s, L_s)$, *where*

- $\mathbb{X}$ *is a finite state space with cardinality* $|\mathbb{X}| = n$;
- $\mathbb{U}$ *is a finite action space with cardinality* $|\mathbb{U}| = m$;
- $T : \mathbb{X} \times \mathbb{X} \times \mathbb{U} \to \mathbb{R}$ *is a transition probability, i.e,* $T(y|x,u)$ *assigns a probability from the state* $x \in \mathbb{X}$ *and the action* $u \in \mathbb{U}$ *to the state* $y \in \mathbb{X}$;
- *a finite set* $\mathcal{AP}_s$ *of atomic propositions;*
- *a labelling function* $L_s : \mathbb{X} \to 2^{\mathcal{AP}_s}$.

For each $x \in \mathbb{X}$, $\mathbb{U}_x \subseteq \mathbb{U}$ is a nonempty set consisting of the admissible actions when the state of the MDP is x. For any $x \in \mathbb{X}$ and $u \in \mathbb{U}_x$, $\sum_{y \in \mathbb{X}} T(y|x,u) = 1$. Let us denote by $\mathcal{P}(\mathbb{X})$ the set of state distributions, which is a simplex in $\mathbb{R}^n$. A state distribution $\pi \in \mathcal{P}(\mathbb{X})$ can be seen as a non-negative row vector $\pi \in \mathbb{R}^n$, such that $\pi \mathbf{1} = 1$, where $\mathbf{1}$ is a column vector with all elements being 1. In this work, two kinds of policies are considered.

Definition 2 (Policies). *A policy is a map* $\mu : \mathbb{X} \to \mathcal{P}(\mathbb{U})$, *i.e., for each* $x \in \mathbb{X}$, $\sum_{u \in \mathbb{U}_x} \mu(u|x) = 1$, *where* $\mathcal{P}(\mathbb{U})$ *is the set of distributions over* $\mathbb{U}$. *Denote by* $\boldsymbol{\mathcal{U}}$ *this set of randomised policies. A deterministic policy is a map* $\mu^d : \mathbb{X} \to \mathbb{U}$, *i.e., for each* $x \in \mathbb{X}$, $\mu^d(x)$ *selects precisely one* u *from* $\mathbb{U}_x$. *Denote by* $\boldsymbol{\mathcal{U}}^d$ *the set of deterministic policies. Furthermore, let* $\boldsymbol{\mathcal{U}} = \{\boldsymbol{\mu} = \mu_0 \mu_1 \dots \mu_k \dots \mid \mu_k \in \boldsymbol{\mathcal{U}}, \forall k \in \mathbb{N}\}$ *be the set of time-dependent policy sequences.*

Note that, for an MDP M, the number of allowable deterministic policies is at most m^n, thus the set $\boldsymbol{\mathcal{U}}^d$ is finite. It follows that a general policy $\mu \in \boldsymbol{\mathcal{U}}$ can be interpreted as a distribution over $\boldsymbol{\mathcal{U}}^d$. Any policy $\mu \in \boldsymbol{\mathcal{U}}$ (and, in particular, any deterministic policy $\mu^d \in \boldsymbol{\mathcal{U}}^d$) induces from T a row-stochastic matrix as:

$$P^{\mu}(x'|x) = \sum_{u \in \mathbb{U}_x} T(x'|x,u)\mu(u|x). \tag{1}$$

Given an initial state distribution $\pi_0 \in \mathcal{P}(\mathbb{X})$ and a time-dependent policy sequence $\boldsymbol{\mu} = \{\mu_k \in \boldsymbol{\mathcal{U}}\}_{k \in \mathbb{N}} \in \boldsymbol{\mathcal{U}}$, the state distribution evolves over (a subset of) the Euclidean space $\mathbb{R}^n$ as

$$\pi_{k+1} = \pi_k P^{\mu_k} = \sum_{x \in \mathbb{X}} \pi_k(x) P^{\mu_k}(\cdot|x). \tag{2}$$

Definition 3 (State-Action Path). *For the MDP* M*, a* state-action path *starting from* $x_0 \in \mathbb{X}$ *is a sequence of states* $\boldsymbol{x} = x_0u_0x_1u_1 \dots x_ku_kx_{k+1}u_{k+1} \dots$*, with* $T(x_{k+1}|x_k,u_k) > 0$*. Given a time-dependent policy sequence* $\boldsymbol{\mu} = \{\mu_k \in \mathcal{U}\}_{k\in\mathbb{N}} \in \boldsymbol{\mathcal{U}}$*, denote by* $\mathsf{SPath}(x_0,\boldsymbol{\mu})$ *the set of state-action paths* $\boldsymbol{x} = x_0u_0x_1u_1 \dots x_ku_kx_{k+1}u_{k+1} \dots$ *with* $T(x_{k+1}|x_k,u_k) > 0$ *and* $\mu_k(u_k|x_k) > 0$*. Denote by* $\mathsf{SPath}(x_0)$ *the set of all the state-action paths starting from* x_0 *under all possible* $\boldsymbol{\mu} \in \boldsymbol{\mathcal{U}}$*, i.e.,* $\mathsf{SPath}(x_0) = \bigcup_{\boldsymbol{\mu}\in\boldsymbol{\mathcal{U}}} \mathsf{SPath}(x_0,\boldsymbol{\mu})$.

As much as the MDP dynamics over its states (namely, its state-action paths) are governed by the transition probability matrix, the state distribution of an MDP follows the action-dependent dynamics in (2). We can thus equivalently look at the MDP dynamics from the perspective of the following MDP-induced transition system, which evolves over the space of distributions, as follows.

Definition 4. *Given the MDP* $\mathsf{M} = (\mathbb{X},\mathbb{U},T,\mathcal{AP}_s,L_s)$*, the MDP-induced transition system* MTS *is a tuple* $\mathsf{MTS} = (\mathcal{P}(\mathbb{X}),\boldsymbol{\mathcal{U}},\rightarrow,\mathcal{AP}_d,L_d)$*, consisting of*

- *a space* $\mathcal{P}(\mathbb{X})$ *of distributions over states, namely a subset of* $\mathbb{R}^n$*;*
- *a set* $\boldsymbol{\mathcal{U}}$ *of policies for* M*;*
- *a transition relation* $\rightarrow \in \mathcal{P}(\mathbb{X}) \times \boldsymbol{\mathcal{U}} \times \mathcal{P}(\mathbb{X})$*, i.e., for* $\pi,\pi' \in \mathcal{P}(\mathbb{X})$ *and* $\mu \in \boldsymbol{\mathcal{U}}$*,* $\pi \xrightarrow{\mu} \pi'$ *if and only if* $\pi' = \pi P^{\mu}$*;*
- *a finite set* $\mathcal{AP}_d$ *of atomic propositions;*
- *a labeling function* $L_d : \mathcal{P}(\mathbb{X}) \to 2^{\mathcal{AP}_d}$.

The MTS model is a (non-deterministic) dynamical system evolving over an uncountably-infinite state space, namely a subset (the unit simplex) of $\mathbb{R}^n$, and with dynamics that are governed by the difference equation in (2). Note that the labeling function L_d and the associated atomic proposition set $\mathcal{AP}_d$ are in general different from L_s and $\mathcal{AP}_s$ in the MDP M. In Section 3.3, we show that if L_d and $\mathcal{AP}_d$ are appropriately defined based on L_s and $\mathcal{AP}_s$, we can express related properties for M and MTS, respectively.

Definition 5 (Distribution-Policy Path). *For the MDP-induced transition system* MTS*, a* distribution-policy path $\boldsymbol{\pi}$ *starting from* $\pi_0 \in \mathcal{P}(\mathbb{X})$ *is a sequence of state distributions* $\boldsymbol{\pi} = \pi_0\mu_0\pi_1\mu_1 \dots \pi_k\mu_k \dots$ *such that* $\forall k \in \mathbb{N}$, $\pi_k \xrightarrow{\mu_k} \pi_{k+1}$*. Denote by* $\mathsf{DPath}(\pi_0)$ *the set of distribution-policy paths starting from* π_0.

3.2 Specifications - (Probabilistic) Computation Tree Logic

CTL comprises state and path formulae defined over a general alphabet $\mathcal{AP}$, and encompasses both propositional and temporal logic operators. CTL *state formulae* are formed by

$$\Phi ::= \text{true} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \exists\varphi \mid \forall\varphi,$$

where $a \in \mathcal{AP}$ is an atomic proposition and φ is a path formula. *Path formulae* are instead shaped according to the following rules:

$$\varphi ::= \bigcirc\Phi \mid \Phi_1 \mathsf{U} \Phi_2,$$

where $\bigcirc$ and U denote the "next" and "until" operators, respectively, and Φ, Φ_1, and Φ_2 are state formulae. We can introduce more complex formulae, amongst which we shall use in this work (1) disjunction, $\Phi_1 \vee \Phi_2 = \neg(\neg\Phi_1 \wedge \neg\Phi_2)$; (2) eventually, $\exists\Diamond\Phi = \exists(\text{true}\mathsf{U}\Phi)$ and $\forall\Diamond\Phi = \forall(\text{true} \cup \Phi)$; and (3) always, $\exists\Box\Phi = \neg\forall\Diamond\neg\Phi$ and $\forall\Box\Phi = \neg\exists\Diamond\neg\Phi$.

PCTL formulae are similarly defined over a finite alphabet $\mathcal{AP}$: state formulae Φ are much like CTL, except that in CTL they are quantified either existentially or universally (that is, $\exists\varphi$ and $\forall\varphi$), whereas PCTL formulae now depend on a probabilistic operator, as

$$\Phi ::= [\dots] \mid \mathsf{Pr}_{\sim p}(\varphi),$$

where φ is a path formula and Pr denotes the probabilistic operator, $\sim \in \{>,<,\geq,\leq\}$, and $p \in [0,1]$. Path formulae φ are defined exactly as in CTL, with the addition of the following bounded-time operator

$$\varphi ::= [\dots] \mid \Phi_1 \mathsf{U}^{\leq k} \Phi_2,$$

where now $k \in \mathbb{N}$ is a finite index. The bounded-until operator $\mathsf{U}^{\leq k}$, as natively used in PCTL formulae [7], can be naturally expressed also in CTL by evaluating their semantics over finite paths [38] (see below).

We now tailor the (P)CTL syntax above to the models of interest, namely the MDP M and the MDP-induced transition system MTS, by discussing the associated satisfaction semantics [7]. Notice that the generic alphabet $\mathcal{AP}$ will be tailored to the labelling maps introduced in the definitions of the models above (respectively $\mathcal{AP}_s$ and $\mathcal{AP}_d$), and that the semantics will hinge on state-action paths for M and on distribution-policy paths for MTS, respectively.

Definition 6 (PCTL Semantics). *Consider the MDP* M*. Given an atomic proposition* $a \in \mathcal{AP}_s$*, a state* $x \in \mathbb{X}$*, PCTL state formulae* Φ*,* Φ_1*, and* Φ_2*, and a path formula* φ*, the satisfaction relation* $\vDash$ *is defined for state formulae by*

$$x \vDash a \Leftrightarrow a \in L_s(x);\ x \vDash \neg\Phi \Leftrightarrow x \nvDash \Phi;$$
$$x \vDash \Phi_1 \wedge \Phi_2 \Leftrightarrow x \vDash \Phi_1 \wedge x \vDash \Phi_2;$$
$$x \vDash \mathsf{Pr}_{\sim p}(\varphi) \Leftrightarrow \mathsf{Pr}\big(\boldsymbol{x} \in \mathsf{SPath}(x) \mid \boldsymbol{x} \vDash \varphi\big) \sim p \Leftrightarrow \mathsf{Pr}\big(\boldsymbol{x} \in \mathsf{SPath}(x,\boldsymbol{\mu}) \mid \boldsymbol{x} \vDash \varphi\big) \sim p, \forall \boldsymbol{\mu} \in \boldsymbol{\mathcal{U}}.$$

Given a state-action path $\boldsymbol{x} = x_0u_0x_1u_1 \dots x_ku_k \dots$*, the satisfaction relation* $\vDash$ *is defined for path formulae by*

$$\boldsymbol{x} \vDash \bigcirc\Phi \Leftrightarrow x_1 \vDash \Phi;\ \boldsymbol{x} \vDash \Phi_1\mathsf{U}^{\leq k}\Phi_2 \Leftrightarrow \begin{cases} \exists j \in \mathbb{N}_{[0,k]}\ s.t.\ x_j \vDash \Phi_2, \\ \forall i \in \mathbb{N}_{[0,j-1]}, x_i \vDash \Phi_1; \end{cases}$$

$$\boldsymbol{x} \vDash \Phi_1\mathsf{U}\Phi_2 \Leftrightarrow \begin{cases} \exists j \in \mathbb{N}\ s.t.\ x_j \vDash \Phi_2, \\ \forall i \in \mathbb{N}_{[0,j-1]}, x_i \vDash \Phi_1. \end{cases}$$

The semantics of $\mathsf{Pr}_{\sim p}(\varphi)$ can be further tailored to a single policy sequence $\boldsymbol{\mu} \in \boldsymbol{\mathcal{U}}$. Let us introduce the operator $\mathsf{Pr}^{\boldsymbol{\mu}}_{\sim p}(\varphi)$ and define its semantics as:

$$x \vDash \mathsf{Pr}^{\boldsymbol{\mu}}_{\sim p}(\varphi) \Leftrightarrow \mathsf{Pr}\big(\boldsymbol{x} \in \mathsf{SPath}(x,\boldsymbol{\mu}) \mid \boldsymbol{x} \vDash \varphi\big) \sim p. \tag{3}$$

CTL formulae are instead adapted to MTS as follows.

Definition 7 (CTL semantics). *Consider the MDP-induced transition system* MTS*, a state distribution* $\pi \in \mathcal{P}(\mathbb{X})$*, an atomic proposition* $a \in \mathcal{AP}_d$*, CTL state formulae* Φ*,* Φ_1*, and* Φ_2*, and a path formula* φ*. The satisfaction relation* $\vDash$ *is defined for CTL state formulae by*

$$\pi \vDash \exists\varphi \Leftrightarrow \boldsymbol{\pi} \vDash \varphi \text{ for some } \boldsymbol{\pi} \in \mathsf{DPath}(\pi);$$
$$\pi \vDash \forall\varphi \Leftrightarrow \boldsymbol{\pi} \vDash \varphi \text{ for all } \boldsymbol{\pi} \in \mathsf{DPath}(\pi).$$

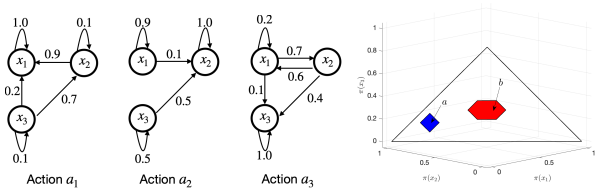


Figure 1: Example 2: (i) Graphical representation of the MDP; (ii) Two sets (in blue and red) of state distributions, with labels a and b, respectively.

Given a distribution-policy path $\boldsymbol{\pi} = \pi_0\mu_0\pi_1\mu_1 \ldots \pi_k\mu_k \ldots$, the satisfaction relation $\models$ is defined for path formulae by

$$\boldsymbol{\pi} \models \bigcirc\Phi \Leftrightarrow \pi_1 \models \Phi;\ \boldsymbol{\pi} \models \Phi_1 \mathsf{U} \Phi_2 \Leftrightarrow \begin{cases} \exists j \in \mathbb{N}\ s.t.\ \pi_j \models \Phi_2, \\ \forall i \in \mathbb{N}_{[0,j-1]}, \pi_i \models \Phi_1. \end{cases}$$

As we commented earlier the bounded-until operator can be also expressed in CTL as intuitive [38].

The verification problem for MDPs (or for corresponding MTSs) denotes the computation of sets of states (resp. of distributions) of the model that satisfy a given (PCTL and CTL, resp.) temporal logic formula. For instance, let us denote by $\mathrm{Sat}(\Phi) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \pi \models \Phi\}$ the *satisfaction set* of a given CTL formula Φ. It should be emphasised that the verification question for *existentially-quantified* CTL formulae elicits a synthesis problem, namely we are interested in generating a policy associated to satisfying traces: the CTL verification algorithms we put forward in this work do produce such a policy. Unlike CTL, which reasons equally with universally- and existentially-quantified formulae, the classical PCTL *model checking problem* (cf. Definition 6, top) considers probabilities under all allowable paths and policies; alternatively, a PCTL *synthesis problem* (cf. Eq.(3)) is formulated as follows: given an initial state x_0, find a policy sequence $\boldsymbol{\mu} \in \boldsymbol{\mathcal{U}}$ such that $x_0 \models \Pr^{\boldsymbol{\mu}}_{\sim p}(\varphi)$.

Remark 1. *The use of distributions to specify probabilistic properties has been explored in many applications: the PageRank algorithm in [2], mobile robots in uncertain environment [24], and a pharmacokinetics system in [11, 26], and M/M/1 queueing system in [30, 31]. We remark that these properties can be naturally expressed by distribution-specified CTL formulae to be studied in this work. Next, we present two instances of distribution-specified CTL formulae.*

Example 1. ***Drug Injection Synthesis for a Pharmacokinetics System.*** *We consider the MDP model of a pharmacokinetics system adapted from [11, 26], which consists of five states: plasma (Pl), interstitial fluid (IF), utilisation and degradation (Ut), drug being injected (Dr), the drug being cleared (Cl), and "dummy" state (Re) (which allows to adjust the amount of drug being initially injected). The transition matrices of this model are detailed in Appendix B. The initial distribution is defined by $\pi_0(Dr) = \alpha$, $\pi_0(Re) = 1 - \alpha$, and $\pi_0(x) = 0$ for $x \in \{Pl, IF, CI, Ut\}$, where α is the amount of drug being initially injected. Following [11, 26], we set thresholds $MEC = 0.13$ and $MTC = 0.20$, and consider the set of atomic propositions $\mathcal{AP}_d = \{\texttt{effective}, \texttt{nontoxic}, \texttt{cleared}\}$. Considering $\pi_k(Ut)$, namely the probability of the drug being in state Ut at time k, we define label* `effective` *as $\pi_k(Ut) \geq MEC$;* `nontoxic` *as $\pi_k(Ut) \geq MTC$; and* `cleared` *as $\pi_k(Cl) \leq \epsilon$ for some given (small) value $\epsilon > 0$. The distribution-specified CTL formula of interest is $\Phi = \Phi_1 \wedge \Phi_2 \wedge \Phi_3$, with $\Phi_1 = \forall\Box\texttt{nontoxic}$, $\Phi_2 = \forall\Diamond(\texttt{effective} \wedge \forall \bigcirc \texttt{effective})$, $\Phi_3 = \forall\Diamond\texttt{cleared}$. Here, Φ_1 encodes the requirement that the drug level always stays in the safe zone; Φ_2 stipulates the drug is eventually effective for at least two consecutive steps; and Φ_3 specifies that the drug ought to be eventually cleared.*

In [11, 26], similar specifications were expressed as ω-regular expressions and shown to be decidable over the MDP. However, these two works did not provide any model checking algorithm. In particular, the computation of feasible values of α was not explored in [11, 26]: this problem can now be tackled by our CTL model checking algorithms, see Section 6.

Example 2. ***Opinion Consensus of Multi-agent Systems.*** *Inspired by [14], consider three agents $\{x_1, x_2, x_3\}$ whose beliefs over a subject (e.g., a economic or social variable) change in time as a consequence of pairwise and asymmetric interactions over a network. The interactions between agents can be 'controlled' through different incentives, which are denoted by $\mathbb{U} = \{a_1, a_2, a_3\}$. This system can be modeled by an MDP, whose corresponding transition probability matrices displayed in Fig. 1(i). We are interested in the set of beliefs from which there exists a policy such that the agents' beliefs approach consensus (i.e., $[1/3\ 1/3\ 1/3]$), whilst some other beliefs (i.e., the set of state distributions) are not visited. More specifically, let us introduce two (polytopic) sets of state distributions, as shown in blue and red in Fig. 1(ii). The set in red is $\{\pi \in \mathcal{P}(\mathbb{X}) \mid \|\pi - [1/3\ 1/3\ 1/3]\|_\infty \leq 0.1\}$, which corresponds to the consensus property, and the set in blue is $\{\pi \in \mathcal{P}(\mathbb{X}) \mid \|\pi - [0.1\ 0.2\ 0.7]\|_\infty \leq 0.05\}$, which is the set of unexpected beliefs. Here $\|\cdot\|_\infty$ denotes the infinity-norm. The label function L_d maps the distributions in the blue region to the atomic proposition a and those in the red region to b. Then, the property of interest can be expressed as a distribution-specified CTL formula $\Phi = \exists(\neg a \mathsf{U} b)$, whose model checking result will be provided in Example 3.*

3.3 Statement of the Problems under Study

The two discussed perspectives, the classical one on the MDP M and the alternative one on the transition model MTS, allow to introduce two different classes of probabilistic temporal requirements: one is by the widely-used PCTL, while another is new and relies on CTL formulae defined over state distributions. A natural question is how to relate these two classes of specifications. In addition, it is of interest to identify the differences between the proposed distribution-specified CTL formulae and other logics that reason over distributions [9, 11, 26, 30]. Thus, we posit the following first problem.

Problem 1. *Given an MDP M and the MDP-induced transition system* MTS*, formally relate the semantics of CTL formulae for* MTS *with that of probabilistic logics, e.g., the PCTL specifications for M.*

The second goal is to study properties over the original MDP M by proposing new CTL model checking algorithms for the MTS.

Problem 2. *Given an MDP M with initial distribution $\pi_0 \in \mathcal{P}(\mathbb{X})$ and the MDP-induced transition system* MTS*, and a CTL formula Φ, verify whether $\pi_0 \models \Phi$.*

Note that, as CTL formulae in this work are defined over a continuous space of distributions, the above verification problem is broadly undecidable [5]. More generally, the control literature has so far *not* developed bespoke algorithms for CTL model checking over continuous-space models [10]. We believe that it is of great interest to re-develop the standard CTL model checking algorithms for finite-state transition systems to MTS models in continuous spaces. A key contribution of this work is to develop a sound, efficient, and scalable computation algorithm for solving the Problem 2.

4 DISTRIBUTION-SPECIFIED CTL VERSUS PROBABILISTIC LOGICS

We begin with a comparison with other distribution-based logics. A first key difference is that existing logics, including the linear temporal logic (LTL)-like in [2], the linear inequality LTL (iLTL) in [30], and the ω-regular properties in [26, 31], are all focused on linear-time requirements, while the CTL used in this work is a branching-time logic. Thus, in addition to be incomparable in expressivity [7], it is CTL that allows to explicitly handle non-determinism in models for model-checking and synthesis. We show that in our work the verification of existentially-quantified CTL formulae elicits a synthesis problem, that is, to compute a feasible policy resulting in satisfying paths. Let us also remark that the central question in these related works is the decidability of model-checking problems, not the derivation of model-checking algorithms. Instead, our work provides sound and computationally tractable algorithms for CTL model checking over (continuous) distribution spaces. Finally, the logic over probabilities proposed in [9] is incomparable with the CTL formulae in our work: there, the existential quantifier is incorporated to specify the individual variable (e.g., time), which is quite different from the handling of model non-determinism by an existential quantifier in CTL formulae. Other differences can be derived from the discussion in [26].

Next, we compare the semantics of CTL formulae over the MDP distributions space with that of PCTL ones over its state space, as provided in Section 2. We first remark that these two sets of formulae are in general not comparable. In particular, there exist distribution-specified CTL formulae for which no corresponding PCTL formulae exist. For example, assume that there exists an atomic proposition a_d such that $L_d^{-1}(a_d)$ is *a subset of the interior* of the distribution space $\mathcal{P}(\mathbb{X})$. The meaningful CTL formulae $\exists \bigcirc a_d$, $\forall \bigcirc a_d$, $\exists \Diamond a_d$, and $\forall \Diamond a_d$ are such that any satisfying distribution in $L_d^{-1}(a_d)$ has a domain corresponding to the whole state space $\mathbb{X}$, for which corresponding PCTL requirements on states of the MDP are vacuous.

Despite their clear differences, we observe that there are semantical connections between distribution-specified CTL formulae and standard PCTL specifications over MDPs. We start with reachability and safety properties, expanding the discussion in Appendix A. The following proposition shows that the satisfaction of CTL reachability properties over distributions space provides a *sufficient* condition to verify PCTL reachability specifications over the state space; dually, the satisfaction of CTL safety properties over the distribution space is a *necessary* condition to verify PCTL safety specifications over the state space.

Proposition 1. *Consider the MDP* $\mathsf{M} = (\mathbb{X}, \mathbb{U}, T, \mathcal{AP}_s, L_s)$ *and two PCTL formulae* $\mathsf{Pr}_{\geq p}(\Phi_1 \mathsf{U}\, \Phi_2)$ *and* $\mathsf{Pr}_{\geq p}(\Box\Phi)$*, where* Φ_1*,* Φ_2*, and* Φ *are PCTL state formulae, and* $p \in [0,1]$*. Let the MDP-induced transition system be* $\mathsf{MTS} = (\mathcal{P}(\mathbb{X}), \boldsymbol{\mathcal{U}}, \rightarrow, \mathcal{AP}_d, L_d)$*, where* $\mathcal{AP}_d = \{a_{d1}, a_{d2}, a_d\}$*, and*

$$\begin{cases} L_d^{-1}(a_{d1}) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \sum_{x \in \mathrm{Sat}(\Phi_1)} \pi(x) = 1\}, \\ L_d^{-1}(a_{d2}) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \sum_{x \in \mathrm{Sat}(\Phi_2)} \pi(x) \geq p\}, \\ L_d^{-1}(a_d) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \sum_{x \in \mathrm{Sat}(\Phi)} \pi(x) \geq p\}, \end{cases}$$

where $\mathrm{Sat}(\cdot)$ *denotes the satisfaction set over the state space. Then,*

(1) *if* $e_{x_0} \vDash \forall(a_{d1} \mathsf{U} a_{d2})$*, then* $x_0 \vDash \mathsf{Pr}_{\geq p}(\Phi_1 \mathsf{U} \Phi_2)$*;*
(2) *if* $x_0 \vDash \mathsf{Pr}_{\geq p}(\Box\Phi)$*, then* $e_{x_0} \vDash \forall\Box a_d$*.*

Here e_{x_0} *is a vector with the* x_0*-th element equal to* 1 *and all the others set to* 0*.*

Proof. We first prove the statement (1). Recall that $e_{x_0} \vDash \forall(a_{d1} \mathsf{U} a_{d2})$ implies that for any distribution-policy path $\boldsymbol{\pi} = e_{x_0}\mu_0\pi_1\mu_1 \ldots \pi_k\mu_k \ldots$, there exists $j \in \mathbb{N}$ such that $\pi_j \in L_d^{-1}(a_{d2})$ and for all $i \in \mathbb{N}_{[0,j-1]}$, $\pi_i \in L_d^{-1}(a_{d1})$. For any $\boldsymbol{\mu} = \{\mu_k\}_{k \in \mathbb{N}}$ and $\boldsymbol{\pi} = e_{x_0}\mu_0\pi_1\mu_1 \ldots \pi_k\mu_k \ldots$, let $\mathrm{SPath}(x_0, \boldsymbol{\mu})$ be the set of state-action paths $\boldsymbol{x} = x_0u_0x_1u_1 \ldots x_ku_kx_{k+1}u_{k+1} \ldots$ with $T(x_{k+1}|x_k, u_k) > 0$ and $\mu_k(u_k|x_k) > 0$. Given the definitions of a_{d1} and a_{d2}, we have that $\mathsf{Pr}\big(\boldsymbol{x} \in \mathrm{SPath}(x_0, \boldsymbol{\mu}) | \boldsymbol{x} \vDash \Phi_1 \mathsf{U} \Phi_2\big) \geq \mathsf{Pr}\big(\boldsymbol{x} \in \mathrm{SPath}(x_0, \boldsymbol{\mu}) | e_{x_0} \vDash \forall(a_{d_1} \mathsf{U} a_{d_2})\big) \geq p$ for all $\boldsymbol{\mu} \in \boldsymbol{\mathcal{U}}$. Thus, $x_0 \vDash \mathsf{Pr}_{\geq p}(\Phi_1 \mathsf{U} \Phi_2)$ if $e_{x_0} \vDash \forall(a_{d_1} \mathsf{U} a_{d_2})$.

For the statement (2), if $x_0 \vDash \mathsf{Pr}_{\geq p}(\Box\Phi)$, we have $\mathsf{Pr}\big(\boldsymbol{x} \in \mathrm{SPath}(x_0, \boldsymbol{\mu}) | \boldsymbol{x} \vDash \Box\Phi)\big) = \mathsf{Pr}\big(\boldsymbol{x} \in \mathrm{SPath}(x_0, \boldsymbol{\mu}) | x_k \in \mathrm{Sat}(\Phi), \forall k \in \mathbb{N}\big) \geq p$. This further implies that the state x_k at each time step k stays in the set $\mathrm{Sat}(\Phi)$ with probability no less than p, which gives the definition of a_d and yields that $e_{x_0} \vDash \forall\Box a_d$. □

Selecting $p = 1$ and $\Phi_1 = \mathrm{true}$ in (1) produces a well known result in probabilistic model checking [7, Ch. 10.2.2] concerning almost-sure reachability. Now recall the semantics of policy-dependent PCTL formulae $\mathsf{Pr}^{\boldsymbol{\mu}}_{\sim p}(\varphi)$ in Eq. (3) and the PCTL synthesis problem in Section 3.2. The following corollary further provides a connection between policy synthesis over PCTL and model checking over distribution-specified CTL, thanks to the existential quantifier.

Corollary 1. *Consider the MDP* M *and the MDP-induced transition system* MTS*, as in Proposition 1. The following statements hold:*

(1) *if* $e_{x_0} \vDash \exists(a_{d1} \mathsf{U} a_{d2})$*, then there exists* $\boldsymbol{\mu} \in \boldsymbol{\mathcal{U}}$ *such that* $x_0 \vDash \mathsf{Pr}^{\boldsymbol{\mu}}_{\geq p}(\Phi_1 \mathsf{U}\, \Phi_2)$*;*
(2) *if* $x_0 \vDash \mathsf{Pr}^{\boldsymbol{\mu}}_{\geq p}(\Box\Phi)$ *for some* $\boldsymbol{\mu} \in \boldsymbol{\mathcal{U}}$*, then* $e_{x_0} \vDash \exists\Box a_d$*.*

It has been shown in [19, 36] that the safety property $\mathsf{Pr}_{\geq p}(\Box\Phi)$ holds for $p \in (0,1]$ only if the set $\mathrm{Sat}(\Phi)$ contains a bottom strongly connected component, which raises a strong assumption on the MDP M. The necessary condition in Proposition 1 suggests that this strong assumption can be relaxed when specifying safety properties in the distributions space. One can interpret this relaxation as follows: $x_0 \vDash \mathsf{Pr}_{\geq p}(\Box\Phi)$ specifies the property that *infinite-horizon* paths stay in $\mathrm{Sat}(\Phi)$ with probability no less than p, while $\forall\Box a_d$ specifies the property that state-action paths *at each time step* stay in $\mathrm{Sat}(\Phi)$ with probability no less than p. Similar arguments hold for the result in Corollary 1, as further discussed in the UAV path planning in Appendix C.

Finally, we add a note on the outcome of model checking algorithms for state-based PCTL and distribution-based CTL specifications, respectively. With reference to the satisfaction semantics in Definitions 6 and 7, we highlight that CTL model checking over the distribution space returns an informative satisfaction set that is a subset of the distributions space of the MDP. This is unlike PCTL model checking, which instead results in a satisfaction set that is a subset of the state space. As such, the novel CTL model checking algorithms in this paper leverage reachability analysis over distribution spaces: we detail this problem in the next section.

5 CTL MODEL CHECKING OVER DISTRIBUTION SPACES

This section will provide a reachability-based solution to Problem 2, i.e., a characterisation of CTL model checking for MDPs over their distribution spaces. We begin by defining two backward-reachability operators with respect to existential and universal quantifiers in CTL formulae, respectively. We then adapt the standard CTL model checking algorithm based on reachability analysis for finite transition systems to one for MDP-induced transition systems, which are instead endowed with a continuous state space. In view of this key feature, the computation of reach sets is in general intractable. We finally provide an approximate method for the numerical computations associated to the new algorithm.

5.1 Reachability-based CTL Model Checking

Consider the MDP $\mathsf{M} = (\mathbb{X}, \mathbb{U}, T, \mathcal{AP}_s, L_s)$ and the corresponding MTS $= (\mathcal{P}(\mathbb{X}), \mathcal{U}, \rightarrow, \mathcal{AP}_d, L_d)$. Define two set-valued maps $\mathcal{BR}_\exists : 2^{\mathcal{P}(\mathbb{X})} \rightarrow 2^{\mathcal{P}(\mathbb{X})}$ and $\mathcal{BR}_\forall : 2^{\mathcal{P}(\mathbb{X})} \rightarrow 2^{\mathcal{P}(\mathbb{X})}$ as

$$\mathcal{BR}_\exists(\Pi) = \left\{\pi \in \mathcal{P}(\mathbb{X}) \;\middle|\; \exists \mu \in \mathcal{U}, \pi P^\mu \in \Pi\right\}, \tag{4}$$

$$\mathcal{BR}_\forall(\Pi) = \left\{\pi \in \mathcal{P}(\mathbb{X}) \;\middle|\; \forall \mu \in \mathcal{U}, \pi P^\mu \in \Pi\right\}, \tag{5}$$

where $\Pi \subseteq \mathcal{P}(\mathbb{X})$. The set $\mathcal{BR}_\exists(\Pi)$ collects all the state distributions that can be steered to set Π under some policy $\mu \in \mathcal{U}$, whereas the set $\mathcal{BR}_\forall(\Pi)$ collects all the state distributions that can be steered to set Π under all possible policies $\mu \in \mathcal{U}$. Let us introduce the Post Set, denoted as $\mathrm{Post}(\pi)$, comprising the direct successors of $\pi \in \mathcal{P}(\mathbb{X})$: this is defined by $\mathrm{Post}(\pi) = \{\pi' \in \mathcal{P}(\mathbb{X}) \mid \exists \mu \in \mathcal{U}, \pi \xrightarrow{\mu} \pi'\}$. The above two maps $\mathcal{BR}_\exists$ and $\mathcal{BR}_\forall$ can then be rewritten, respectively, as

$$\mathcal{BR}_\exists(\Pi) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \mathrm{Post}(\pi) \cap \Pi \neq \emptyset\},$$
$$\mathcal{BR}_\forall(\Pi) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \mathrm{Post}(\pi) \subseteq \Pi\}.$$

Based on the maps defined above and leveraging the standard CTL model checking algorithms for finite-state models [7], we now introduce a new CTL model checking for MDPs over distribution spaces. Recall that $\mathrm{Sat}(\Phi) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \pi \vDash \Phi\}$ is the satisfaction set of a given CTL state formula Φ. Consider an atomic proposition $a \in \mathcal{AP}_d$ and three CTL state formulae Φ, Φ_1, and Φ_2 in PNF. For the propositional fragment of CTL formulae, it is straightforward to see that:

- $\mathrm{Sat}(\mathrm{true}) = \mathcal{P}(\mathbb{X})$ and $\mathrm{Sat}(\mathrm{false}) = \emptyset$;
- $\mathrm{Sat}(a) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid a \in L_d(\pi)\}$
- $\mathrm{Sat}(\neg\Phi) = \mathcal{P}(\mathbb{X}) \setminus \mathrm{Sat}(\Phi)$;
- $\mathrm{Sat}(\Phi_1 \wedge \Phi_2) = \mathrm{Sat}(\Phi_1) \cap \mathrm{Sat}(\Phi_2)$ and $\mathrm{Sat}(\Phi_1 \vee \Phi_2) = \mathrm{Sat}(\Phi_1) \cup \mathrm{Sat}(\Phi_2)$.

From the one-step operators introduced above, it follows that

- $\mathrm{Sat}(\exists \bigcirc \Phi) = \mathcal{BR}_\exists(\mathrm{Sat}(\Phi))$; $\mathrm{Sat}(\forall \bigcirc \Phi) = \mathcal{BR}_\forall(\mathrm{Sat}(\Phi))$.

The satisfaction set for until operator leverages on the iterative computation of backward reachable sets [35]. That is, The following statements hold:

$$\mathrm{Sat}(\exists(\Phi_1 \mathrm{U} \Phi_2)) = \mathbb{T}_\infty, \quad \mathrm{Sat}(\forall(\Phi_1 \mathrm{U} \Phi_2)) = \mathbb{S}_\infty,$$

where $\mathbb{T}_\infty = \lim_{i \rightarrow \infty} \mathbb{T}_i = \bigcup_{i \in \mathbb{N}} \mathbb{T}_i$, $\mathbb{T}_0 = \mathrm{Sat}(\Phi_2)$, $\mathbb{T}_{i+1} = \mathbb{T}_i \cup (\mathrm{Sat}(\Phi_1) \cap \mathcal{BR}_\exists(\mathbb{T}_i))$, $\mathbb{S}_\infty = \lim_{i \rightarrow \infty} \mathbb{S}_i = \bigcup_{i \in \mathbb{N}} \mathbb{S}_i$, $\mathbb{S}_0 = \mathrm{Sat}(\Phi_2)$, and $\mathbb{S}_{i+1} = \mathbb{S}_i \cup (\mathrm{Sat}(\Phi_1) \cap \mathcal{BR}_\forall(\mathbb{S}_i))$. Here the set convergence follows from the Monotone convergence theorem.

To summarise, CTL model checking for MDPs over distribution spaces reduces to the computation of backward reachable sets $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$ from a given set $\Pi \subset \mathcal{P}(\mathbb{X})$. Let us emphasise that, for general sets Π (in particular, dense and non-convex), it can be computational intractable to manipulate $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$. Next, we will develop a sampling-based method for facilitating their numerical computation.

5.2 Sampling-based Algorithm to Compute Backward Reachable Sets

We restrict our attention to the problem of computing backward reachable sets $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$ whenever $\Pi \subseteq \mathcal{P}(\mathbb{X})$ is a convex polytope. According to the Minkowshi-Weyl's Thorem, any convex polytope $\mathbb{Y} \subset \mathbb{R}^n$ can be expressed in a (vertex) V-representation, i.e., $\mathbb{Y} = \mathrm{conv}(\{v_1, \ldots, v_N\}) = \{z = \sum_{i=1}^N \lambda_i v_i \mid \sum_{i=1}^N \lambda_i = 1, \lambda_i \geq 0\}$; or alternatively in a (face, or half-space) H-representation, namely $\mathbb{Y} = \{z \in \mathbb{R}^n \mid Az \leq b\}$, where $v_i \in \mathbb{R}^n$, $A \in \mathbb{R}^{l \times n}$, $b \in \mathbb{R}^l$, and $N, l \in \mathbb{N}$.

Focusing on an MDP model M and the associated MTS, the following result provides a different way to represent $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$ if $\Pi \subseteq \mathcal{P}(\mathbb{X})$ is assumed to be a convex polytope.

Proposition 2. *If $\Pi \subseteq \mathcal{P}(\mathbb{X})$ is a convex polytope, then*

- *$\mathcal{BR}_\exists(\Pi)$ is a convex polytope and can be rewritten as*

$$\mathcal{BR}_\exists(\Pi) = \left\{ (Q\mathbf{1})^T \;\middle|\; \begin{array}{l} Q \in \mathbb{R}^{n \times m},\ Q \geq 0, \\ (Q\mathbf{1})^T \in \mathcal{P}(\mathbb{X}), \pi \in \Pi, \\ \forall y \in \mathbb{X}, \\ \pi(y) = \sum\limits_{x \in \mathbb{X}} \sum\limits_{u \in \mathbb{U}_x} T(y|x,u) Q(x,u) \end{array} \right\}; \tag{6}$$

- *$\mathcal{BR}_\forall(\Pi)$ is a convex polytope and can be rewritten as*

$$\mathcal{BR}_\forall(\Pi) = \left\{\pi \in \mathcal{P}(\mathbb{X}) \;\middle|\; \forall \mu^d \in \mathcal{U}^d, \pi P^{\mu^d} \in \Pi\right\}. \tag{7}$$

Proof. Let us first consider the expression of $\mathcal{BR}_\exists(\Pi)$. From (1), the distribution dynamics (2) can be rewritten as

$$\pi' = \sum_{x \in \mathbb{X}} \pi(x) P^\mu(y|x) = \sum_{x \in \mathbb{X}} \pi(x) \Big(\sum_{u \in \mathbb{U}_x} T(y|x,u) \mu(u|x) \Big)$$

for some $\mu \in \mathcal{U}$. It follows that the product of $\mu(u|x)$ and $\pi(x)$ can be replaced by a matrix $Q \in \mathbb{R}^{n \times m}$ with $Q \geq 0$ and $(Q\mathbf{1})^T \in \mathcal{P}(\mathbb{X})$. Thus, the set $\mathcal{BR}_\exists(\Pi)$ defined in (4) can be rewritten as (6). If $\Pi \subseteq \mathcal{P}(\mathbb{X})$ is a convex polytope, we have that there are a finite number of inequalities and equalities in (6), which define a convex polytope in $\mathbb{R}^{n \times m + n}$. The operation $(Q\mathbf{1})^T$ follows that the set

Algorithm 1 Sampling-based Backward Reach Set Computation

Input: convex polytopes Π and Γ with $\Pi \subseteq \mathcal{P}(\mathbb{X}) \subset \text{int}(\Gamma)$, nr. of samples $N_s \in \mathbb{N}_{\geq 1}$

1: select uniformly at random a group of samples $\{\pi_i^s\}_{i=1}^{N_s}$ from Γ;
2: **for** $i = 1 : N_s$ **do**
3: compute $\pi_i^{1s} = \underset{\pi \in \mathcal{BR}_\exists(\Pi)}{\text{argmin}} \|\pi - \pi_i^s\|^2$ and $\pi_i^{2s} = \underset{\pi \in \mathcal{BR}_\forall(\Pi)}{\text{argmin}} \|\pi - \pi_i^s\|^2$;
4: **end for**
5: **return** $\widehat{\mathcal{BR}}_\exists(\Pi, N_s) = \text{conv}\big(\{\pi_i^{1s}, i \in \mathbb{N}_{[1,N_s]}\}\big)$ and $\widehat{\mathcal{BR}}_\forall(\Pi, N_s) = \text{conv}\big(\{\pi_i^{2s}, i \in \mathbb{N}_{[1,N_s]}\}\big)$.

$\mathcal{BR}_\exists(\Pi)$ is the affine projection of this convex polytope onto $\mathbb{R}^n$, which is still a convex polytope.

For the set $\mathcal{BR}_\forall(\Pi)$, note that each policy $\mu \in \mathcal{U}$ is a distribution over the set of deterministic policies $\mathcal{U}^d$, and that the set $\mathcal{U}^d$ has finite cardinality. Thus, $\mathcal{BR}_\forall(\Pi)$ can be can be rewritten as (7) and it is also a convex polytope if Π is a convex polytope. □

Remark 2 (Policy Synthesis). *The matrix Q used to define the existential backward reachable set in* (7) *is called "occupation measure" in the literature: this enables to recover a policy μ by*

$$\mu(u|x) = \begin{cases} \frac{Q(x,u)}{\sum_{v \in \mathbb{U}_x} Q(x,v)} & \text{if } \sum_{v \in \mathbb{U}_x} Q(x,v) > 0, \\ \frac{1}{|\mathbb{U}_x|}, & \text{if } \sum_{v \in \mathbb{U}_x} Q(x,v) = 0 \ \& \ u \in \mathbb{U}_x. \end{cases}$$

The use of occupation measures allows to reformulate a constrained MDP problem as a linear program [6], whose solution can be used to recover a sequence of (time-dependent) policies for a finite-horizon problem (cf. Case Study 2). □

From Proposition 2 we can observe that, even if the set Π is a polytope in the form of either V-representation or H-representation, it can still be quite challenging computationally to exactly compute the polytopic sets $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$, particularly whenever the MDP M has a large number n of states. The main reason is that both $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$ depend on extra variables (e.g., the matrix Q in (6)), and that the necessary set projection has exponential computational complexity with respect to the space dimension [23]. In the following we discuss Algorithm 1, a scalable, sampling-based method to under-approximate sets $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$.

The input to Algorithm 1 consists of two convex polytopes Π and Γ with $\Pi \subseteq \mathcal{P}(\mathbb{X}) \subset \text{int}(\Gamma)$, and the number of samples $N_s \in \mathbb{N}_{\geq 1}$, where int denotes the set interior. In line 1, we select uniformly at random samples $\{\pi_i^s\}_{i=1}^{N_s}$ in $\mathbb{R}^n$ from Γ. Then, these samples are used to generate samples $\pi_i^{1s} \in \mathcal{BR}_\exists(\Pi)$ and $\pi_i^{2s} \in \mathcal{BR}_\forall(\Pi)$ in line 3, by projecting π_i^s onto $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$, respectively, w.r.t to the two-norm $\|\cdot\|$. The output of Algorithm 1 comprises the convex hulls of these projected samples, namely $\widehat{\mathcal{BR}}_\exists(\Pi, N_s)$ and $\widehat{\mathcal{BR}}_\forall(\Pi, N_s)$. The following proposition discusses two important properties of Algorithm 1.

Proposition 3. *Consider two convex polytopes Π and Γ with $\Pi \subseteq \mathcal{P}(\mathbb{X}) \subset \text{int}(\Gamma)$ and an integer $N_s \in \mathbb{N}_{\geq 1}$. Let $N_\exists^v$ and $N_\forall^v$ be the number of the vertices of $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$, respectively. Under Algorithm 1, the sets $\widehat{\mathcal{BR}}_\exists(\Pi, N_s)$ and $\widehat{\mathcal{BR}}_\forall(\Pi, N_s)$ are under approximations of $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$, respectively, for all $N_s \in \mathbb{N}_{\geq 1}$. In particular, there exist $0 < \alpha, \beta < 1$ such that*

$$\Pr(\widehat{\mathcal{BR}}_\exists(\Pi, N_s) = \mathcal{BR}_\exists(\Pi)) \geq 1 - N_\exists^v \alpha^{N_s},$$
$$\Pr(\widehat{\mathcal{BR}}_\forall(\Pi, N_s) = \mathcal{BR}_\forall(\Pi)) \geq 1 - N_\forall^v \beta^{N_s}.$$

Proof. The under approximation relation between $\widehat{\mathcal{BR}}_\exists(\Pi, N_s)$ and $\mathcal{BR}_\exists(\Pi)$ (or $\widehat{\mathcal{BR}}_\forall(\Pi, N_s)$ and $\mathcal{BR}_\forall(\Pi)$) directly follows from that the projected samples $\pi_i^{1s} \in \mathcal{BR}_\exists(\Pi)$ and $\pi_i^{2s} \in \mathcal{BR}_\forall(\Pi)$ (see line 3 in Algorithm 1).

Denote by η a uniform probability measure assigned to Γ, i.e., $\int_\Gamma \eta(t)dt = 1$, and by $\mathcal{V}_\exists$ the set of vertices of $\mathcal{BR}_\exists(\Pi)$. Since each vertex $\pi \in \mathcal{V}_\exists$ of the set $\widehat{\mathcal{BR}}_\exists(\Pi, N_s) \subseteq \mathcal{P}(\mathbb{X}) \subset \text{int}(\Gamma))$, we have the following two facts: (1) for each vertex $\pi \in \mathcal{V}_\exists$, the subset of Γ from which the projection onto $\mathcal{BR}_\exists(\Pi)$ is π has measure strictly greater than 0, denoted by $\eta(\pi)$; (2) these subsets are disjoint for different vertices. To show this, let us treat the projection in Algorithm 1 as a multi-parametric quadratic program, where the π_i^s is the parameter. Then, the above two facts follow from the result in [37] which shows that the optimal solution to the multi-parametric quadratic program results in a polyhedral partition of the parameter set. Given these two facts, we have

$$\Pr(\widehat{\mathcal{BR}}_\exists(\Pi, N_s) = \mathcal{BR}_\exists(\Pi)) = \Pr\Big(\mathcal{V}_\exists \subseteq \{\pi_{1i}^s, i \in \mathbb{N}_{[1,N_s]}\}\Big) \geq 1 - \sum_{\pi \in \mathcal{V}_\exists} (1 - \eta(\pi))^{N_s}.$$

We complete the proof by choosing $\alpha = \max\{1 - \eta(\pi) \mid \pi \in \mathcal{V}_\exists\}$. Similar arguments apply to the approximation between $\widehat{\mathcal{BR}}_\forall(\Pi, N_s) = \mathcal{BR}_\forall(\Pi)$. □

Computational Aspects. Uniform sampling from a general convex polytope Γ is usually challenging. To facilitate the sampling, a good choice of Γ is a hyperrectangle in $\mathbb{R}^n$ which is a strict superset of $\mathcal{P}(\mathbb{X})$. In this case, the complexity of line 1 in Algorithm 1 is $O(N_s)$. The computational complexity of the for-loop is linear with the number of samples N_s and polynomial with the number of states n and with the number of actions m. Since the sets $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$ are convex polytopes, projecting each sample π_i^s onto $\mathcal{BR}_\exists(\Pi)$ and $\mathcal{BR}_\forall(\Pi)$ in line 3 is a convex quadratic program with $n + nm$ decision variables, whose complexity is $O((n + nm)^3)$ using the interior point method in [39]. Thus, the total computational complexity of Algorithm 1 is $O(N_s(n + nm)^3 + N_s)$.

5.3 Approximate CTL Model Checking over Distribution Spaces

Leveraging the sampling-based computation of backward reachable sets, discussed in the previous subsection, we are now ready to design an approximate, yet sound, CTL model checking algorithm for MDPs over their distribution spaces. We focus our attention to a fragment of CTL formulae expressed in the following form:

$$\begin{cases} \Phi ::= \text{true} \mid \text{false} \mid a \mid \neg a \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \vee \Phi_2 \mid \exists \varphi \mid \forall \varphi, \\ \varphi ::= \bigcirc \Phi \mid \Phi_1 \mathsf{U} \Phi_2. \end{cases} \tag{8}$$

where $a \in \mathcal{AP}_d$. The CTL formulae in (8) assumes that negations can only occur to basic atomic propositions.

Next, we will show that the satisfaction set of each CTL formula in (8) can be soundly under-approximated by the union of a set of convex polytopes, under the following assumption. We remark that this assumption and the latter Lemma 3 echo why we focus on the CTL formulae in (8): if the set $\mathrm{Sat}(\Phi)$ is under-approximated by unions of convex polytopes, such under-approximation will in general not hold for $\mathrm{Sat}(\neg\Phi)$, unless Φ is an atomic proposition.

Assumption 1. *For each atomic proposition $a \in \mathcal{AP}_d$, the set of distributions associated with the labeling function L_d, denoted by $L_d^{-1}(a) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid a \in L_d(\pi)\}$, is a convex polytope, later considered in its H-representation.*

Let us recall how to compute the set complement of a convex polytope.

Lemma 1. *Consider a convex polytope $\mathbb{Y}_0$ with $\mathbb{Y}_0 \subseteq \mathcal{P}(\mathbb{X})$. Suppose $\mathbb{Y}_0 = \{z \in \mathbb{R}^n \mid Az \leq b\}$ with $A \in \mathbb{R}^{l\times n}$ and $b \in \mathbb{R}^l$. Let $\mathbb{Y}_i = \{z \in \mathcal{P}(\mathbb{X}) \mid [A]_i z \geq b_i + \epsilon_i\}, \forall i \in \mathbb{N}_{[1,l]}$, where $[A]_i$ and b_i denotes the i-th row of A and b, respectively, and ϵ_i is a small positive constant. Then, $\bigcup_{i=1}^{l} \mathbb{Y}_i \subset \mathcal{P}(\mathbb{X}) \setminus \mathbb{Y}_0$.*

The use of ϵ_i is to ensure the closure of the set $\mathbb{Y}_i$. Lemma 1 implies that, under Assumption 1, for each atomic proposition $a \in \mathcal{AP}_d$, the satisfaction set $\mathrm{Sat}(\neg a)$, i.e., the complement of $L_d^{-1}(a)$ with respect to $\mathcal{P}(\mathbb{X})$, can be under-approximated by the union of a set of convex polytopes.

The following lemma shows that the backward-reachable sets obtained from the union of a set of convex polytopes can be under-approximated by the union of the backward reachable sets obtained from the corresponding convex polytopes.

Lemma 2. *Consider a group of sets $\{\Pi_i\}_{i=1}^{M}$, where each $\Pi_i \subseteq \mathcal{P}(\mathbb{X})$ is a convex polytope. For any $N_i \in \mathbb{N}_{\geq 1}$, $i \in \mathbb{N}_{[1,M]}$, we have*

$$\begin{cases} \cup_{i=1}^{M} \widehat{\mathcal{BR}}_{\exists}(\Pi_i, N_i) \subseteq \mathcal{BR}_{\exists}(\cup_{i=1}^{M} \Pi_i), \\ \cup_{i=1}^{M} \widehat{\mathcal{BR}}_{\forall}(\Pi_i, N_i) \subseteq \mathcal{BR}_{\forall}(\cup_{i=1}^{M} \Pi_i). \end{cases} \tag{9}$$

The proof of Lemma 2 directly follows from Proposition 3. The next lemma shows that the satisfaction sets obtained by applying propositional and temporal operator can be under-approximated by the union of convex polytopes.

Lemma 3. *Consider three CTL formulae Φ, Φ_1 and Φ_2 in (8). Suppose that their satisfaction sets $\mathrm{Sat}(\Phi)$, $\mathrm{Sat}(\Phi_1)$, $\mathrm{Sat}(\Phi_2)$ are respectively under-approximated by unions of convex polytopes. Then, the sets $\mathrm{Sat}(\Phi_1 \wedge \Phi_2)$, $\mathrm{Sat}(\Phi_1 \vee \Phi_2)$, $\mathrm{Sat}(\exists \bigcirc \Phi)$, $\mathrm{Sat}(\forall \bigcirc \Phi)$, $\mathrm{Sat}(\exists \Phi_1 \mathrm{U} \Phi_2)$, and $\mathrm{Sat}(\forall \Phi_1 \mathrm{U} \Phi_2)$ can be under-approximated by finite unions of convex polytopes.*

Proof. Suppose $\cup_{i=1}^{M_\Phi} \Pi_i^\Phi \subseteq \mathrm{Sat}(\Phi)$, $\cup_{i=1}^{M_{\Phi_1}} \Pi_i^{\Phi_1} \subseteq \mathrm{Sat}(\Phi_1)$, $\cup_{i=1}^{M_{\Phi_2}} \Pi_i^{\Phi_2} \subseteq \mathrm{Sat}(\Phi_2)$, where Π_i^Φ, $\Pi_j^{\Phi_1}$, and $\Pi_k^{\Phi_2}$ are convex polytopes for all $i \in \mathbb{N}_{[1,M_\Phi]}$, $j \in \mathbb{N}_{[1,M_{\Phi_1}]}$, and $k \in \mathbb{N}_{[1,M_{\Phi_2}]}$.

Let us first consider $\mathrm{Sat}(\Phi_1 \wedge \Phi_2)$. Applying the distributive law of set operations, we obtain that

$$\begin{aligned} \mathrm{Sat}(\Phi_1 \wedge \Phi_2) &\supseteq \left(\cup_{i=1}^{M_{\Phi_1}} \Pi_i^{\Phi_1}\right) \cap \left(\cup_{j=1}^{M_{\Phi_2}} \Pi_j^{\Phi_2}\right) \\ &= \cup_{\substack{i\in\mathbb{N}_{[1,M_{\Phi_1}]} \\ j\in\mathbb{N}_{[1,M_{\Phi_2}]}}} \left(\Pi_i^{\Phi_1} \cap \Pi_j^{\Phi_2}\right). \end{aligned}$$

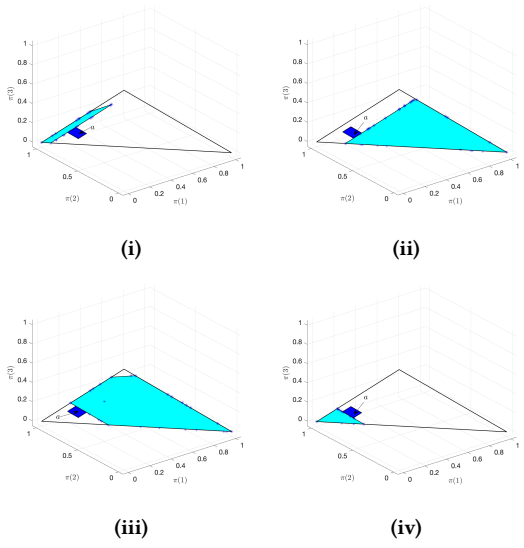


Figure 2: Four distribution sets in cyan, whose union under-approximates $\mathrm{Sat}(\exists(\neg a \mathrm{U} b))$. Here a labels the blue polytope. The stars are projections of the samples onto the boundary of satisfaction sets.

Note that the intersection of two convex polytopes is either a convex polytope or an empty set. Thus, $\mathrm{Sat}(\Phi_1 \wedge \Phi_2)$ can be under-approximated by the union of convex polytopes. For the disjunction operator, it is straightforward to see that $\mathrm{Sat}(\Phi_1 \vee \Phi_2) \supseteq \left(\cup_{i=1}^{M_{\Phi_1}} \Pi_i^{\Phi_1}\right) \cup \left(\cup_{j=1}^{M_{\Phi_2}} \Pi_j^{\Phi_2}\right)$.

Let us now consider basic formulae with temporal operators. For the next operator, it follows from Lemma 2 that $N_i \in \mathbb{N}_{\geq 1}$, $i \in \mathbb{N}_{[1,M]}$,

$$\begin{aligned} \mathrm{Sat}(\exists \bigcirc \Phi) &\supseteq \mathcal{BR}_{\exists}(\cup_{i=1}^{M_\Phi} \Pi_i^\Phi) \supseteq \cup_{i=1}^{M_\Phi} \widehat{\mathcal{BR}}_{\exists}(\Pi_i^\Phi, N_i), \\ \mathrm{Sat}(\forall \bigcirc \Phi) &\supseteq \mathcal{BR}_{\forall}(\cup_{i=1}^{M_\Phi} \Pi_i^\Phi) \supseteq \cup_{i=1}^{M_\Phi} \widehat{\mathcal{BR}}_{\forall}(\Pi_i^\Phi, N_i). \end{aligned}$$

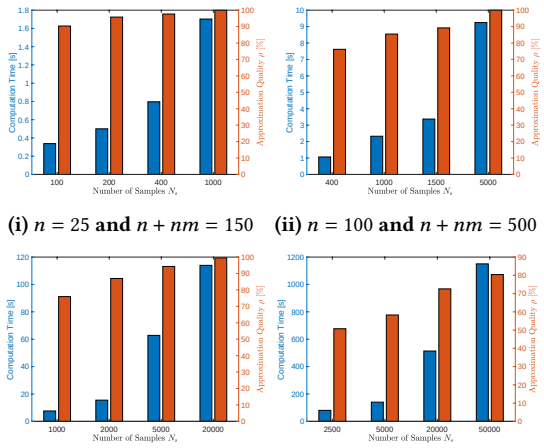
For the until operator, let us define the set sequences for each set $\Pi_j^{\Phi_2}$, $j \in \mathbb{N}_{[1,M_{\Phi_2}]}$:

$$\begin{aligned} \hat{\mathbb{T}}_{i+1,j} &= \left(\cup_{i=1}^{M_{\Phi_1}} \Pi_i^{\Phi_1}\right) \cap \widehat{\mathcal{BR}}_{\exists}(\hat{\mathbb{T}}_{i,j}, N_{i,j}) \text{ with } \hat{\mathbb{T}}_{0,j} = \Pi_j^{\Phi_2}, \\ \hat{\mathbb{S}}_{i+1,j} &= \left(\cup_{i=1}^{M_{\Phi_1}} \Pi_i^{\Phi_1}\right) \cap \widehat{\mathcal{BR}}_{\forall}(\hat{\mathbb{S}}_{i,j}, N_{i,j}) \text{ with } \hat{\mathbb{S}}_{0,j} = \Pi_j^{\Phi_2}. \end{aligned}$$

where $N_{i,j} \in \mathbb{N}_{\geq 1}$. By the distributive law and following Lemma 2, we can recursively show that both $\hat{\mathbb{T}}_{i,j}$ and $\hat{\mathbb{S}}_{i,j}$ can be represented as the union of a finite number of convex polytopes. It follows from that $\mathrm{Sat}(\exists \Phi_1 \mathrm{U} \Phi_2)$ and $\mathrm{Sat}(\forall \Phi_1 \mathrm{U} \Phi_2)$ can be, respectively, under-approximated by $\cup_{j=1}^{M_{\Phi_1}} \cup_{i=1}^{N_1} \hat{\mathbb{T}}_{i,j}$ and $\cup_{j=1}^{M_{\Phi_1}} \cup_{i=1}^{N_2} \hat{\mathbb{S}}_{i,j}$, for all $N_1, N_2 \in \mathbb{N}_{\geq 1}$, both of which are union of convex polytopes. □

The following example shows satisfaction sets of the CTL formulae in Example 2.

Example 3. *Let us recall the opinion consensus in Example 2. Consider the formula $\exists(\neg a \mathrm{U} b)$. Applying Algorithm 1 and Lemmata 1–3, we obtain the under-approximation of the satisfaction set*



(i) $n = 25$ **and** $n + nm = 150$ **(ii)** $n = 100$ **and** $n + nm = 500$

(iii) $n = 400$ **and** $n+nm = 2400$ **(iv)** $n = 1225$ **and** $n+nm = 7350$

Figure 3: Computation time and approximation quality ρ for different MDPs with different number of samples N_s. Here, n=number of states, $n + nm$=space dimension.

$\mathsf{Sat}(\exists(\neg a \mathsf{U} b))$, *which is the union of the four cyan sets shown in Fig. 2(i)-(iv). This is the set of beliefs from which there exists a policy such that the agents' beliefs approach consensus (reaching the set of* $L^{-1}(b)$*) while avoiding the beliefs in the set* $L^{-1}(a)$.

We now summarise the solution to Problem 2 by formalising a result on CTL model checking over the distribution space of MDPs.

Theorem 2. *Consider the MDP* M *and the MDP-induced transition system* MTS. *Suppose that, given a CTL formula* Φ *in* (8)*, Assumption 1 holds. Then, Algorithm 1 can be straightforwardly used to design a reachability-based CTL model checking algorithm that computes a sound under-approximation of the satisfaction set* $\mathsf{Sat}(\Phi)$ *as a finite union of convex polytopes.*

Proof. Recall that the CTL model checking can be performed by a recursive procedure that calculates the satisfaction set for all subformulae of Φ. If Φ is in (8), it follows from Lemmas 1– 3 that under Assumption 1, the satisfaction set of each subformula of Φ can be under-approximated by the finite union of convex polytopes. The under-approximated set is computed by applying Algorithm 1, which is later shown to be computationally tractable.

Due to the under-approximation relation, we have that if an initial distribution π_0 belongs to this under-approximated set, then $\pi_0 \vDash \Phi$, which implies the soundness of the overall approach. □

6 QUALITY AND SCALABILITY OF ALGORITHM 1

We test the scalability and approximation quality of Algorithm 1 under different MDPs, with increasing number of states. The experiment was run with MATLAB 2021b on an ARM system with 8GB RAM. We compute existential backward reachable sets over multiple runs. Recall that the computation of existential backward reachable set is based on the polytope projection from $\mathbb{R}^{nm+n}$ to $\mathbb{R}^n$, where n and m are number of states and actions in the MDP, and N_s is the number of samples employed in Algorithm 1 (see Eq. (6)). Denote by $\mathcal{BR}$ the exact set and by $\widehat{\mathcal{BR}}_\exists(N_s)$ the approximate set from Algorithm 1 using N_s samples. Since computing the volume of the convex polytopes efficiently in high-dimensional spaces is challenging, we define the following quantity to measure the approximation quality: $\rho = \frac{1}{n}\sum_{i=1}^{n} \frac{d_{i,\max}-d_{i,\min}}{D_{i,\max}-D_{i,\min}}$ where $d_{i,\max} = \max_{\pi\in\widehat{\mathcal{BR}}_\exists(N_s)} e_i^T\pi$, $d_{i,\min} = \min_{\pi\in\widehat{\mathcal{BR}}_\exists(N_s)} e_i^T\pi$, $D_{i,\max} = \max_{\pi\in\mathcal{BR}_\exists} e_i^T\pi$, and $D_{i,\min} = \min_{\pi\in\mathcal{BR}_\exists} e_i^T\pi$. The set $\prod_{i=1}^{n}[D_{i,\min}, D_{i,\max}]$ is the smallest hyper-rectangle that contains $\mathcal{BR}_\exists$, while $\prod_{i=1}^{n}[d_{i,\min}, d_{i,\max}]$ is the smallest hyper-rectangle that contains $\widehat{\mathcal{BR}}_\exists(N_s)$. The value of ρ quantifies the average ratios of the corresponding edges of these two hyper-rectangles, and is a reasonable measure for the approximation quality of Algorithm 1 (since $\widehat{\mathcal{BR}}_\exists(N_s) = \mathcal{BR}_\exists$ implies $\rho = 1$).

Fig. 3 reports the computation time and corresponding value of ρ for different MDPs, under different N_s. The quadratic programs in Algorithm 1 were solved by *Yalmip* [32] and *Mosek* [1]. Note that Algorithm 1 performs set projections on spaces with a dimension up to $nm + n = 2400$ very efficiently, and for 7350-dimensional spaces in a manageable time. We observe that the computation time is linear with respect to N_s. The average time to solve the QP in line 3 of Algorithm 1 is 0.0015[s] ($n = 25$), 0.0020[s] ($n = 100$), 0.0071[s] ($n = 400$), and 0.0243 [s] ($n = 1225$). The approximation quality ρ increases with respect to N_s. In particular, as shown in Fig. 3, ρ can reach 1 for the MDPs with $n = 25, 100$, and 400, that is, we obtain a tight approximation between $\widehat{\mathcal{BR}}_\exists(N_s)$ and $\mathcal{BR}_\exists$. For the MDP with $n = 1225$, ρ can reach 80%, which implies a good approximation in a high-dimensional space.

Limitations of the state of the art. The exponential computational complexity restricts the polytope projections in high-dimensional spaces in [23], where the experiments are restricted to spaces no greater than 35. To the best of our knowledge, the current algorithms can only estimate the volume of convex polytopes in $\mathbb{R}^n$ with $n < 200$ and the computation time for the set in $\mathbb{R}^{196}$ reaches 8.8 [h], see [16]. Known tools in computational geometry, e.g., *MPT3* [21], *Qhull* [8], and *bensolve* [33], are not usable to handle the cases in our experiments with space dimensions $n + nm \geq 150$. Thus, Algorithm 1 significantly expands the frontiers of the state of the art: it scales much better, and provides remarkable approximations in high-dimensional spaces, which validates its usability for CTL model checking on large-scale MDPs.

7 CASE STUDIES

In this section we validate our model checking algorithms on a pharmacokinetics model and on an unmanned aerial vehicle (UAV) path planning problem.

Drug Injection Synthesis for a Pharmacokinetics System. Recall in Example 1 the MDP model and the distribution-specified CTL formula $\Phi = \Phi_1 \wedge \Phi_2 \wedge \Phi_3$, with $\Phi_1 = \forall\Box\texttt{nontoxic}$, $\Phi_2 = \forall\Diamond(\texttt{effective} \wedge \forall \bigcirc \texttt{effective})$, and $\Phi_3 = \forall\Diamond\texttt{cleared}$. Despite the use of universal quantifiers in Φ, we shall still look at a synthesis problem over the values of α. Since the state Cl is the unique absorbing state, for any $\epsilon > 0$ (cf. Example 1), the formula Φ_3 always holds. Using the MPT3 toolbox [21], we obtain that the feasible set for α is $[0.0538, 0.0590]$, with a computation time equal to 11.27 [s].

UAV Path Planning. As shown in Fig. 4, a UAV roams a 5×5 "slippery grid world" and has five possible actions {*up, down, left, right, stay*}. Due to environmental uncertainties (e.g., noise), we assume that the first four actions will move the state to multiple neighboring states. More details can be found in Appendix C. The cyan regions are obstacles, which we assume are absorbing. Denote by Obs the set of obstacle states. The red region $(5, 5)$ is the target state, denoted by Target. The initial state is the blue square at $(1, 1)$. We consider the following path planning problem: *to find a feasible policy such that the UAV, starting from the initial state, reaches the target set, whilst avoiding the obstacle states, with a desired probability.*

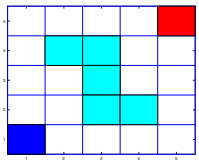
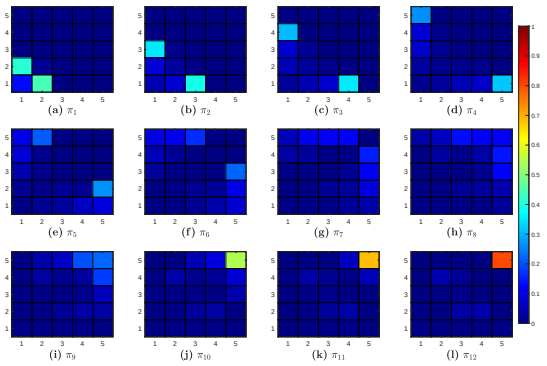


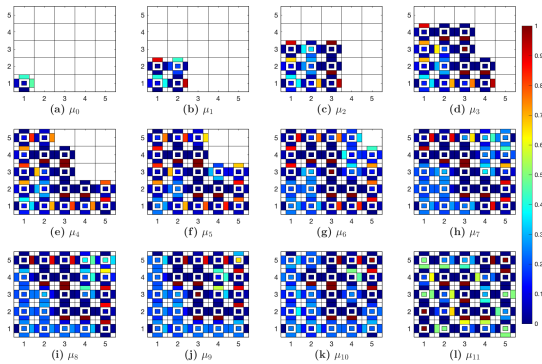
Figure 4: UAV Motion Planning

This problem can be studied by introducing an MDP model M and the corresponding MDP-induced transition system MTS. The distribution space $\mathcal{P}(\mathbb{X})$ is a subset of $\mathbb{R}^{25}$, and the policy set $\mathcal{U}$ is defined as in Definition 2. Let the set of atomic propositions be $\mathcal{AP}_d = \{b^{\beta}_{\text{unsafe}}, b^{p}_{\text{target}}\}$, and introduce a labeling function be L_d such that $L_d^{-1}(b^{\beta}_{\text{unsafe}}) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \sum_{x\in\text{Obs}} \pi(x) \leq \beta\}$ and $L_d^{-1}(b^{p}_{\text{target}}) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \pi(\text{Target}) \geq p\}$, where $\beta \in [0, 1]$ and $p \in (0, 1]$. That is, the sets $L_d^{-1}(b^{\beta}_{\text{unsafe}})$ and $L_d^{-1}(b^{p}_{\text{target}})$ are parameterised by β and p, respectively. Let us denote by π_0 the distribution concentrated on the deterministic initial state $\{(1, 1)\}$. The existence of a policy solving the motion planning problem can be asserted if π_0 satisfies the CTL formula $\Phi_{\text{CTL}} = \exists(\neg b^{\beta}_{\text{unsafe}} \cup b^{p}_{\text{target}})$, and is obtained according to Remark 2.

We set β to 0.15 and p to 0.80, which in practice means that the UAV is required to stay in the safe region with probability greater than 0.85 before the probability of reaching the Target set is greater than 0.80. Implementing the approximate CTL model checking algorithm, we find that π_0 is in the sound approximate satisfaction set (see result in Theorem 2). We can find a feasible realisation of the transient state distributions over 12 steps, which is shown in Fig. 5(i). We see that as desired, the probability of entering the obstacles (as in Fig. 4) is no greater than 0.15 within the 12 steps. As there is a transition probability into undesired neighbouring states under the first four actions, the agent might slip into the obstacle with very small probability: this explains why the probability within the obstacle is changing, as per Fig. 5(i). According to Remark 2, we can further distill a feasible policy, as shown in Fig. 5(ii): here, the small blocks within each state are positioned corresponding to the actions {*up, down, left, right, stay*}, and the heat map indicates the distributions over these actions. The experiment was run on an ARM system with 8GB RAM using *Yalmip* [32] and *Mosek* [1]. The computation time for model checking was 24.57 [s] whilst that for policy synthesis was 1.74 [s]. The case that β is set to 0 is discussed in Appendix C.



(i) State distribution $\pi_k, k = 1, \ldots, 12$



(ii) Generated policy $\mu_k, k = 0, \ldots, 11$

Figure 5: UAV Path Planning on MDP with noisy transitions. (i) Evolution of state distribution that, initialised as $\pi_0 = e_{(1,1)}$ and under a feasible policy, reaches the target state $(5, 5)$ with probability grater than 0.80, while possibly entering the obstacles (as in Fig. 4) with probability no greater than 0.15 at all times. (ii) Evolution of policy μ_k which is a distribution over actions {*up, down, left, right, stay*} (we visualise policies only at the states whose transient probabilities are greater than 0).

8 CONCLUSIONS

We have introduced a verification framework for finite MDPs over the space of their transient distributions. We have employed CTL to specify temporal properties, and shown that this provides an alternative way to express probabilistic specifications for the MDP. We have compared the semantics of CTL formulae over distribution spaces with traditional PCTL specifications. We have proposed novel reachability-based CTL model checking algorithms over distribution spaces, as well as more tractable sample-based procedures for computing reachable sets: it is shown that the satisfaction set of the CTL formula can be soundly under-approximated by the union of convex polytopes.

In parallel with the distribution-based CTL model checking, another worthwhile goal is the policy synthesis for distribution-specified LTL requirements. We are also interested in exploring

finite-state, non-stochastic abstractions of the MDP-induced transition system MTS and developing the theory of simulation relations between the abstract model and the concrete MTS.

ACKNOWLEDGMENTS

This work is supported in by Swedish Research Council Distinguished Professor Grant 2017-01078, Swedish Research Council International Postdoc Grant 2021-06727, and Knut and Alice Wallenberg Foundation Wallenberg Scholar Grant.

REFERENCES

[1] [n. d.]. *MOSEK Software*. https://www.mosek.com/
[2] Manindra Agrawal, Sundararaman Akshay, Blaise Genest, and PS Thiagarajan. 2015. Approximate verification of the symbolic dynamics of Markov chains. *J. ACM* 62, 1 (2015), 1–34.
[3] S Akshay, Timos Antonopoulos, Joël Ouaknine, and James Worrell. 2015. Reachability problems for Markov chains. *Inform. Process. Lett.* 115, 2 (2015), 155–158.
[4] S Akshay, Krishnendu Chatterjee, Tobias Meggendorfer, and Đorđe Žikelić. 2023. MDPs as distribution transformers: affine invariant synthesis for safety objectives. In *International Conference on Computer Aided Verification*. 86–112.
[5] S Akshay, Blaise Genest, and Nikhil Vyas. 2018. Distribution-based objectives for Markov Decision Processes. In *33rd Annual ACM/IEEE Symposium on Logic in Computer Science*. 36–45.
[6] Eitan Altman. 1999. *Constrained Markov Decision Processes: Stochastic Modeling*. Routledge.
[7] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of Model Checking*. MIT press.
[8] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. 1996. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)* 22, 4 (1996), 469–483.
[9] Daniele Beauquier, Alexander Rabinovich, and Anatol Slissenko. 2002. A logic of probability with decidable model-checking. In *International Workshop on Computer Science Logic*. 306–321.
[10] Calin Belta, Boyan Yordanov, and Ebru Aydin Gol. 2017. *Formal Methods for Discrete-time Dynamical Systems*. Springer.
[11] Rohit Chadha, Vijay Anand Korthikanti, Mahesh Viswanathan, Gul Agha, and YoungMin Kwon. 2011. Model checking MDPs with a unique compact invariant set of distributions. In *8th International Conference on Quantitative Evaluation of Systems*. 121–130.
[12] Edmund M Clarke and E Allen Emerson. 1981. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Workshop on Logic of Programs*. 52–71.
[13] Edmund M. Clarke, E Allen Emerson, and A Prasad Sistla. 1986. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems* 8, 2 (1986), 244–263.
[14] Giacomo Como and Fabio Fagnani. 2015. Robustness of large-scale stochastic matrices to localized perturbations. *IEEE Transactions on Network Science and Engineering* 2, 2 (2015), 53–64.
[15] Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. 2017. A STORM is coming: A modern probabilistic model checker. In *International Conference on Computer Aided Verification*. 592–600.
[16] Ioannis Z Emiris and Vissarion Fisikopoulos. 2018. Practical polytope volume approximation. *ACM Trans. Math. Software* 44, 4 (2018), 1–21.
[17] Yuan Feng and Lijun Zhang. 2014. When equivalence and bisimulation join forces in probabilistic automata. In *International Symposium on Formal Methods*. 247–262.
[18] Vojtěch Forejt, Marta Kwiatkowska, Gethin Norman, and David Parker. 2011. Automated verification techniques for probabilistic systems. In *International School on Formal Methods for the Design of Computer, Communication and Software Systems*. 53–113.
[19] Yulong Gao, Karl Henrik Johansson, and Lihua Xie. 2020. Computing probabilistic controlled invariant sets. *IEEE Trans. Automat. Control* 66, 7 (2020), 3138–3151.
[20] Hans Hansson and Bengt Jonsson. 1994. A logic for reasoning about time and reliability. *Formal Aspects of Computing* 6, 5 (1994), 512–535.
[21] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. 2013. Multi-Parametric Toolbox 3.0. In *European Control Conference*. 502–510.
[22] Holger Hermanns, Jan Krčál, and Jan Křetínskỳ. 2014. Probabilistic bisimulation: naturally on distributions. In *International Conference on Concurrency Theory*. 249–265.
[23] Rui-Juan Jing, Marc Moreno-Maza, and Delaram Talaashrafi. 2020. Complexity estimates for Fourier-Motzkin elimination. In *22nd International Workshop on Computer Algebra in Scientific Computing*. 282–306.
[24] Austin Jones, Mac Schwager, and Calin Belta. 2013. Distribution temporal logic: Combining correctness with quality of estimation. In *52nd IEEE Conference on Decision and Control*. 4719–4724.
[25] Joost-Pieter Katoen. 2016. The probabilistic model checking landscape. In *31st Annual ACM/IEEE Symposium on Logic in Computer Science*. 31–45.
[26] Vijay Anand Korthikanti, Mahesh Viswanathan, Gul Agha, and YoungMin Kwon. 2010. Reasoning about MDPs as transformers of probability distributions. In *7th International Conference on the Quantitative Evaluation of Systems*. 199–208.
[27] Marta Kwiatkowska, Gethin Norman, and David Parker. 2007. Stochastic model checking. In *International School on Formal Methods for the Design of Computer, Communication and Software Systems*. 220–270.
[28] Marta Kwiatkowska, Gethin Norman, and David Parker. 2009. PRISM: probabilistic model checking for performance and reliability analysis. *ACM SIGMETRICS Performance Evaluation Review* 36, 4 (2009), 40–45.
[29] Marta Kwiatkowska, Gethin Norman, and David Parker. 2018. Probabilistic model checking: advances and applications. In *Formal System Verification*. Springer, 73–121.
[30] YoungMin Kwon and Gul Agha. 2004. Linear inequality LTL (iLTL): A model checker for discrete time Markov chains. In *International Conference on Formal Engineering Methods*. 194–208.
[31] YoungMin Kwon and Gul Agha. 2010. Verifying the evolution of probability distributions governed by a DTMC. *IEEE Transactions on Software Engineering* 37, 1 (2010), 126–141.
[32] J. Löfberg. 2004. YALMIP : A Toolbox for Modeling and Optimization in MATLAB. In *In Proceedings of the CACSD Conference*.
[33] Andreas Löhne and Benjamin Weißing. 2016. Equivalence between polyhedral projection, multiple objective linear programming and vector linear programming. *Mathematical Methods of Operations Research* 84 (2016), 411–426.
[34] Kenneth L McMillan. 1993. *Symbolic Model Checking*. Springer.
[35] R Tyrrell Rockafellar and Roger J-B Wets. 2009. *Variational Analysis*. Springer.
[36] Ilya Tkachev and Alessandro Abate. 2014. Characterization and computation of infinite-horizon specifications over Markov processes. *Theoretical Computer Science* 515 (2014), 1–18.
[37] Petter Tøndel, Tor Arne Johansen, and Alberto Bemporad. 2003. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* 39, 3 (2003), 489–497.
[38] M. Y. Vardi and L. Stockmeyer. 1985. Improved upper and lower bounds for modal logics of programs. In *ACM Symposium on Theory of Computing*. 240–251.
[39] Yinyu Ye and Edison Tse. 1989. An extension of Karmarkar's projective algorithm for convex quadratic programming. *Mathematical programming* 44 (1989), 157–179.

APPENDIX A: ADDITIONAL CONNECTIONS BETWEEN PCTL AND DISTRIBUTION-SPECIFIED CTL

We expand the connections between PCTL and distribution-specified CTL from reachability and safety in Section 3.3 to other properties. Let us begin with the 'next' operator $\bigcirc$. In the following proposition, we show that the distribution-specified CTL model checking can provide *necessary and sufficient* conditions for PCTL model checking.

Proposition 4. *Consider the MDP* $\mathsf{M} = (\mathbb{X}, \mathbb{U}, T, \mathcal{AP}_s, L_s)$ *and two PCTL formulae* $\mathsf{Pr}_{\sim p}(\bigcirc\Phi_1)$ *and* $\mathsf{Pr}_{\geq p}(\Phi_2 \vee \bigcirc\Phi_2)$*, where* Φ_1 *and* Φ_2 *are PCTL state formulae,* $\sim \in \{>, <, \geq, \leq, =\}$*, and* $p \in [0, 1]$*. Let the MDP-induced transition system be* $\mathsf{MTS} = (\mathcal{P}(\mathbb{X}), \mathcal{U}, \rightarrow, \mathcal{AP}_d, L_d)$*, where* $\mathcal{AP}_d = \{a_{d1}, a_{d2}\}$*, and*

$$\begin{cases} L_d^{-1}(a_{d1}) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \sum_{x \in \mathrm{Sat}(\Phi_1)} \pi(x) \sim p\}, \\ L_d^{-1}(a_{d2}) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \sum_{x \in \mathrm{Sat}(\Phi_2)} \pi(x) \geq p\}, \end{cases}$$

where $\mathrm{Sat}(\cdot)$ *denotes the satisfaction set over the state space. The the following statements hold:*

(1) $x_0 \vDash \mathsf{Pr}_{\sim p}(\bigcirc\Phi_1)$ *if and only if* $e_{x_0} \vDash \forall(\bigcirc a_{d1})$*;*
(2) $x_0 \vDash \mathsf{Pr}_{\geq p}(\Phi_2 \vee \bigcirc\Phi_2)$ *if and only if* $e_{x_0} \vDash \forall(a_{d2} \vee \bigcirc a_{d2})$*.*

Again, e_{x_0} *is a vector with the* x_0*-th element being* 1 *and all the others being* 0*.*

The proof of Proposition 4 is similar to that of Proposition 1. Next, we further specify the results in Proposition 1 to qualitative properties.

Proposition 5. *Consider the MDP* $\mathsf{M} = (\mathbb{X}, \mathbb{U}, T, \mathcal{AP}_s, L_s)$ *and two PCTL formulae* $\mathsf{Pr}_{=1}(\Diamond\Phi)$ *and* $\mathsf{Pr}_{=1}(\Box\Phi)$*, where* Φ *is a PCTL state formula. Let the MDP-induced transition system be* $\mathsf{MTS} = (\mathcal{P}(\mathbb{X}), \mathcal{U}, \rightarrow, \mathcal{AP}_d, L_d)$*, where* $\mathcal{AP}_d = \{a_d\}$*, and* $L_d^{-1}(a_d) = \{\pi \in \mathcal{P}(\mathbb{X}) \mid \sum_{x \in \mathrm{Sat}(\Phi)} \pi(x) = 1\}$*. The the following statements hold:*

(1) $x_0 \vDash \mathsf{Pr}_{=1}(\Diamond\Phi)$ *if* $e_{x_0} \vDash \forall(\Diamond a_d)$*;*
(2) $x_0 \vDash \mathsf{Pr}_{\geq 1}(\Box\Phi)$ *if and only if* $e_{x_0} \vDash \forall\Box a_d$*.*

The first statement in Proposition 5 implies that there exists a distribution-specified CTL formula $\forall\Diamond a_d$ that corresponds to the qualitative PCTL formula $\mathsf{Pr}_{=1}(\Diamond\Phi)$. However, it is known [7] that there exists no state-based CTL formula that is equivalent to this PCTL formula. On the other hand, the second statement provides a tighter connection in the qualitative safety than Proposition 1. In addition, we remark that both Propositions 4 and 5 can be further restated to connect the PCTL synthesis and existential-quantified CTL model checking, like Corollary 1.

APPENDIX B: MDP MODEL OF A PHARMACOKINETICS SYSTEM

We consider the MDP model of a pharmacokinetics system given in [11, 26], which consists of five states: plasma (Pl), interstitial fluid (IF), utilisation and degradation (Ut), drug being injected (Dr), the drug being cleared (Cl). An additional "dummy" state (Re) allows to adjust the amount of drug being injected initially. As a slight deviation from the MDP semantics in our work, the MDP model of the pharmacokinetics system is governed by two stochastic matrices P_{normal} and $P_{\text{saturated}}$:

$$P_{\text{normal}} = \begin{bmatrix} 0.94000 & 0.02634 & 0.02564 & 0.00780 & 0.00024 & 0 \\ 0 & 0.20724 & 0.48298 & 0.29624 & 0.01354 & 0 \\ 0 & 0.15531 & 0.42539 & 0.39530 & 0.02400 & 0 \\ 0 & 0.02598 & 0.10778 & 0.77854 & 0.0877 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$P_{\text{saturated}} = \begin{bmatrix} 0.9400 & 0.02425 & 0.02558 & 0.00809 & 0.00012 & 0 \\ 0 & 0.20728 & 0.48329 & 0.30257 & 0.00686 & 0 \\ 0 & 0.15540 & 0.42612 & 0.40627 & 0.01221 & 0 \\ 0 & 0.02653 & 0.11080 & 0.81776 & 0.04491 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

Further introduce the set of all possible matrices in the convex combination of P_{normal} and $P_{\text{saturated}}$, denoted by $\mathcal{S} = \{P \mid P = \lambda P_{\text{normal}} + (1-\lambda)P_{\text{saturated}}, \lambda \in [0,1]\}$. The MDP selects non-deterministically any matrix within set $\mathcal{S}$: in other words, given an initial distribution π_0, the dynamics are $\pi_{k+1} = \pi_k P_k$ for $k > 0$, where matrices can be selected as $P_k \in \mathcal{S}$ at any time index k. We note that the finite-action MDP model in this work is more general than the MDP from [11, 26]: indeed, the MDP semantics in [11, 26] and in this example can be expressed by associating the finite set of stochastic matrices to (two) actions, and thus to deterministic policies. As such, randomised policies allow the selection of matrices $P_k \in \mathcal{S}$.

The initial distribution is defined by $\pi_0(\text{Dr}) = \alpha$, $\pi_0(\text{Re}) = 1-\alpha$, and $\pi_0(x) = 0$ for $x \in \{\text{Pl}, \text{IF}, \text{CI}, \text{Ut}\}$, where α is interpreted as the amount of drug being injected initially. Following [11, 26], we further set thresholds MEC = 0.13 and MTC = 0.20, and consider the set of atomic propositions $\mathcal{AP}_d = \{\texttt{effective}, \texttt{nontoxic}, \texttt{cleared}\}$. Considering $\pi_k(\text{Ut})$, namely the probability of the drug being in the compartment Ut at time k, we define label effective as $\pi_k(\text{Ut}) \geq \text{MEC}$; nontoxic as $\pi_k(\text{Ut}) \geq \text{MTC}$; and cleared as $\pi_k(\text{Cl}) \leq \epsilon$ for some given (small) value $\epsilon > 0$. The distribution-specified CTL formula of interest is $\Phi = \Phi_1 \wedge \Phi_2 \wedge \Phi_3$ with $\Phi_1 = \forall\Box\texttt{nontoxic}$, $\Phi_2 = \forall\Diamond(\texttt{effective} \wedge \forall \bigcirc \texttt{effective})$, $\Phi_3 = \forall\Diamond\texttt{cleared}$. Here, Φ_1 encodes the requirement that the drug level always stays in in the safe zone; Φ_2 stipulates the drug is eventually effective for at least two consecutive steps; and Φ_3 specifies that the drug ought to be eventually cleared.

APPENDIX C: UAV PATH PLANNING

As shown in Fig. 4, a UAV roams a 5×5 "slippery grid world" and has five possible actions {*up, down, left, right, stay*}. Due to environmental uncertainties (e.g., noise), we assume that the first four actions will move the state to the desired next configuration with probability 0.95, and to other neighboring states with likelihood $\frac{0.05}{N_{neigh}}$, where N_{neigh} is the number of available/feasible neighboring states (not including the desired state): we say that a state (x_1, y_1) is a neighboring state of a state (x_2, y_2) if $\max\{|x_1 - x_2|, |y_1 - y_2|\} \leq 1$. The movements of the UAV can be modeled as an MDP M. The state space $\mathbb{X}$ corresponds to the set comprising the 25 squares in the grid world, whereas the action space $\mathbb{U}$ is {*up, down, left, right, stay*}. The transition probability T is defined according to the description above. Let $\mathcal{AP}_s = \{a_{\text{unsafe}}, a_{\text{target}}\}$ be the set of atomic proposition, and L_s a labeling function, such that $L_s^{-1}(a_{\text{unsafe}}) = \text{Obs}$ and $L_s^{-1}(a_{\text{target}}) = \text{Target}$. The motion planning problem is to find a policy such that the PCTL formula $\Phi_{\text{PCTL}} = \mathsf{Pr}_{\geq p}(\neg a_{\text{unsafe}} \mathsf{U} a_{\text{target}})$ is fulfilled, where $p \in (0, 1]$ is the same parameter to define b^p_{target} in Section 6.

If parameters β and p are set to be 0 and 0.8 respectively, we find that no policy exists such that the PCTL formula Φ_{PCTL} is feasible. This entails that the distribution π_0 does not satisfy the CTL formula Φ_{CTL} as per Corollary 1.