Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties

Alberto Puggelli, Wenchao Li, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia

Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
{puggelli, wenchaol, alberto, sseshia}@eecs.berkeley.edu

Abstract. We address the problem of verifying Probabilistic Computation Tree Logic (PCTL) properties of Markov Decision Processes (MDPs) whose state transition probabilities are only known to lie within uncertainty sets. We first introduce the model of Convex-MDPs (CMDPs), i.e., MDPs with convex uncertainty sets. CMDPs generalize Interval-MDPs (IMDPs) by allowing also more expressive (convex) descriptions of uncertainty. Using results on strong duality for convex programs, we then present a PCTL verification algorithm for CMDPs, and prove that it runs in time polynomial in the size of a CMDP for a rich subclass of convex uncertainty models. This result allows us to lower the previously known algorithmic complexity upper bound for IMDPs from co-NP to PTIME. We demonstrate the practical effectiveness of the proposed approach by verifying a consensus protocol and a dynamic configuration protocol for IPv4 addresses.

1 Introduction

Stochastic models such as Discrete-Time Markov Chains (DTMCs) [1] and Markov Decision Processes (MDPs) [2] are used to formally represent systems that exhibit probabilistic behaviors. These systems need *quantitative* analysis [3] to answer questions such as "what is the probability that a request will be eventually served?". Properties of these systems can be expressed and analyzed using logics such as Probabilistic Computation Tree Logic (PCTL) [4] — a probabilistic logic derived from CTL — as well as techniques for probabilistic model checking [5]. These methods often rely on deriving a probabilistic model of the underlying process, hence the formal guarantees they provide are only as good as the estimated model. In a real setting, these estimations are affected by uncertainties due, for example, to measurement errors or approximation of the real system by mathematical models.

Interval-valued Discrete-Time Markov Chains (IDTMCs) have been introduced to capture modeling uncertainties [6]. IDTMCs are DTMC models where each transition probability lies within a close interval. Two semantic interpretations have been proposed for IDTMCs [7]: Uncertain Markov Chains (UMCs) and Interval Markov Decision Processes (IMDPs). An UMC is interpreted as a family of DTMCs, where each member is a DTMC whose transition probabilities lie within the interval range given in the UMC. In IMDPs, the uncertainty is resolved through non-determinism. Each time a state is visited, a transition distribution within the interval is adversarially picked, and a probabilistic step is taken accordingly. Thus, IMDPs model a non-deterministic choice made

from a set of (possibly uncountably many) choices. In this paper we do not consider UMCs and focus on IMDPs.

An upper-bound on the complexity of model checking PCTL properties on IMDPs was previously shown to be co-NP [8]. This result relies on the construction of an equivalent MDP that encodes all behaviors of the IMDP. For each state in the new MDP, the set of transition probabilities is equal to the Basic Feasible Solutions (BFS) of the set of inequalities specifying the transition probabilities of the IMDP. Since the number of BFS is exponential in the number of states in the IMDP, the equivalent MDP can have size exponential in the size of the IMDP. In this paper, we describe a *polynomial-time algorithm* (in both size of the model and size of the formula) based on Convex Programming (CP) for the same fragment of PCTL considered in [7, 8] (the *Bounded Until* operator is disallowed). This shows that the problem is in the complexity class PTIME. With *Bounded Until*, the time complexity of our algorithm only increases to pseudo-polynomial in the maximum integer time bound.

An interval model of uncertainty may appear to be the most intuitive. However, there are significant advantages in accommodating also more expressive (and less pessimistic) uncertainty models. In [9], a financial portfolio optimization case-study is analyzed in which uncertainty arises from estimating the asset return rates. The authors claim that the interval model is too conservative in this scenario, because it would suggest to invest the whole capital into the asset with the smallest worst-case return. The ellipsoidal model proposed in that paper returns instead the more profitable strategy of spreading the capital across multiple assets. Further, depending on the field, researchers use different models to represent uncertainty. Maximum likelihood models are often used, for example, to estimate chemical reaction parameters [10]. To increase modeling expressiveness, we introduce the model of *Convex-MDP (CMDP)*, i.e., an MDP whose state transition probabilities are only known to lie within convex uncertainty sets. The proposed algorithms can be extended to verify CMDPs for all the models of uncertainty that satisfy a technical condition introduced later in the paper, while maintaining the same complexity results proven for IMDPs. This condition is not a limitation in practical scenarios, and we show that all the models in the wide and relevant class of convex uncertainty sets introduced in [11] (e.g. interval, ellipsoidal and likelihood models) satisfy it. Heterogeneous models of uncertainty can then be used within the same CMDP to represent different sources of uncertainty. We also note that the complexity results presented in [7] and [8] cannot be trivially extended to verifying CMDPs. This is because BFS are not defined for generic convex inequalities, so the construction of an equivalent MDP would not be possible. The complexity results are compared in Table 1.

To summarize, the contributions of this paper are as follows.

1. We give a polynomial-time algorithm for model checking PCTL properties (without *Bounded Until*) on IMDPs. This improves the co-NP result in [8] to PTIME.
2. We extend the algorithm to full PCTL and show that its time complexity becomes pseudo-polynomial in the maximum integer bound in *Bounded Until*.
3. We show that our complexity results extend to Convex-MDPs (CMDPs) for a wide and expressive subclass of the convex models of uncertainty.

Table 1: Known Upper-Bound on the Complexity of PCTL Model Checking.

Model	DTMC [4]	IMDP [8]	IMDP/CMDP [ours]
Complexity	PTIME	co-NP	PTIME

4. We demonstrate the relevance of our approach with case studies, where a small uncertainty in the probability transitions indeed yields a significant change in the verification results.

An extended version of the paper with details of all verification algorithms and proofs of correctness is available [12].

The paper is organized as follows. Section 2 gives background on MDPs, PCTL, and the analyzed uncertainty models. Section 3 presents related work. Section 4 gives an overview of the proposed approach. In Section 5, we describe the proposed algorithm in detail and prove the PTIME complexity result. Section 6 describes two case studies, and we conclude and discuss future directions in Section 7.

2 Preliminaries

Definition 2.1. *A Probability Distribution (PD) over a finite set Z of cardinality n is a vector $\mu \in \mathbb{R}^n$ satisfying $\mathbf{0} \leq \mu \leq \mathbf{1}$ and $\mathbf{1}^T\mu = 1$. The element $\mu[i]$ represents the probability of realization of event z_i. We call $Dist(Z)$ the set of distributions over Z.*

2.1 Convex Markov Decision Process (CMDP)

Definition 2.2. *A CMDP is a tuple $\mathcal{M}_\mathcal{C} = (S, S_0, A, \Omega, \mathcal{F}, \mathcal{A}, \mathcal{X}, L)$, where S is a finite set of states of cardinality $N = |S|$, S_0 is the set of initial states, A is a finite set of actions ($M = |A|$), Ω is a finite set of atomic propositions, $\mathcal{F}$ is a finite set of convex sets of transition PDs, $\mathcal{A} : S \to 2^A$ is a function that maps each state to the set of actions available at that state, $\mathcal{X} = S \times A \to \mathcal{F}$ is a function that associates to state s and action a the corresponding convex set $\mathcal{F}_s^a \in \mathcal{F}$ of transition PDs, and $L : S \to 2^\Omega$ is a labeling function.*

The set $\mathcal{F}_s^a = Dist_s^a(S)$ represents the uncertainty in defining a transition distribution for $\mathcal{M}_\mathcal{C}$ given state s and action a. We call $\mathbf{f}_s^a \in \mathcal{F}_s^a$ an observation of this uncertainty. Also, $\mathbf{f}_s^a \in \mathbb{R}^N$ and we can collect the vectors $\mathbf{f}_s^a, \forall s \in S$ into an observed transition matrix $F^a \in \mathbb{R}^{N \times N}$. Abusing terminology, we call $\mathcal{F}^a$ the uncertainty set of the transition matrices, and $F^a \in \mathcal{F}^a$. $\mathcal{F}_s^a$ is interpreted as the row of $\mathcal{F}^a$ corresponding to state s. Finally, $f_{s_i s_j}^a = \mathbf{f}_{s_i}^a[j]$ is the observed probability of transitioning from s_i to s_j when action a is selected.

A transition between state s to state s' in a CMDP occurs in three steps. First, an action $a \in \mathcal{A}(s)$ is chosen. The selection of a is nondeterministic. Secondly, an observed PD $\mathbf{f}_s^a \in \mathcal{F}_s^a$ is chosen. The selection of $\mathbf{f}_s^a$ models uncertainty in the transition. Lastly, a successor state s' is chosen randomly, according to the transition PD $\mathbf{f}_s^a$.

A path π in $\mathcal{M}_\mathcal{C}$ is a finite or infinite sequence of the form $s_0 \xrightarrow{f_{s_0 s_1}^{a_0}} s_1 \xrightarrow{f_{s_1 s_2}^{a_1}}, \cdots$, where $s_i \in S$, $a_i \in \mathcal{A}(s_i)$ and $f_{s_i, s_{i+1}}^{a_i} > 0 \ \forall i \geq 0$. We indicate with Π_{fin} (Π_{inf}) the

set of all finite (infinite) paths of $\mathcal{M}_\mathcal{C}$. $\pi[i]$ is the i^{th} state along the path and, for finite paths, $last(\pi)$ is the last state visited in $\pi \in \Pi_{fin}$. $\Pi_s = \{\pi \mid \pi[0] = s\}$ is the set of paths starting in state s.

To model uncertainty in state transitions, we make the following assumptions:

Assumption 2.1. $\mathcal{F}^a$ can be factored as the Cartesian product of its rows, i.e., its rows are uncorrelated. Formally, for every $a \in A$, $\mathcal{F}^a = \mathcal{F}^a_{s_0} \times \cdots \times \mathcal{F}^a_{s_{N-1}}$. In [11] this assumption is referred to as *rectangular uncertainty*.

Assumption 2.2. If the probability of a transition is zero (non-zero) for at least one PD in the uncertainty set, then it is zero (non-zero) for all PDs.
Formally, $\exists \mathbf{f}^a_s \in \mathcal{F}^a_s : f^a_{ss'} = (\neq)0 \implies \forall \mathbf{f}^a_s \in \mathcal{F}^a_s : f^a_{ss'} = (\neq)0$.
The assumption guarantees the correctness of the preprocessing verification routines used later in the paper, which rely on state reachability of the MDP underlying graph.

We determine the size $\mathcal{R}$ of the CMDP $\mathcal{M}_\mathcal{C}$ as follows. $\mathcal{M}_\mathcal{C}$ has N states, $O(M)$ actions per state and $O(N^2)$ transitions for each action. Let D^a_s denote the number of constraints required to express the rectangular uncertainty set $\mathcal{F}^a_s$ (e.g. $D^a_s = O(2N)$ for the interval model, to express the upper and lower bounds of the transition probabilities from state s to all states $s' \in S$), and $D = \max_{s \in S, a \in A} D^a_s$. The overall size of $\mathcal{M}_\mathcal{C}$ is thus $\mathcal{R} = O(N^2M + NMD)$.

In order to analyze *quantitative* properties of CMDPs, we need a probability space over infinite paths [13]. However, a probability space can only be constructed once nondeterminism and uncertainty have been resolved. We call each possible resolution of nondeterminism an *adversary*, which chooses an action in each state of $\mathcal{M}_\mathcal{C}$.

Definition 2.3. Adversary. *A randomized adversary for $\mathcal{M}_\mathcal{C}$ is a function $\alpha = \Pi_{fin} \times A \to [0,1]$, with $\sum_{\mathcal{A}(last(\pi))} \alpha(\pi, a) = 1$, and $a \in \mathcal{A}(last(\pi))$ if $\alpha(\pi, a) > 0$. We call Adv the set of all adversaries α of $\mathcal{M}_\mathcal{C}$.*

Conversely, we call a *nature* each possible resolution of uncertainty, i.e., a nature chooses a transition PD for each state and action of $\mathcal{M}_\mathcal{C}$.

Definition 2.4. Nature. *Given action $a \in A$, a randomized nature is the function $\eta^a : \Pi_{fin} \times Dist(S) \to [0,1]$ with $\int_{\mathcal{F}^a_{last(\pi)}} \eta^a(\pi, \mathbf{f}^a_s) = 1$, and $\mathbf{f}^a_s \in \mathcal{F}^a_{last(\pi)}$ if $\eta^a(\pi, \mathbf{f}^a_s) > 0$. We call Nat the set of all natures η^a of $\mathcal{M}_\mathcal{C}$.*

An adversary α (nature η^a) is memoryless if it depends only on $last(\pi)$. Also, α (η^a) is deterministic if $\alpha(\pi, a) = 1$ for some $a \in \mathcal{A}(last(\pi))$ ($\eta^a(\pi, f^a_s) = 1$ for some $\mathbf{f}^a_s \in \mathcal{F}^a_{last(\pi)}$).

2.2 Models of Uncertainty

We only consider CMDPs whose transition PDs lie in uncertainty sets that satisfy Assumption 5.1 (introduced later for ease of presentation). This assumption holds for all the uncertainty models analyzed in [11]. We report results for the interval, likelihood and ellipsoidal models. Results for the entropy model are available in [12].

Interval Model. Intervals commonly describe uncertainty in transition matrices:

$$\mathcal{F}_s^a = \{\mathbf{f}_s^a \in \mathbb{R}^N \mid \mathbf{0} \leq \underline{\mathbf{f}}_s^a \leq \mathbf{f}_s^a \leq \overline{\mathbf{f}}_s^a \leq \mathbf{1}, \mathbf{1}^T\mathbf{f}_s^a = 1\} \tag{1}$$

where $\underline{\mathbf{f}}_s^a, \overline{\mathbf{f}}_s^a \in \mathbb{R}^N$ are the element-wise lower and upper bounds of $\mathbf{f}$. This model is suitable when the matrix components are individually estimated by statistical data.

Likelihood Model. This model is appropriate when the transition probabilities are determined experimentally. The transition frequencies associated to action $a \in A$ are collected in matrix H^a. Uncertainty in each row of H^a can be described by the likelihood region [14]:

$$\mathcal{F}_s^a = \{\mathbf{f}_s^a \in \mathbb{R}^N \mid \mathbf{f}_s^a \geq \mathbf{0}, \mathbf{1}^T\mathbf{f}_s^a = 1, \textstyle\sum_{s'} h_{ss'}^a \log(f_{ss'}^a) \geq \beta_s^a\} \tag{2}$$

where $\beta_s^a < \beta_{s,max}^a = \sum_{s'} h_{ss'}^a \log(h_{ss'}^a)$ represents the uncertainty level. Likelihood regions are less conservative uncertainty representations than intervals, which arise from projections of the uncertainty region onto each row component.

Ellipsoidal Model. Ellipsoidal models can be seen as a second-order approximation of the likelihood model [11]. Formally:

$$\mathcal{F}_s^a = \{\mathbf{f}_s^a \in \mathbb{R}^N \mid \mathbf{f}_s^a \geq \mathbf{0}, \mathbf{1}^T\mathbf{f}_s^a = 1, \|R_s^a\,(\mathbf{f}_s^a - \mathbf{h}_s^a)\,\|_2 \leq 1, R_s^a \succ 0\} \tag{3}$$

where matrix R_s^a represents an ellipsoidal approximation of the likelihood Region (2).

Remark 2.1. Each set $\mathcal{F}_s^a$ within the same CMDP can be expressed with a different uncertainty model to represent different sources of uncertainty.

2.3 Probabilistic Computation Tree Logic (PCTL)

We use PCTL, a probabilistic logic derived from CTL which includes a probabilistic operator P [4], to express properties of CMDPs. The syntax of this logic is:

$$\begin{aligned} \phi &::= True \mid \omega \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid P_{\bowtie p}\,[\psi] && \text{state formulas} \\ \psi &::= \mathcal{X}\phi \mid \phi_1\,\mathcal{U}^{\leq k}\phi_2 \mid \phi_1\,\mathcal{U}\phi_2 && \text{path formulas} \end{aligned}$$

where $\omega \in \Omega$ is an atomic proposition, $\bowtie \in \{\leq, <, \geq, >\}$, $p \in [0,1]$ and $k \in \mathbb{N}$.

Table 2: PCTL semantics for CMDP

$s \models$ True		$s \models \omega$ iff $\omega \in L(s)$
$s \models \neg\phi$	iff $s \not\models \phi$	$s \models \phi_1 \wedge \phi_2$ iff $s \models \phi_1 \wedge s \models \phi_2$
$s \models P_{\bowtie p}\,[\psi]$	iff $Prob\,(\{\pi \in \Pi_s(\alpha, \eta^a) \mid \pi \models \psi\}) \bowtie p$	$\forall \alpha \in Adv$ and $\eta^a \in Nat$
$\pi \models \mathcal{X}\phi$	iff $\pi[1] \models \phi$	
$\pi \models \phi_1\,\mathcal{U}^{\leq k}\phi_2$	iff $\exists i \leq k \mid \pi[i] \models \phi_2 \wedge \forall j < i\ \pi[j] \models \phi_1$	
$\pi \models \phi_1\,\mathcal{U}\phi_2$	iff $\exists k \geq 0 \mid \pi \models \phi_1\,\mathcal{U}^{\leq k}\phi_2$	

Path formulas ψ use the *Next* $(\mathcal{X})$, *Bounded Until* $\left(\mathcal{U}^{\leq k}\right)$ and *Unbounded Until* $(\mathcal{U})$ operators. These formulas are evaluated over paths and only allowed as parameters to the $P_{\bowtie p}[\psi]$ operator. The size $\mathcal{Q}$ of a PCTL formula is defined as the number of Boolean connectives plus the number of temporal operators in the formula. For the *Bounded Until* operator, we denote separately the maximum time bound that appears in the formula as k_{max}. Probabilistic statements about MDPs typically involve universal quantification over adversaries $\alpha \in Adv$. With uncertainties, for each action a selected by adversary α, we will further quantify across nature $\eta^a \in Nat$ to compute the worst case condition within the action range of η^a, i.e., the uncertainty set $\mathcal{F}_s^a$. We define $P_s(\alpha, \eta^a)[\psi] \triangleq Prob(\{\pi \in \Pi_s(\alpha, \eta^a) \mid \pi \models \psi\})$ the probability of taking a path $\pi \in \Pi_s$ that satisfies ψ under adversary α and nature η^a. If α and η^a are Markov deterministic in state s, we write $P_s(a, \mathbf{f}_s^a)$, where a and $\mathbf{f}_s^a$ are the action and resolution of uncertainty that are deterministically chosen at each execution step by α and η^a. $P_s^{max}[\psi]$ ($P_s^{min}[\psi]$) denote the maximum (minimum) probability $P_s(\alpha, \eta^a)[\psi]$ across all adversaries $\alpha \in Adv$ and natures $\eta^a \in Nat$, and the vectors $\mathbf{P}^{max}[\psi], \mathbf{P}^{min}[\psi] \in \mathbb{R}^N$ collect these probabilities $\forall s \in S$. The semantics of the logic is reported in Table 2, where we write $\models$ instead of $\models_{Adv,Nat}$ for simplicity.

For ease of computation, we would like to consider only memoryless and deterministic adversaries and natures to compute *quantitative* probabilities, i.e., solve:

$$P_s^{max}[\psi] = \max_{a \in \mathcal{A}(s)} \max_{\mathbf{f}_s^a \in \mathcal{F}_s^a} P_s(a, \mathbf{f}_s^a)[\psi] \qquad P_s^{min}[\psi] = \min_{a \in \mathcal{A}(s)} \min_{\mathbf{f}_s^a \in \mathcal{F}_s^a} P_s(a, \mathbf{f}_s^a)[\psi] \quad (4)$$

We extend a result from [15] to prove that this is possible (see [12] for the proof).

Proposition 2.1. *Given a CMDP $\mathcal{M}_\mathcal{C}$ and a target state $s_t \in S$, there always exist deterministic and memoryless adversaries and natures for $\mathcal{M}_\mathcal{C}$ that achieve the maximum (minimum) probabilities of reaching s_t, if A is finite and the inner optimization in Problem (4) always attains its optimum $\sigma_s^*(a)$ over the sets $\mathcal{F}_s^a, \forall s \in S, \forall a \in \mathcal{A}(s)$, i.e., there exists a finite feasible $\mathbf{f}_s^a \in \mathcal{F}_s^a$ such that $P_s(a, \mathbf{f}_s^a)[\psi] = \sigma_s^*(a)$.*

The verification algorithm V determines whether a state $s \in S_0$ is (is not) contained in the set $Sat(\phi) = \{s \in S \mid s \models \phi\}$. We define the following properties for V:

Definition 2.5. Soundness *(*Completeness*). Algorithm V is sound (complete) if:*

$$s \in Sat_V(\phi) \Rightarrow s \in Sat(\phi) \qquad (s \notin Sat_V(\phi) \Rightarrow s \notin Sat(\phi))$$

where $Sat_V(\phi)$ ($Sat(\phi)$) is the computed (actual) satisfaction set.

Algorithms to verify non-probabilistic formulas are sound and complete, because they are based on reachability analysis over the finite number of states of $\mathcal{M}_\mathcal{C}$ [16]. Conversely, we will show in Section 5 that algorithms to verify probabilistic formulas $\phi = P_{\bowtie p}[\psi]$ in the presence of uncertainties require to solve convex optimization problems over the set $\mathbb{R}$ of the real numbers. Optima of these problems can be arbitrary real numbers, so, in general, they can be computed only to within a desired accuracy ϵ_d. We consider an algorithm to be sound and complete if the error in determining the satisfaction probabilities of ϕ is bounded by such a parameter ϵ_d, since the returned result will still be accurate enough in most settings.

3 Related Work

Probabilistic model checking tools such as PRISM [5] have been used to analyze a multitude of applications, from communication protocols and biological pathways to security problems. In this paper, we further consider *uncertainties* in the probabilistic transitions of the MDP for model checking PCTL specifications. Prior work [6–8, 17] in similar verification problems also dealt with uncertainties in the probabilistic transitions. However, they considered only interval models of uncertainty, while we incorporate more expressive models such as ellipsoidal and likelihood. Further, we consider nature as adversarial and study how it affects the MDP execution in the worst case. The developers of PARAM [18] consider instead uncertainties as possible values that parameters in the model can take, and synthesize the optimal parameter values to maximize the satisfaction probability of a given PCTL specification.

We improve the previously best-known complexity result of co-NP in [8] to PTIME, for the fragment of PCTL without $\mathcal{U}^{\leq k}$. For the full PCTL syntax, our algorithm runs in $O(poly(\mathcal{R}) \times \mathcal{Q} \times k_{max})$ time, where k_{max} is the maximum bound in $\mathcal{U}^{\leq k}$. This result is pseudo-polynomial in k_{max}, i.e., polynomial (exponential) if k_{max} is counted in its unary (binary) representation. Conversely, classical PCTL model checking for DTMCs [4] runs in time polynomial in k_{max} counted in its binary representation. The difference stems from the computation of the set $Sat\left(P_{\bowtie p}\left[\phi_1\, \mathcal{U}^{\leq k} \phi_2\right]\right)$. For (certain) MDPs, this computation involves raising the transition matrices $F^a, \forall a \in A$ to the k^{th} power, to model the evolution of the system in k steps. With uncertainties, we cannot do matrix exponentiation, because $F^a \in \mathcal{F}^a$ might change at each step. However, both $\mathcal{Q}$ and k_{max} are typically small in practical applications [19], so the dominant factor for runtime is the size of the model $\mathcal{R}$. We note that the complexity results of [7] and [8] can be extended to the PCTL with $\mathcal{U}^{\leq k}$.

The convex uncertainty models [11] analyzed in this paper have been considered recently in the robust control literature. In [20], an algorithm is given to synthesize a robust optimal controller for an MDP to satisfy a Linear Temporal Logic (LTL) specification where only one probabilistic operator is allowed. Their technique first converts the LTL specification to a Rabin automaton (which is worst-case doubly exponential in the size of the LTL formula), and composes it with the MDP. Robust dynamic programming is then used to solve for the optimal control policy. We consider PCTL, which allows nested probability operators, and propose an algorithm which is polynomial both in the size of the model and of the formula.

The robustness of PCTL model checking has been analyzed [21] based on the notion of an Approximate Probabilistic Bisimulation (APB) tailored to the finite-precision approximation of a numerical model. We instead verify MDPs whose transition probabilities are affected by uncertainties due to estimation errors or imperfect information about the environment.

4 Probabilistic Model Checking with Uncertainties

We define the problem under analysis, and overview the proposed approach to solve it.

PCTL model checking with uncertainties. **Given** a Markov Decision Process model with convex uncertainties $\mathcal{M}_{\mathcal{C}}$ of size $\mathcal{R}$ and a PCTL formula ϕ of size $\mathcal{Q}$ over a set of atomic propositions Ω, **verify** ϕ over the uncertainty sets $\mathcal{F}_s^a \in \mathcal{F}$ of $\mathcal{M}_{\mathcal{C}}$.

As in verification of CTL [22], the algorithm traverses bottom-up the parse tree for ϕ, recursively computing the set $Sat(\phi')$ of states satisfying each sub-formula ϕ'. At the end of the traversal, the algorithm computes the set of states satisfying ϕ and it determines if $s \models \phi$ by checking if $s \in Sat\,(\phi)$. For the non-probabilistic PCTL operators, the satisfying states are computed as: $Sat\,(True) = S$, $Sat(\omega) = \{s \in S \mid \omega \in L(s)\}$, $Sat(\neg\phi) = S \setminus Sat(\phi)$ and $Sat(\phi_1 \wedge \phi_2) = Sat(\phi_1) \cap Sat(\phi_2)$. For the probabilistic operator $P \bowtie [\psi]$, we compute:

$$Sat\,(P_{\lhd p}\,[\psi]) = \{s \in S \mid P_s^{max}(\psi) \lhd p\}\,, \quad Sat\,(P_{\rhd p}\,[\psi]) = \{s \in S \mid P_s^{min}(\psi) \rhd p\} \quad (5)$$

We propose polynomial-time routines to compute Sets 5 for MDPs whose transition matrices F^a are only known to lie within convex uncertainty sets $\mathcal{F}^a$, $\forall a \in A$.

Using Proposition 2.1, the proposed routines encode the transitions of $\mathcal{M}_{\mathcal{C}}$ under the sets of deterministic and memoryless adversaries and natures into convex programs and solve them. From the returned solution, it is then possible to determine the *quantitative* satisfaction probabilities $P_s^{max}[\psi]$ (or $P_s^{min}[\psi]$) $\forall s \in S$, which get compared in linear time to the threshold p to compute the set $Sat\,(P_{\bowtie p}\,[\psi])$. To prove the polynomial-time complexity of the model-checking algorithm, we use the following key result from convex theory [23].

Proposition 4.1. *Given the convex program:*

$$\begin{aligned} &\min_{\mathbf{x}} \; f_0(\mathbf{x}) \\ &\text{s.t. } f_i(\mathbf{x}) \leq 0 \qquad\qquad i = 1, \cdots, m \end{aligned}$$

with $\mathbf{x} \in \mathbb{R}^n$ and $f_i, i = 0, \cdots, m$ convex functions, the optimum σ^* can be found to within $\pm\epsilon_d$ in time complexity polynomial in the problem size (n, m) and $\log(1/\epsilon_d)$.

We are now ready to state the main contribution of this paper:

Theorem 4.1. Complexity of PCTL model-checking for CMDPs.

1. *The problem of verifying if a CMDP $\mathcal{M}_{\mathcal{C}}$ of size $\mathcal{R}$ satisfies a PCTL formula ϕ without $\mathcal{U}^{\leq k}$ is in PTIME.*
2. *A formula ϕ' with $\mathcal{U}^{\leq k}$ can be verified with time complexity $O\,(poly(\mathcal{R}) \times \mathcal{Q}' \times k_{max})$, i.e., pseudo-polynomial in the maximum time bound k_{max} of $\mathcal{U}^{\leq k}$.*

Sketch of proof. The proof is constructive. Our verification algorithm parses ϕ in time linear in the size $\mathcal{Q}$ of ϕ [22], computing the satisfiability set of each operator in ϕ. For the non-probabilistic operators, satisfiability sets can be computed in time polynomial in $\mathcal{R}$ using set operations, i.e., set inclusion, complementation and intersection. For the probabilistic operator, we leverage Proposition 4.1 and prove that the proposed verification routines: 1) solve a number of convex problems polynomial in $\mathcal{R}$; 2) generate these convex programs in time polynomial in $\mathcal{R}$. It thus follows that the overall algorithm runs in time polynomial in $\mathcal{R}$ and in the size of ϕ. The correctness and time-complexity for formulas involving the *Unbounded Until* operator are formalized in Lemma 5.1. Results regarding the *Next* and *Bounded Until* operator can be found in [12]. □

5 Verification Routines

We detail the routine to verify the *Unbounded Until* operator. Routines to verify the *Next* and *Bounded Until* operators can be found in the extended version [12].

5.1 *Unbounded Until* Operator

We verify $\phi = P_{\triangleleft p}[\phi_1 \mathcal{U} \phi_2]$ on a CMDP of size $\mathcal{R}$. First, the sets $S^{yes} \triangleq Sat\left(P_{\geq 1}[\phi_1 \mathcal{U} \phi_2]\right)$, $S^{no} \triangleq Sat\left(P_{\leq 0}[\phi_1 \mathcal{U} \phi_2]\right)$ and $S^{?} = S \setminus (S^{no} \cup S^{yes})$ are precomputed in time polynomial in $\mathcal{R}$ using reachability routines over the CMDP underlying graph [16]. Second, Equation (4) is evaluated for all $s \in S$ using the Convex Programming procedure described next. Finally, the computed probabilities are compared to p.

Convex Programming Procedure (CP). We start from the classical LP formulation to solve the problem without the presence of uncertainty [16]:

$$\begin{aligned} &\min_{\mathbf{x}} \mathbf{x^T 1} \\ &\text{s.t. } x_s = 0;\ x_s = 1; && \forall s \in S^{no};\ s \in S^{yes}; \\ &\quad x_s \geq \mathbf{x^T f_s^a} && \forall s \in S^{?}, \forall a \in \mathcal{A}(s) \end{aligned} \tag{6}$$

where $\mathbf{P}^{max}[\phi_1 \mathcal{U} \phi_2] = \mathbf{x}^*$ is computed solving only one LP. Problem (6) has N unknowns and $N - Q + MQ$ constraints, where $Q = |S^{?}| = O(N)$, so its size is polynomial in $\mathcal{R}$.

Proposition 2.1 allows us to rewrite Problem (6) in the uncertain scenario as:

$$\begin{aligned} &\min_{\mathbf{x}} \mathbf{x^T 1} \\ &\text{s.t. } x_s = 0;\ x_s = 1; && \forall s \in S^{no};\ \forall s \in S^{yes}; \\ &\quad x_s \geq \max_{\mathbf{f_s^a} \in \mathcal{F}_s^a} \left(\mathbf{x^T f_s^a}\right) && \forall s \in S^{?}, \forall a \in \mathcal{A}(s) \end{aligned} \tag{7}$$

i.e., we maximize the lower bound on x_s across the nature action range. The decision variable of the inner problem is $\mathbf{f}_s^a$ and its optimal value $\sigma^*(\mathbf{x})$ is parameterized in the outer problem decision variable $\mathbf{x}$. Problem (7) can be written in convex form for an arbitrary uncertainty model by replacing the last constraint with one constraint for each point in $\mathcal{F}_s^a$. However, this approach results in infinite constraints if the set $\mathcal{F}_s^a$ contains infinitely many points, as in the cases considered in the paper. We solve this difficulty using duality, which allows to rewrite Problem (7) with a number of constraints polynomial in $\mathcal{R}$. We start by replacing the primal inner problem in the outer Problem (7) with its dual $\forall s \in S^{?}$ and $\forall a \in \mathcal{A}(s)$:

$$\sigma_s^a(\mathbf{x}) = \max_{\mathbf{f}_s^a \in \mathcal{F}_s^a} \mathbf{x}^T \mathbf{f}_s^a \qquad \Rightarrow \qquad d_s^a(\mathbf{x}) = \min_{\lambda_\mathbf{s}^\mathbf{a} \in \mathcal{D}_s^a} g(\lambda_\mathbf{s}^\mathbf{a}, \mathbf{x}) \tag{8}$$

where $\lambda_\mathbf{s}^\mathbf{a}$ is the (vector) Lagrange multiplier and $\mathcal{D}_s^a$ is the feasibility set of the dual. In the dual, the decision variable is $\lambda_\mathbf{s}^\mathbf{a}$ and its optimal value $d_s^a(\mathbf{x})$ is parameterized in $\mathbf{x}$. The dual function $g(\lambda_s^a, \mathbf{x})$ and the set $\mathcal{D}_s^a$ are convex by construction in λ_s^a for arbitrary uncertainty models, so the dual is convex. Further, since also the primal is convex, strong duality holds, i.e., $\sigma_s^a = d_s^a$, $\forall \mathbf{x} \in \mathbb{R}^N$, because the primal satisfies Slater's condition [24] for any non-trivial uncertainty set $\mathcal{F}_s^a$. Any dual solution overestimates

the primal solution. When substituting the primals with the duals in Problem (7), we drop the inner optimization operators because the outer optimization operator will find the least overestimates, i.e., the dual solutions $d_s^a, \forall s \in S, a \in \mathcal{A}(s)$, to minimize its cost function. We get the CP formulation:

$$\begin{aligned}
&\min_{\mathbf{x}} \mathbf{x^T 1} && \min_{\mathbf{x},\lambda} \mathbf{x^T 1} && \\
&\text{s.t. } x_s = 0;\ x_s = 1; && \text{s.t. } x_s = 0;\ x_s = 1; && \forall s \in S^{no}; \forall s \in S^{yes}; && \text{(9a)}\\
&\quad x_s \geq \min_{\lambda_s^a \in \mathcal{D}_s^a} g\left(\lambda_s^a, \mathbf{x}\right) && \Rightarrow \quad x_s \geq g\left(\lambda_s^a, \mathbf{x}\right); && \forall s \in S^?, \forall a \in \mathcal{A}(s); && \text{(9b)}\\
& && \quad \lambda_s^a \in \mathcal{D}_s^a && \forall s \in S^?, \forall a \in \mathcal{A}(s) && \text{(9c)}
\end{aligned}$$

The decision variables of Problem (9) are both $\mathbf{x}$ and λ_s^a, so the CP formulation is convex only if the dual function $g(\lambda_s^a, \mathbf{x})$ is jointly convex in λ_s^a and $\mathbf{x}$. While this condition cannot be guaranteed for arbitrary uncertainty models, we prove constructively that it holds for the ones considered in the paper. For example, for the interval model, Problem (9) reads:

$$\begin{aligned}
&\min_{\mathbf{x},\lambda_{\mathbf{s}}^{\mathbf{a}}} \mathbf{x^T 1} && \\
&\text{s.t. } x_s = 0;\ x_s = 1; && \forall s \in S^{no}; \forall s \in S^{yes}; && \\
&\quad x_s \geq \lambda_{1,s}^a - (\underline{\mathbf{f}}_a^s)^T \lambda_{\mathbf{2,s}}^{\mathbf{a}} + (\overline{\mathbf{f}}_a^s)^T \lambda_{\mathbf{3,s}}^{\mathbf{a}}; && \forall s \in S^?, \forall a \in \mathcal{A}(s); && \text{(10a)}\\
&\quad \mathbf{x} + \lambda_{\mathbf{2,s}}^{\mathbf{a}} - \lambda_{\mathbf{3,s}}^{\mathbf{a}} - \lambda_{1,s}^a \mathbf{1} = \mathbf{0}; && \forall s \in S^?, \forall a \in \mathcal{A}(s); && \text{(10b)}\\
&\quad \lambda_{\mathbf{2,s}}^{\mathbf{a}} \geq \mathbf{0},\ \lambda_{\mathbf{3,s}}^{\mathbf{a}} \geq \mathbf{0} && \forall s \in S^?, \forall a \in \mathcal{A}(s) && \text{(10c)}
\end{aligned}$$

which is an LP, so trivially jointly convex in $\mathbf{x}$ and λ_s^a. Analogously, Problem (9) for the ellipsoidal model is a Second-Order Cone Program (SOCP), so again jointly convex in $\mathbf{x}$ and λ_s^a [12]. For the likelihood model, Constraints (10a-10c) become:

$$\begin{aligned}
&x_s \geq \lambda_{1,s}^a - (1+\beta_s^a)\lambda_{2,s}^a + \lambda_{2,s}^a \textstyle\sum_{s'} h_{ss'}^a \log\left(\frac{\lambda_{2,s}^a h_{ss'}^a}{\lambda_{1,s}^a - x_{s'}}\right); && \forall s \in S^?, \forall a \in \mathcal{A}(s); && \text{(11a)}\\
&\lambda_{1,s}^a \geq \max_{s' \in S} x_{s'};\ \lambda_{2,s}^a \geq 0 && \forall s \in S^?, \forall a \in \mathcal{A}(s) && \text{(11b)}
\end{aligned}$$

We prove its joint convexity in $\mathbf{x}$ and λ_s^a as follows. Constraint (11a) is generated by a primal-dual transformation, so, according to convex theory, it is convex in the dual variables λ_s^a by construction. Convex theory also guarantees that the affine subtraction of $\mathbf{x}$ from $\lambda_{1,s}^a$ preserves convexity, given $\lambda_{1,s}^a \geq \max x_{s'}, \forall s \in S$ in Constraint (11b), so we conclude that Problem (11) is convex.

For general CMDPs, we will assume:

Assumption 5.1. Given a CMDP $\mathcal{M}_\mathcal{C}$, for all convex uncertainty sets $\mathcal{F}_s^a \in \mathcal{F}$, the dual function $g(\lambda_s^a, \mathbf{x})$ in Problem (8) is jointly convex in both $\lambda_{\mathbf{s}}^{\mathbf{a}}$ and $\mathbf{x}$.

According to Proposition 4.1, Problem (9) can thus be solved in polynomial time. Also, $\mathbf{P}^{max}[\phi_1 \mathcal{U} \phi_2] = \mathbf{x}^*$, so all the satisfaction probabilities can be computed by solving only one convex problem. Finally, we can combine models of uncertainty different from one another within a single CP formulation, since each dual problem is independent from the others according to Assumption 2.1. As an example, if both the interval and ellipsoidal models are used, the overall CP formulation is an SOCP.

Lemma 5.1. *The routine to verify the* Unbounded Until *operator is sound, complete and guaranteed to terminate with algorithmic complexity polynomial in the size $\mathcal{R}$ of $\mathcal{M}_\mathcal{C}$, if $\mathcal{M}_\mathcal{C}$ satisfies Assumption 5.1.*

Proof. The routine solves only one convex program, generated in time polynomial in $\mathcal{R}$ as follows. We formulate Constraints (9b) and (9c) $\forall s \in S^?$ and $a \in \mathcal{A}(s)$, i.e., $O(MQ)$ constraints, where $Q = |S^?| = O(N)$. They are derived from MQ primal-dual transformations as in Equation (8). Each primal problem has N unknowns, $N+1$ constraints to represent the probability simplex and D_s^a constraints to represent the uncertainty set $\mathcal{F}_s^a$. From duality theory, the corresponding dual inner problem has $N + 1 + D_s^a$ unknowns and $2N + 1 + D_s^a$ constraints. Overall, Problem (9) has $O\left((N + 1 + D)MQ\right)$ more unknowns and $O\left((2N + 1 + D)MQ\right)$ more constraints of Problem (6), so its size is polynomial in $\mathcal{R}$. If $\mathcal{M}_\mathcal{C}$ satisfies Assumption 5.1, Problem (9) is convex. Using Proposition 4.1, we conclude that it can be solved in time polynomial in $\mathcal{R}$. Finally, when strong duality holds for the transformation in Equation (8), soundness and completeness of the final solution are preserved because the dual and primal optimal value of each inner problem are equivalent. □

6 Case Studies

We implemented the proposed verification algorithm in Python, and interfaced it with PRISM [5] to extract information about the CMDP model. We used MOSEK [25] to solve the LPs generated for the interval model and implemented customized numerical solvers for the other models of uncertainty. The implemented tool is available at [26]. The algorithm was tested on all the case studies collected in the PRISM benchmark suite [27]. Due to space limits, we report only two of them: the verification of a consensus protocol and of a dynamic configuration protocol for IPv4 addresses. The runtime data were obtained on a 2.4 GHz Intel Xeon with 32GB of RAM.

6.1 Consensus Protocol

Consensus problems arise in many distributed environments, where a group of distributed processes attempt to reach an agreement about a decision to take by accessing some shared entity. A consensus protocol ensures that the processes will eventually terminate and take the same decision, even if they start with initial guesses that might differ from one another.

We analyze the randomized consensus protocol presented in [19, 28]. The protocol guarantees that the processes return a preference value $v \in \{1, 2\}$, with probability parameterized by a process independent value R ($R \geq 2$) and the number of processes P. The processes communicate with one another by accessing a shared counter of value c. The protocol proceeds in rounds. At each round, a process flips a local coin, increments or decrements the shared counter depending on the outcome and then reads its value c. If $c \geq PR$ ($c \leq -PR$), it chooses $v = 1$ ($v = 2$). Note that the larger the value of R, the longer it takes on average for the processes to reach the decision. Nondeterminism is used to model the asynchronous access of the processes to the shared counter, so the overall protocol is modeled as an MDP.

We verify the property **Agreement:** all processes must agree on the same decision, i.e., choose a value $v \in \{1, 2\}$. We compute the minimum probability of **Agreement** and compare it against the theoretical lower bound $(R-1)/2R$ [19]. In PCTL syntax:

$$P_{s_0}^{min}[\psi] := P_{s_0}^{min}(\mathbf{F}(\{finished\} \wedge \{all\_coins\_equal\_1\})) \tag{12}$$

We consider the case where one of the processes is unreliable or adversarial, i.e., it throws a biased coin instead of a fair coin. Specifically, the probability of either outcome lies in the uncertainty interval $[(1-u)p_0, (1+u)p_0]$, where $p_0 = 0.5$ according to the protocol. This setting is relevant to analyze the protocol robustness when a process acts erroneously due to a failure or a security breach. In particular, our approach allows to study attacks that deliberately hide under the noise threshold of the protocol. In such attacks, the compromised node defers agreement by producing outputs whose statistical properties are within the noise tolerance of an uncompromised node, so that it is harder to detect its malicious behavior.

Figure 1 shows the effect of different levels of uncertainty on the computed probabilities for $P = 4$. With no uncertainty ($u = 0$), $P_{s_0}^{min}$ increases as R increases, because a larger R drives the decision regions further apart, making it more difficult for the processes to decide on different values of v. As R goes to infinity, $P_{s_0}^{min}$ approaches the theoretical lower bound $lim_{R\to\infty}(R-1)/2R = 0.5$. However, even with a small uncertainty ($u = 0.01$), $P_{s_0}^{min}$ soon decreases for increasing R. With a large uncertainty ($u = 0.15$), $P_{s_0}^{min}$ quickly goes to 0. A possible explanation is that the faulty process has more opportunities to deter agreement for a high R, since R also determines the expected time to termination. Results thus show that the protocol is vulnerable to uncertainties. This fact may have serious security implication, i.e., a denial-of-service attack could reduce the availability of the distributed service, since a compromised process may substantially alter the expected probability of agreement.

Lastly, we study the scalability of the CP procedure, by evaluating Equation (12) while sweeping R both for $P = 2$ and $P = 4$. We use MOSEK [25] to solve Problem (9) and set the Time Out (TO) to one hour. In Figure 2, we plot the sum $(N + T)$ of the number of states (N) and transitions (T) of the CMDP, which are independent of the uncertainty in the transition probabilities, to represent the model size (top), the sum $(V + C)$ of the number of variables (V) and constraints (C) of the generated LP instances of Problem (9) (center), and the running time t_{CP} (bottom). $V+C$ always scales linearly with $N + T$ (the lines have the same slope), supporting the polynomial complexity result for our algorithm. Instead, t_{CP} scales linearly only for smaller problems ($P = 2$), while it has a higher-order polynomial behavior for larger problems ($P = 4$) (the line is still a straight line but with steeper slope, so it is polynomial on logarithmic axes). This behavior depends on the performance of the chosen numerical solver, and it can improve benefiting of future advancements in the solver implementation. In Table 3, we compare the CP procedure with two tools, PRISM [5] and PARAM [18], in terms of runtime, for varying values of P and R. Although neither tool solves the same problem addressed in this paper, the comparison is useful to assess the practicality of the proposed approach. In particular, PRISM only verifies PCTL properties of MDPs with no uncertainties. PARAM instead derives a symbolic expression of the satisfaction probabilities as a function of the model parameters, to then find the parameter

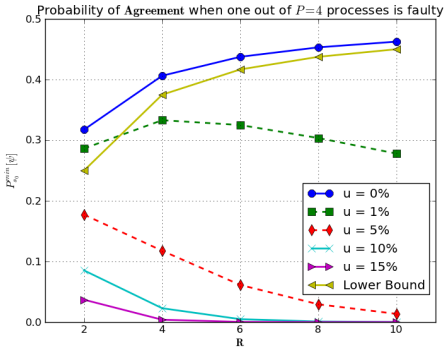

Fig. 1: Value of Eq. 12 in function of R while varying the uncertainty level u.

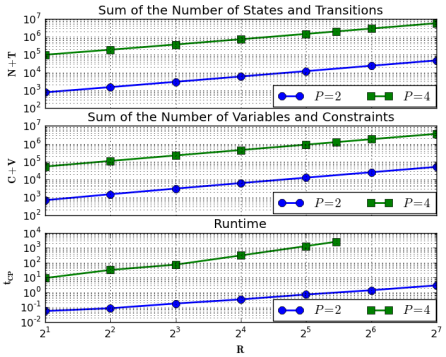

Fig. 2: Scalability of the CP procedure.

values that satisfy the property. Hence, PRISM only considers a special case of the models considered in this paper, while our approach only returns the worst-case scenario computed by PARAM. Results show that the CP procedure runs faster than PRISM for some benchmarks, but it is slower for larger models. This is expected since the scalability of our approach depends mainly on the problem size, while the performance of the iterative engine in PRISM depends on the problem size and on the number of iterations required to achieve convergence, which is dependent on the problem data. Finally, our approach is orders of magnitude faster than PARAM, so it should be preferred to perform worst-case analysis of system performances.

6.2 ZeroConf Dynamic Configuration Protocol for IPv4 Link-Local Addresses

The ZeroConf protocol [29,30] is an Internet Protocol (IP)-based configuration protocol for local (e.g. domestic) networks. In such a local context, each device should configure its own unique IP address when it gets connected to the network, with no user intervention. The protocol thus offers a distributed "plug-and-play" solution in which address configuration is managed by individual devices when they are connected to the network. The network is composed of DV_{tot} devices. After being connected, a new device chooses randomly an IP address from a pool of $IP_A = 65024$ available ones, as specified by the standard. The address is non-utilized with probability $p_0 = 1 - DV_{tot}/IP_A$. It then sends messages to the other devices in the network, asking whether the chosen IP

Table 3: Runtime Comparison

Tool	$P=2, R=2$ $N+T=764$	$R=7$ $2,604$	$R=128$ $47,132$	$P=4, R=2$ $97,888$	$R=32$ $1,262,688$	$R=44$ $1,979,488$	$P=6, R=4$ $14,211,904$
CP	$0.02s$	$0.1s$	$2.1s$	$8.3s$	$1,341s$	$2,689$	TO
PRISM	$0.01s$	$0.09s$	$196s$	$1s$	$2,047s$	TO	$1860s$
PARAM	$22.8s$	$657s$	TO	TO	TO	TO	TO

address is already in use. If no reply is received, the device starts using the IP address, otherwise the process is repeated.

The protocol is both probabilistic and timed: probability is used in the randomized selection of an IP address and to model the eventuality of message loss; timing defines intervals that elapse between message retransmissions. In [30], the protocol has been modeled as an MDP using the digital clock semantic of time. In this semantic, time is discretized in a finite set of epochs which are mapped to a finite number of states in an MDP, indexed by the epoch variable t_e. To enhance the user experience and, in battery-powered devices, to save energy, it is important to guarantee that a newly-connected device manages to select a unique IP address within a given deadline *dl*. For numerical reasons, we study the maximum probability of *not* being able to select a valid address within *dl*. In PCTL syntax:

$$P_{s_0}^{max}\left[\psi\right] := P_{s_0}^{max}\left(\neg\{unique\_address\}\,\mathcal{U}\,\{t_e > dl\}\right) \tag{13}$$

We analyzed how network performances vary when there is uncertainty in estimating: 1) the probability of selecting an IP address, and; 2) the probability of message loss during transmission. The former may be biased in a faulty or malicious device. The latter is estimated from empirical data, so it is approximated. Further, the IMDP semantic of IDTMCs (Section 1), which allows a nature to select a different transition distribution at each execution step, properly models the time-varying characteristics of the transmission channel.

In Figure 3, we added uncertainty only to the probability of message loss using the likelihood model, which is suitable for empirically-estimated probabilities. Using classical results from statistics [11], we computed the value of parameter β from Set (2) corresponding to several confidence levels C_L in the measurements. In particular, $0 \leq C_L \leq 1$ and $C_L = 1 - cdf_{\chi_d^2}\left(2 * (\beta_{max} - \beta)\right)$, where $cdf_{\chi_d^2}$ is the cumulative density function of the Chi-squared distribution with d degrees of freedom ($d = 2$ here because there are two possible outcomes, message lost or received). Results show that the value of $P_{s_0}^{max}$ increases by up to $\sim 10\times$ for decreasing C_L, while classical model-checking would only report the value for $C_L = 1$, which roughly over-estimates network performance. The plot can be used by a designer to choose *dl* to make the protocol robust to varying channel conditions, or by a field engineer to assess when the collected measurements are enough to estimate network performances.

In Figure 4, we compose different models of uncertainty, i.e., we also add uncertainty in the probability of selecting the new IP address using the interval model. This probability thus lies in the interval $[(1-u)p_0, (1+u)p_0]$. We, arbitrarily, fixed $dl = 25$ and swept DV_{tot} in the range $[10 - 100]$, which covers most domestic applications, to study how network congestion affects the value of Equation 13. We studied four scenarios: the *ideal* scenario, returned by classical model-checking techniques; the *confident, normal, conservative* scenarios, where we added increasing uncertainty to model different knowledge levels of the network behavior, a situation that often arises during the different design phases, from conception to deployment. Results show that $P_{s_0}^{max}\left[\psi\right]$ gets up to $\sim\ 15\times$ higher than the ideal scenario, an information that designers can use to determine the most sensitive parameters of the system and to assess the impact of their modeling assumptions on the estimation of network performances.

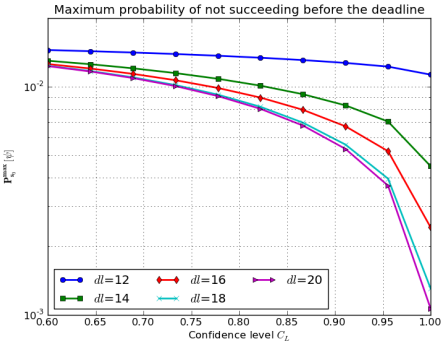

Fig. 3: Value of Equation 13 (top) and verification runtime (bottom).

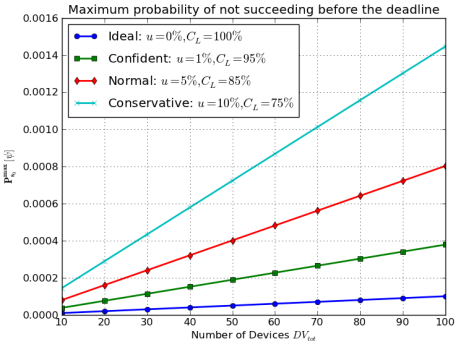

Fig. 4: Value of Eq. 13 for increasing number of devices in the network.

7 Conclusions and Future Work

We addressed the problem of verifying PCTL properties of Convex-MDPs (CMDPs), i.e., MDPs whose transition probabilities lie within convex uncertainty sets. Using results on strong duality for convex programs, we proved that model checking is decidable in PTIME for the fragment of PCTL without the *Bounded Until* operator. For the entire PCTL syntax, the algorithmic complexity is pseudo-polynomial in the size of the property. Verification results on two case studies show that uncertainty can greatly alter the computed probabilities, thus revealing the importance of the proposed analysis.

As future work, we aim to relax the *rectangular uncertainty* assumption, to obtain a less conservative analysis. Also, we plan to verify a complex physical system, e.g. an airplane power system, in which modeling uncertainties are present both in the underlying physical process and in the failure probabilities of its components.

8 Acknowledgments

The authors thank John B. Finn for the contribution in the first stages of the project and the reviewers for their helpful comments. The research was partially funded by DARPA Award Number HR0011-12-2-0016 and by STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

References

1. C. Courcoubetis and M. Yannakakis, "The Complexity of Probabilistic Verification," *Journal of ACM*, vol. 42(4), pp. 857–907, 1995.
2. A. Bianco and L. De Alfaro, "Model Checking of Probabilistic and Nondeterministic Systems," in *Proc. of FSTTCS*, ser. LNCS, vol. 1026, 1995, pp. 499–513.
3. M. Kwiatkowska, "Quantitative Verification: Models, Techniques and Tools," in *Proc. of SIGSOFT*, 2007, pp. 449–458.

4. H. Hansson and B. Jonsson, "A Logic for Reasoning About Time and Reliability," *Formal Aspects of Computing*, vol. 6(5), pp. 512–535, 1994.
5. M. Kwiatkowska *et al.*, "PRISM 4.0: Verification of Probabilistic Real-Time Systems," *Proc. of CAV*, pp. 585–591, 2011.
6. I. Kozine and L. Utkin, "Interval-Valued Finite Markov Chains," *Reliable Computing*, vol. 8(2), pp. 97–113, 2002.
7. K. Sen *et al.*, "Model-Checking Markov Chains in the Presence of Uncertainties," *Proc. of TACAS*, vol. 3920, pp. 394–410, 2006.
8. K. Chatterjee, K. Sen, and T. Henzinger, "Model-Checking ω-regular Properties of Interval Markov Chains," in *Proc. of FOSSACS*, 2008, pp. 302–317.
9. A. Ben-Tal and A. Nemirovski, "Robust Solutions of Uncertain Linear Programs," *Oper. Res. Lett.*, vol. 25(1), pp. 1–13, 1999.
10. A. Andreychenko *et al.*, "Parameter Identification for Markov Models of Biochemical Reactions," in *Proc. of CAV*, 2011, pp. 83–98.
11. A. Nilim and L. El Ghaoui, "Robust Control of Markov Decision Processes with Uncertain Transition Matrices," *Journal of Operations Research*, pp. 780–798, 2005.
12. A. Puggelli *et al.*, "Polynomial-Time Verification of PCTL Properties of MDPs with Convex Uncertainties," UC-Berkeley, Tech. Rep. UCB/EECS-2013-24, Apr 2013. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-24.html
13. M. Y. Vardi, "Automatic Verification of Probabilistic Concurrent Finite State Programs," in *Proc. of SFCS*, 1985, pp. 327–338.
14. E. Lehmann and G. Casella, *Theory of Point Estimation*. Springer-Verlag, New York, 1998.
15. M. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.
16. V. Forejt *et al.*, "Automated Verification Techniques for Probabilistic Systems," *SFM*, vol. 6659, pp. 53–113, 2011.
17. R. Barbuti *et al.*, "Probabilistic Model Checking of Biological Systems with Uncertain Kinetic Rates," in *Reachability Problems*. Springer, 2009, vol. 5797, pp. 64–78.
18. E. M. Hahn *et al.*, "Synthesis for PCTL in Parametric Markov Decision Processes," 2011.
19. M. Kwiatkowska *et al.*, "Automated Verification of a Randomized Distributed Consensus Protocol Using Cadence SMV and PRISM," in *Proc. of CAV*, 2001, pp. 194–206.
20. E. Wolff *et al.*, "Robust Control of Uncertain Markov Decision Processes with Temporal Logic Specifications," *CDC*, 2012.
21. A. D'Innocenzo *et al.*, "Robust PCTL Model Checking," in *Proc. of HSCC*, 2012, pp. 275–286.
22. E. Clarke and A. Emerson, "Design and Synthesis of Synchronization Skeletons Using Branching Time Temporal Logic," *Proc. of WLP*, vol. 131, 1981.
23. Y. Nesterov and A. Nemirovski, *Interior-Point Polynomial Algorithms in Convex Programming*, ser. Studies in Applied and Numerical Mathematics, 1994.
24. S. Boyd and L. Vandenberghe, "Convex Optimization," *Cambridge University Press*, 2004.
25. "MOSEK," http://www.mosek.com.
26. Online: http://www.eecs.berkeley.edu/~puggelli/.
27. Online: http://www.prismmodelchecker.org/benchmarks/.
28. J. Aspnes and M. Herlihy, "Fast Randomized Consensus Using Shared Memory," *Journal of Algorithms*, vol. 11(3), pp. 441–461, 1990.
29. S. Cheshire, B. Adoba, and E. Gutterman, "Dynamic configuration of IPv4 link local addresses," available from http://www.ietf.org/rfc/rfc3927.txt.
30. M. Kwiatkowska *et al.*, "Performance Analysis of Probabilistic Timed Automata Using Digital Clocks," *Formal Methods in System Design*, vol. 29, pp. 33–78, 2006.