Model-Checking Markov Chains in the presence of Uncertainties

Koushik Sen, Mahesh Viswanathan, Gul Agha
Department of Computer Science,
University of Illinois at Urbana-Champaign.
{ksen,vmahesh,agha}@uiuc.edu

Abstract. We investigate the problem of model checking Interval-valued Discrete-time Markov Chains (IDTMC). IDTMCs are discrete-time finite Markov Chains for which the exact transition probabilities are not known. Instead in IDTMCs, each transition is associated with an interval in which the actual transition probability must lie. We consider two semantic interpretations for the uncertainty in the transition probabilities of an IDTMC. In the first interpretation, we think of an IDTMC as representing a (possibly uncountable) family of (classical) discrete-time Markov Chains, where each member of the family is a Markov Chain whose transition probabilities lie within the interval range given in the IDTMC. This semantic interpretation we call Uncertain Markov Chains (UMC). In the second semantics for an IDTMC, which we call Interval Markov Decision Process (IMDP), we view the uncertainty as being resolved through non-determinism. In other words, each time a state is visited, we adversarially pick a transition distribution that respects the interval constraints, and take a probabilistic step according to the chosen distribution. We show that the PCTL model checking problem for both Uncertain Markov Chain semantics and Interval Markov Decision Process semantics is decidable in PSPACE. We also prove lower bounds for these model checking problems.

1 Introduction

Discrete time stochastic models such as *Discrete Time Markov Chains* (DTMCs) have been used to analyze the correctness, reliability, and performance of systems [8, 11, 21, 15]. In a DTMC, the system is assumed to have finitely many states, and the system's future behavior is completely determined by its current state. From each state of the system, the probability of transitioning to any other given state at the next step is fixed and is given by the transition probability matrix of the DTMC.

The assumption that the system makes transitions according to a fixed distribution at each step and that this distribution is precisely known when modeling, is a strong assumption that may often not hold in practice [14, 17, 28, 16]. If the system being modeled is an open system, i.e., interacts with an environment, then uncertainty in the transitions may arise due to imperfect information about the environment. For example, consider a system that interacts with an imperfect communication medium that may lose messages. The probability of message loss may either depend on choice of the communication medium or on a complicated, time-varying dependence on events that are not precisely understood at the time of modeling the system. Another source of imprecision is that the transition probabilities in the system model are often estimated through statistical experiments, which only provide bounds on the transition probabilities.

In order to faithfully capture these system uncertainties in stochastic models, the model of *Interval-valued Discrete-time Markov Chains (IDTMC)* has been introduced [14, 16]. These are DTMC models where the exact probability of taking a state transition is not known, and instead the transition probability is assumed to lie within a range associated with the transition. Two semantic interpretations have been suggested for such models. *Uncertain Markov Chains* (UMC) [14] is an interpretation of an IDTMC as a family of (possibly uncountably many) DTMCs, where each member of the family is a DTMC whose transition probabilities lie within the interval range given in the IDTMC. In the second interpretation, called *Interval Markov Decision Process* (IMDP), we view the uncertainty as being resolved through non-determinism. In other words, each time a state is visited, we adversarially pick a transition distribution that respects the interval constraints, and take a probabilistic step according to the chosen distribution. Thus, IMDPs allow the possibility of modeling a non-deterministic choice made from a set of (possibly) uncountably many choices. An IMDP can be seen as a generalization of Markov Decision Processes (MDPs) [19, 3, 24].

We investigate the problem of model checking PCTL specifications for IDTMC. The two semantic interpretations of IDTMCs yield very different model checking results (whenever the property has at least two probabilistic operators, not necessarily nested; see example in Figure 1) and require different algorithmic techniques. For the case of UMCs, we show that PCTL model checking problem can be reduced to finding feasible solutions to inequality constraints, much like in the case of DTMC and MDP [8, 4, 3, 21, 7]. However, there is one important difference. The constraints to be solved in the case of UMCs are polynomial and not just linear (as for DTMCs and MDPs). Since the *existential theory of reals* is decidable in PSPACE [20, 6], the feasibility of the polynomial constraints arising in model checking, can be determined by making a "query" to the existential theory of reals. Thus, the PCTL model checking problem for UMCs is in PSPACE. In practice, however, this algorithm may not be the most efficient. The constraints we obtain during model checking all take a special form: the polynomials are *bilinear* [1]. Therefore, it might be more efficient to instead use algorithms for solving *bilinear matrix inequalities* (BMIs) [10, 9] or tools developed for this purpose [18]. Checking feasibility of BMIs is known to be NP-hard [26], but the exact complexity, which is lower than PSPACE, is unknown. On the other hand, in the case of IMDPs, we show that the model checking problem can be reduced to model checking an MDP of exponential size. We then use known results for MDPs to show that IMDPs can be model checked in PSPACE. We also present an iterative model checking algorithm for IMDPs which may prove to be more efficient in practice.

In addition to demonstrating the decidability of the model checking problem, we also prove lower bounds on the complexity of the model checking problem. We show that the model checking problem for UMCs is NP-hard and co-NP-hard; thus, for UMCs the problem is unlikely to be in P. A straightforward corollary of our results is that solving BMIs is also co-NP-hard. For IMDPs, we can only show P-hardness; in fact, even this is a consequence of the P-hardness of (classical) DTMC model checking.

[1] The highest power of any variable in the polynomial is 1, and any term is the product of at most two variables.

The rest of the paper is organized as follows. We briefly discuss related work next. In Section 2 we formally define IDTMC and give its semantics as UMC and IMDP. PCTL and the model checking problem is introduced in Section 3. We then revisit the model checking algorithm for DTMC (Section 4) and present a modified version of the classical algorithm. The ideas in the section play a key role in our UMC model checking algorithm. Section 5 (UMC) and Section 6 (IMDP) contain our main results about the model checking problem, providing both upper and lower bounds. Finally we present our conclusions in Section 7. Motivating examples of UMCs and IMDPs and observations about BMI optimization problems are deferred to Appendix.

Related Work. The model of IDTMCs has been introduced independently by Jonsson and Larsen [14] and Kozine and Utkin [16] under the names *interval specification systems* and *interval-valued finite Markov chains*, respectively. However, they consider different semantic interpretations. Jonsson and Larsen consider the UMC interpretation and study bisimulation and simulation preorders for such an interpretation. Kozine and Utkin, on the other hand, take the IMDP interpretation and present algorithms to compute the probability distribution on the states after t steps. Neither of these papers investigate the PCTL model checking problem which is the focus of this paper. We introduce new names to emphasize the subtle semantic difference in the two interpretations. A more general model called *generalized Markov processes* for describing infinite families of Markov Chains was introduced in [1]. In that paper, they showed that model checking such models with respect to PCTL* (a more general logic than PCTL) is decidable and has elementary complexity. PCTL model checking for classical DTMC and MDP models has been considered in [8, 4, 3, 21, 7].

2 Formal Models

Definition 1. A discrete-time Markov chain *(DTMC) is a 4-tuple* $\mathcal{M} = (S, s_I, \mathbf{P}, L)$, *where*

1. S *is a finite set of* states,
2. $s_I \in S$ *is the* initial *state,*
3. $\mathbf{P}\colon S \times S \to [0, 1]$ *is a* transition probability matrix, *such that* $\sum_{s' \in S} \mathbf{P}(s, s') = 1$, *and*
4. $L\colon S \to 2^{\mathrm{AP}}$ *is a* labeling *function that maps states to sets of atomic propositions from a set* AP.

A non-empty sequence $\pi = s_0 s_1 s_2 \cdots$ is called a *path* of $\mathcal{M}$, if each $s_i \in S$ and $\mathbf{P}(s_i, s_{i+1}) > 0$ for all $i \geq 0$. We denote the i^{th} state in a path π by $\pi[i] = s_i$. We let *Path*(s) be the set of paths starting at state s. A probability measure on paths is induced by the matrix $\mathbf{P}$ as follows.

Let $s_0, s_1, \ldots, s_k \in S$ with $\mathbf{P}(s_i, s_{i+1}) > 0$ for all $0 \leq i < k$. Then $C(s_0 s_1 \ldots s_k)$ denotes a *cylinder set* consisting of all paths $\pi \in$ *Path*(s_0) such that $\pi[i] = s_i$ (for $0 \leq i \leq k$). Let $\mathcal{B}$ be the smallest σ-algebra on *Path*(s_0) which contains all the cylinders $C(s_0 s_1 \ldots s_k)$. The measure μ on cylinder sets can be defined as follows

$$\mu(C(s_0 s_1 \ldots s_k)) = \begin{cases} 1 & \text{if } k = 0 \\ \mathbf{P}(s_0, s_1) \cdots \mathbf{P}(s_{k-1}, s_k) & \text{otherwise} \end{cases}$$

The *probability measure* on $\mathcal{B}$ is then defined as the unique measure that agrees with μ (as defined above) on the cylinder sets.

Definition 2. An Interval-valued Discrete-time Markov chain *(IDTMC) is a 5-tuple* $\mathcal{I} = (S, s_I, \check{\mathbf{P}}, \hat{\mathbf{P}}, L)$*, where*

1. S *is a finite set of* states*,*
2. $s_I \in S$ *is the* initial *state,*
3. $\check{\mathbf{P}} \colon S \times S \to [0,1]$ *is a* transition probability matrix*, where each* $\check{\mathbf{P}}(s,s')$ *gives the* lower bound *of the transition probability from the state* s *to the state* s'*,*
4. $\hat{\mathbf{P}} \colon S \times S \to [0,1]$ *is a* transition probability matrix*, where each* $\hat{\mathbf{P}}(s,s')$ *gives the* upper bound *of the transition probability from the state* s *to the state* s'*,*
5. $L \colon S \to 2^{\mathrm{AP}}$ *is a* labeling *function that maps states to sets of atomic propositions from a set* AP.

We consider two semantics interpretations of an IDTMC model, namely Uncertain Markov Chains (UMC) and Interval Markov Decision Processes (IMDP).

Uncertain Markov Chains An IDTMC $\mathcal{I}$ may represent an infinite set of DTMCs, denoted by $[\mathcal{I}]$, where for each DTMC $(S, s_I, \mathbf{P}, L) \in [\mathcal{I}]$ the following is true,

– $\check{\mathbf{P}}(s,s') \leq \mathbf{P}(s,s') \leq \hat{\mathbf{P}}(s,s')$ for all pairs of states s and s' in S

In the Uncertain Markov Chains semantics, or simply, in the UMCs, we assume that the external environment non-deterministically picks a DTMC from the set $[\mathcal{I}]$ at the beginning and then all the transitions take place according to the chosen DTMC. Note that in this semantics, the external environment makes only one non-deterministic choice. Henceforth, we will use the term UMC to denote an IDTMC interpreted according to the Uncertain Markov Chains semantics.

Interval Markov Decision Processes In the Interval Markov Decision Processes semantics, or simply, in the IMDPs, we assume that before every transition the external environment non-deterministically picks a DTMC from the set $[\mathcal{I}]$ and then takes a one-step transition according to the probability distribution of the chosen DTMC. Note that in this semantics, the external environment makes a non-deterministic choice before every transition. Henceforth, we will use the term IMDP to denote an IDTMC interpreted according to the Interval Markov Decision Processes semantics. We now formally define this semantics.

Let *Steps*(s) be the set of probability density functions over S defined as follows:

$$\textit{Steps}(s) = \{\mu \colon S \to \mathbb{R}^{\geq 0} \mid \sum_{s' \in S} \mu(s') = 1 \text{ and } \check{\mathbf{P}}(s,s') \leq \mu(s') \leq \hat{\mathbf{P}}(s,s') \text{ for all } s' \in S\}$$

In an IMDP, at every state $s \in S$, a probability density function μ is chosen non-deterministically from the set *Steps*(s). A successor state s' is then chosen according to the probability distribution μ over S.

A *path* π in an IMDP $\mathcal{I} = (S, s_I, \check{\mathbf{P}}, \hat{\mathbf{P}}, L)$ is a non-empty sequence of the form $s_0 \xrightarrow{\mu_1} s_1 \xrightarrow{\mu_2} \ldots$, where $s_i \in S$, $\mu_{i+1} \in$ *Steps*(s_i), and $\mu_{i+1}(s_{i+1}) > 0$ for all $i \geq 0$. A path can be either finite or infinite. We use π_{fin} to denote a finite path. Let *last*(π_{fin}) be the last state in the finite path π_{fin}. As in DTMC, we denote the i^{th} state in a path π by $\pi[i] = s_i$. We let *Path*(s) and *Path*$_{\mathrm{fin}}(s)$ be the set of all infinite and finite paths, respectively, starting at state s. To associate a probability measure with the paths, we resolve the non-deterministic choices by an *adversary*, which is defined as follows:

Definition 3. *An* adversary A *of an IMDP* $\mathcal{I}$ *is a function mapping every finite path* π_{fin} *of* $\mathcal{I}$ *onto an element of the set Steps*$(last(\pi_{\mathrm{fin}}))$. *Let* $\mathcal{A}_{\mathcal{I}}$ *denote the set of all possible adversaries of the IMDP* $\mathcal{I}$. *Let Path*$^A(s)$ *denote the subset of Path*(s) *which corresponds to* A.

The behavior of an IMDP $\mathcal{I} = (S, s_I, \check{\mathbf{P}}, \hat{\mathbf{P}}, L)$ under a given adversary A is purely deterministic. The behavior of a IMDP $\mathcal{I}$ from a state s can be described by an infinite-state DTMC $\mathcal{M}^A = (S^A, s_I^A, \mathbf{P}^A, L^A)$ where

- $S^A =$ *Path*$_{\mathrm{fin}}(s)$,
- $s_I^A = s$, and
- $\mathbf{P}^A(\pi_{\mathrm{fin}}, \pi'_{\mathrm{fin}}) = \begin{cases} A(\pi_{\mathrm{fin}})(s') & \text{if } \pi'_{\mathrm{fin}} \text{ is of the form } \pi_{\mathrm{fin}} \xrightarrow{A(\pi_{\mathrm{fin}})} s' \\ 0 & \text{otherwise} \end{cases}$

There is a one-to-one correspondence between the paths of $\mathcal{M}^A$ and *Path*$^A(s)$ of $\mathcal{I}$. Therefore, we can define a probability measure $Prob_s^A$ over the set of paths *Path*$^A(s)$ using the probability measure of the DTMC $\mathcal{M}^A$.

3 Probabilistic Computation Tree Logic (PCTL)

In this paper we consider a sub-logic of PCTL that excludes the steady-state probabilistic operators. The formal syntax and semantics of this logic is as follows.

PCTL Syntax

$$\begin{aligned} \phi &::= \mathit{true} \mid a \mid \neg\phi \mid \phi \wedge \phi \mid \mathcal{P}_{\bowtie p}(\psi) \\ \psi &::= \phi\, \mathcal{U}\, \phi \mid \mathbf{X}\phi \end{aligned}$$

where $a \in \mathrm{AP}$ is an atomic propositions, $\bowtie \in \{<, \leq, >, \geq\}$, $p \in [0,1]$, and $k \in \mathbb{N}$. Here ϕ represents a *state* formula and ψ represents a *path* formula.

PCTL Semantics for DTMC

The notion that a state s (or a path π) *satisfies* a formula ϕ in a DTMC $\mathcal{M}$ is denoted by $s \models_{\mathcal{M}} \phi$ (or $\pi \models_{\mathcal{M}} \phi$), and is defined inductively as follows:

$$\begin{array}{ll} s \models_{\mathcal{M}} \mathit{true} & \\ s \models_{\mathcal{M}} a & \text{iff } a \in L(s) \\ s \models_{\mathcal{M}} \neg\phi & \text{iff } s \not\models_{\mathcal{M}} \phi \\ s \models_{\mathcal{M}} \phi_1 \wedge \phi_2 & \text{iff } s \models_{\mathcal{M}} \phi_1 \text{ and } s \models_{\mathcal{M}} \phi_2 \\ s \models_{\mathcal{M}} \mathcal{P}_{\bowtie p}(\psi) & \text{iff } Prob\{\pi \in \mathit{Path}(s) \mid \pi \models_{\mathcal{M}} \psi\} \bowtie p \\ \pi \models_{\mathcal{M}} \mathbf{X}\phi & \text{iff } \pi[1] \models_{\mathcal{M}} \phi \\ \pi \models_{\mathcal{M}} \phi_1\, \mathcal{U}\, \phi_2 & \text{iff } \exists i \geq 0\ (\pi[i] \models_{\mathcal{M}} \phi_2 \text{ and } \forall j < i.\ \pi[j] \models_{\mathcal{M}} \phi_1) \end{array}$$

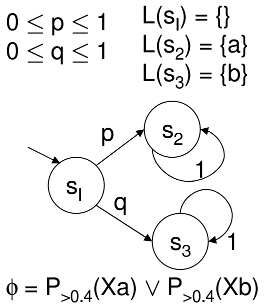

Fig. 1. Example IDTMC and PCTL formula ϕ. The UMC interpretation of the IDTMC satisfies ϕ, whereas the IMDP interpretation of the IDTMC violates ϕ

$s \models true$	
$s \models a$	iff $a \in \mathrm{AP}(s)$
$s \models \neg\phi$	iff $s \not\models \phi$
$s \models \phi_1 \wedge \phi_2$	iff $s \models \phi_1$ and $s \models \phi_2$
$s \models \mathcal{P}_{\bowtie p}(\psi)$	iff $Prob_s^A(\{\pi \in Path^A(s) \mid \pi \models \psi\}) \bowtie p$ for all $A \in \mathcal{A}$
$\pi \models \mathbf{X}\phi$	iff $\pi[1] \models \phi$
$\pi \models \phi_1 \,\mathcal{U}\, \phi_2$	iff $\exists i \geq 0\ (\pi[i] \models \phi_2$ and $\forall j < i.\ \pi[j] \models \phi_1)$

Fig. 2. PCTL semantics for IMDP

It can shown that for any path formula ψ and any state s, the set $\{\pi \in Path(s) \mid \pi \models_{\mathcal{M}} \psi\}$ is measurable [27]. A formula $\mathcal{P}_{\bowtie p}(\psi)$ is satisfied by a state s if $Prob[$path starting at s satisfies $\psi] \bowtie p$. The path formula $\mathbf{X}\phi$ holds over a path if ϕ holds at the second state on the path. The formula $\phi_1 \,\mathcal{U}\, \phi_2$ is true over a path π if ϕ_2 holds in some state along π, and ϕ holds along all prior states along π.

Given a DTMC $\mathcal{M}$ and a PCTL state formula ϕ, $\mathcal{M} \models \phi$ iff $s_I \models_{\mathcal{M}} \phi$.

PCTL Semantics for UMC

Given a UMC $\mathcal{I}$ and a PCTL state formula ϕ, we say $\mathcal{I} \models \phi$ iff, for all $\mathcal{M} \in [\mathcal{I}]$, $\mathcal{M} \models \phi$. Note that $\mathcal{I} \not\models \phi$ does not imply that $\mathcal{I} \models \neg\phi$. This because if $\mathcal{I} \not\models \phi$, there may exist $\mathcal{M}, \mathcal{M}' \in [\mathcal{I}]$ such that $\mathcal{M} \models \phi$ and $\mathcal{M}' \models \neg\phi$.

PCTL Semantics for IMDP

The interpretation of a state formula and a path formula of PCTL for IMDPs is same as for DTMCs except for the state formulas of the form $\mathcal{P}_{\bowtie p}(\psi)$.

The notion that a state s (or a path π) *satisfies* a formula ϕ in a IMDP $\mathcal{I}$ is denoted by $s \models \phi$ (or $\pi \models \phi$), and is defined inductively in Figure 2.

The model checking of IDTMC with respect to the two semantics can give different results. For example, consider the IDTMC in Figure 1 and the PCTL formula ϕ. The UMC semantics of this IDTMC satisfies ϕ, while the IMDP semantics violates ϕ.

4 Revisiting DTMC Model-Checking

In this section we outline the basic model checking algorithm for (classical) DTMCs. The algorithm that we outline here for DTMCs is not the most efficient (like the one presented in [8]); however the main ideas presented here will form the crux of our model checking algorithm for UMCs.

The algorithm for model checking DTMCs will reduce the problem to checking the feasibility of simultaneously satisfying a finite set of polynomial inequalities. This feasibility test can be done by checking if a first-order formula with existential quantifiers about the real numbers is true. More precisely, we need to check if a formula of the form $\exists x_1, \ldots, x_n P(x_1, \ldots, x_n)$ is valid over the reals, where P is a boolean function of atomic predicates of the form $f_i(x_1, \ldots, x_n) \bowtie 0$, where f_i is a multivariate polynomial and $\bowtie \in \{=, \neq, \leq, \geq, <, >\}$. It is well-known that this problem can be decided in PSPACE [20, 6] [2].

The model checking algorithm for DTMC takes a DTMC $\mathcal{M} = (S, s_I, \mathbf{P}, L)$ and a PCTL formula ϕ as input. The output is the set $\mathrm{Sat}(\phi) = \{s \in S \mid s \models_{\mathcal{M}} \phi\}$, i.e., the set of all states of the model that satisfy ϕ. We say $\mathcal{M} \models \phi$ iff $s_I \in \mathrm{Sat}(\phi)$.

The algorithm works by recursively computing the set $\mathrm{Sat}(\phi')$ for each sub-formula ϕ' of ϕ as follows.

$$\begin{array}{rlrl} \mathrm{Sat}(\mathit{true}) &= S & \mathrm{Sat}(a) &= \{s \mid a \in L(S)\} \\ \mathrm{Sat}(\neg\phi) &= S \setminus \mathrm{Sat}(\phi) & \mathrm{Sat}(\phi_1 \wedge \phi_2) &= \mathrm{Sat}(\phi_1) \cap \mathrm{Sat}(\phi_2) \\ \mathrm{Sat}(\mathcal{P}_{\bowtie p}(\psi)) &= \{s \in S \mid p_s(\psi) \bowtie p\} & & \end{array}$$

where $p_s(\psi) = \mathit{Prob}\{\pi \in \mathit{Path}(s) \mid \pi \models_{\mathcal{M}} \psi\}$. The computation of the set $\mathrm{Sat}(\mathcal{P}_{\bowtie p}(\psi))$ requires the computation of $p_s(\psi)$ at every state $s \in S$.

If $\psi = \mathbf{X}\phi$, then $p_s(\psi) = \sum_{s' \in \mathrm{Sat}(\phi)} \mathbf{P}(s, s')$.

To compute $p_s(\phi_1 \,\mathcal{U}\, \phi_2)$, we first split the set of states S into three disjoint subsets, S^{no}, S^{yes}, and $S^{?}$ where $S^{\mathrm{no}} = \mathrm{Sat}(\neg\phi_1 \wedge \neg\phi_2)$, $S^{\mathrm{yes}} = \mathrm{Sat}(\phi_2)$, and $S^{?} = S \setminus (S^{\mathrm{no}} \cup S^{\mathrm{yes}})$. Moreover, let $S^{?\mathrm{no}}$ be the set $\{s \mid p_s(\phi_1 \,\mathcal{U}\, \phi_2) = 0\} \setminus S^{\mathrm{no}}$ and $S^{>0}$ be the set $\{s \mid p_s(\phi_1 \,\mathcal{U}\, \phi_2) > 0\}$. Note that $S = S^{>0} \cup S^{?\mathrm{no}} \cup S^{\mathrm{no}}$. By [8], $\{x_s = p_s(\phi_1 \,\mathcal{U}\, \phi_2) \mid s \in S\}$ is a solution of the following linear equation system.

$$x_s = \begin{cases} 0 & \text{if } s \in S^{\mathrm{no}} \\ 1 & \text{if } s \in S^{\mathrm{yes}} \\ \sum_{s' \in S} \mathbf{P}(s, s') x_{s'} & \text{if } s \in S^{?} \end{cases} \tag{1}$$

Note that the equation system (1) can have infinite number of solutions. For example, consider the formula $\mathit{true}\ \mathcal{U}\ a$, where a is an atomic proposition and the DTMC $\mathcal{M} = (\{s\}, s, \mathbf{P}, L)$, where $\mathbf{P}(s, s) = 1$ and $L(s) = \emptyset$. Note that $s \in S^{?\mathrm{no}}$. The linear equation system (1) that is instantiated for computing $p_s(\mathit{true}\,\mathcal{U}\, a)$ for $\mathcal{M}$ is $x_s = x_s$. The system has infinite number of solutions.

We can ensure that $\{x_s = p_s(\phi_1 \,\mathcal{U}\, \phi_2) \mid s \in S\}$ is a unique solution of a system of equations as follows. Fix a γ such that $0 < \gamma < 1$. Consider the following linear equation system.

$$x'_s = \begin{cases} 0 & \text{if } s \in S^{\mathrm{no}} \\ 1 & \text{if } s \in S^{\mathrm{yes}} \\ \sum_{s' \in S} \gamma \mathbf{P}(s, s') x'_{s'} & \text{if } s \in S^{?} \end{cases} \tag{2}$$

Lemma 1. *$x'_s > 0$ iff $s \in S^{>0}$.*

Proof. (Case $\Leftarrow$) If $s \in S^{>0}$, then $p_s(\phi_1 \,\mathcal{U}\, \phi_2) > 0$. This implies that there is a finite path from s to a $s' \in S^{\mathrm{yes}}$ such that all the states on the path are in $S^{>0}$ and all the edges in the path have non-zero probability. Let $l(s)$ be the length of the shortest such

[2] If one takes the computational model to be Turing machines, then this result holds when the coefficients of the polynomials are rationals. One the other hand, if one considers a model of computation that is appropriate for real number computation, like the one proposed by Blum, Shub, and Smale [5], then the algorithm can handle even real coefficients.

path from s. For a state s, if $l(s) = 0$, then $s \in S^{\text{yes}}$ and hence $x'_s = 1$. Let us assume that for any s, such that $l(s) = i$, $x'_s > 0$. Consider a s, such that $l(s) = i + 1$. Then there exists an s' such that $l(s') = i$ and $\mathbf{P}(s, s') > 0$. Therefore, the equation $x'_s = \sum_{s' \in S} \gamma \mathbf{P}(s, s') x'_{s'}$ if $s \in S^?$ in (2) implies that $x'_s \geq \gamma \mathbf{P}(s, s') x'_{s'} > 0$. This is because $x'_{s'} > 0$ by the induction hypothesis. This proves that $s \in S^{>0}$ implies $x'_s > 0$.

(Case ⇒) We prove this by contradiction. Let us assume that there is a s such that $x'_s > 0$ and $s \notin S^{>0}$. Let X be the set $\{x'_{s'} \mid s' \notin S^{>0} \text{ and } x'_{s'} > 0\}$. Let s be such that x'_s =max(X). If $x'_s = 1$, then s must be in S^{yes} by the system of linear equations in (2), which is a contradiction. If $1 > x'_s > 0$, then consider the equation $x'_s = \sum_{s' \in S} \gamma \mathbf{P}(s, s') x'_{s'}$ in (2). Let s' be such that $x'_{s'}$ =max$\{x'_{s''} \mid s'' \in S \text{ and } \mathbf{P}(s, s'') > 0\}$. Then $x'_s \leq \gamma x'_{s'} \sum_{s'' \in S} \mathbf{P}(s, s'') \leq \gamma x'_{s'} < x'_{s'}$. Because $s \notin S^{>0}$ and $\mathbf{P}(s, s') > 0$, s' must not be in $S^{>0}$. Since $x'_{s'} > x'_s$ and $s' \notin S^{>0}$, s' is in X. Therefore, x'_s is not max(X), which is a contradiction. □.

Lemma 2. *The system of linear equations in* (2) *has a unique solution.*

Proof. The proof is by contradiction. Let $\{x'_s = \hat{x}'_s \mid s \in S\}$ and $\{x'_s = \bar{x}'_s \mid s \in S\}$ be two different solutions of the system of linear equations in (2). Let s be such that $\hat{x}'_s \neq \bar{x}'_s$. Then we can find λ_0 and λ_1 in $\mathbb{R}$ such that $\lambda_0 \hat{x}'_s + (1 - \lambda_0)\bar{x}'_s = 0$ and $\lambda_1 \hat{x}'_s + (1 - \lambda_1)\bar{x}'_s > 0$. Note that both $\{x'_s = \lambda_0 \hat{x}'_s + (1 - \lambda_0)\bar{x}'_s \mid s \in S\}$ and $\{x'_s = \lambda_1 \hat{x}'_s + (1 - \lambda_1)\bar{x}'_s \mid s \in S\}$ are also solutions of the system of equations in (2). The fact that $\{x'_s = \lambda_0 \hat{x}'_s + (1 - \lambda_0)\bar{x}'_s \mid s \in S\}$ is a solution and $\lambda_0 \hat{x}'_s + (1 - \lambda_0)\bar{x}'_s = 0$ implies that $s \notin S^{>0}$ (by Lemma 1). On the other hand, the fact that $\{x'_s = \lambda_1 \hat{x}'_s + (1 - \lambda_1)\bar{x}'_s \mid s \in S\}$ is solution and $\lambda_1 \hat{x}'_s + (1 - \lambda_1)\bar{x}'_s > 0$ implies that $s \in S^{>0}$ (by Lemma 1), which is a contradiction. □

Lemma 3. $x'_s = 0$ *iff* $s \in S^{?\text{no}} \cup S^{\text{no}}$.

Proof. Consider the subset of equations from (2)

$$x'_s = \begin{cases} 0 & \text{if } s \in S^{\text{no}} \\ \sum_{s' \in S} \gamma \mathbf{P}(s, s') x'_{s'} & \text{if } s \in S^{?\text{no}} \end{cases} \tag{3}$$

By the definition of $S^{?\text{no}}$, since $p_s(\phi_1 \mathcal{U} \phi_2) = 0$, any state s' that is reachable from s by an edge with non-zero probability is in $S^{?\text{no}} \cup S^{\text{no}}$. Therefore, the set of equations (3) only involve variables in $\{x'_{s'} \mid s' \in S^{?\text{no}} \cup S^{\text{no}}\}$. By Lemma 2, this set of equations has an unique solution. Note that $\{x'_s = 0 \mid s \in S^{?\text{no}} \cup S^{\text{no}}\}$ is a solution to the system of equations in (3). Hence, the unique solution of the system of equation (2) is such that $x'_s = 0$ for all $s \in S^{?\text{no}} \cup S^{\text{no}}$. □

Consider the following system of constraints.

$$x'_s = 0 \text{ iff } x_s = 0 \text{ for all } s \in S \tag{4}$$

where x'_s are variables of (2) and x_s are variables of (1).

Lemma 4. *The system of linear equations in* (1) *and* (2) *has a unique solution given that the constraints in* (4) *hold. Moreover, for this unique solution* $x_s = p_s(\phi_1 \, \mathcal{U} \, \phi_2)$, *for all* $s \in S$.

Proof. The proof is by contradiction. Let $\{x_s = \hat{x}_s \mid s \in S\} \cup \{x'_s = \hat{x}'_s \mid s \in S\}$ and $\{x_s = \bar{x}_s \mid s \in S\} \cup \{x'_s = \bar{x}'_s \mid s \in S\}$ be two solutions of (1) and (2) such that (4) hold. By Lemma 2, for all $s \in S$, $\hat{x}'_s = \bar{x}'_s$. Fix a s such that $\hat{x}_s \neq \bar{x}_s$. We can pick a λ such that $\lambda\hat{x}_s + (1-\lambda)\bar{x}_s = 0$. Note that $\{x_s = \lambda\hat{x}_s + (1-\lambda)\hat{x}_s \mid s \in S\} \cup \{x'_s = \lambda\hat{x}'_s + (1-\lambda)\hat{x}'_s \mid s \in S\}$ is also a solution to the set of constraints. This implies that $\lambda\hat{x}'_s + (1-\lambda)\hat{x}'_s = \hat{x}'_s = \bar{x}'_s = 0$ by the constraints (4). Again by (4), $\hat{x}'_s = \bar{x}'_s = 0$ implies that $\hat{x}_s = \bar{x}_s = 0$, which is a contradiction.

Note that $\{x_s = p_s(\phi_1 \,\mathcal{U}\, \phi_2) \mid s \in S\} \cup \{x'_s = \hat{x}'_s \mid s \in S\}$ is a solution to the system of linear equations in (1) and (2). Moreover, this solution satisfies the constraints in (4). Hence, the solution is an unique solution to the system of linear equations in (1) and (2) such that the constraints in (4) hold. □

Note that the set of constraints (1), (2), and (4) can be written compactly as follows.

$$x_s = \begin{cases} 0 & \text{if } s \in S^{\text{no}} \\ 1 & \text{if } s \in S^{\text{yes}} \\ \sum_{s' \in S} \mathbf{P}(s,s')x_{s'} & \text{if } s \in S^{?} \end{cases} \qquad x'_s = \begin{cases} 0 & \text{if } s \in S^{\text{no}} \\ 1 & \text{if } s \in S^{\text{yes}} \\ \sum_{s' \in S} \gamma\mathbf{P}(s,s')x'_{s'} & \text{if } s \in S^{?} \end{cases} \tag{5}$$

$$\delta_s > 0 \qquad x_s = \delta_s x'_s$$

where for each $s \in S$, we introduce the variable δ_s, such that we can impose the constraint that $x_s = 0$ iff $x'_s = 0$. The satisfiability of the set of constraints (5) can be easily reduced to checking if a formula with existential quantifiers belongs to the theory of reals. The constructed formula is linear in the size of the DTMC.

5 Model Checking UMC

In this section, we reduce the problem of model-checking a UMC to checking the feasibility of a bilinear matrix inequality. (More details about bilinear matrix inequality can be found in Appendix.) In the non-trivial reduction, we introduce a number of auxiliary variables to achieve the goal. Note that a simpler PSPACE algorithm, which avoids the extra auxiliary variables by guessing their values non-deterministically, is possible and is easy to come up from our reduction. However, we believe that the following reduction is important from the perspective of implementation in practice using algorithms to solve bilinear matrix inequalities (BMIs).

Given a UMC $\mathcal{I}$ and a PCTL state formula ϕ, our goal is to check whether $\mathcal{I} \models \phi$. In other words, for every $\mathcal{M} \in [\mathcal{I}]$, $\mathcal{M} \models \phi$. Thus, to check whether $\mathcal{I} \models \phi$, we check if there exists some $\mathcal{M} \in [\mathcal{I}]$ such that $\mathcal{M} \models \neg\phi$. If such an $\mathcal{M}$ does not exist, we conclude that $\mathcal{I} \models \phi$. We will view the problem of discovering whether a $\mathcal{M} \in [\mathcal{I}]$ satisfies $\neg\phi$ as problem of checking the feasibility of a set of bilinear inequality constraints as follows. Each transition probability of the DTMC $\mathcal{M}$ that we are searching for, will be a variable taking a value within the bounds. We will also have variables denoting the satisfaction (or non-satisfaction) of each subformula at each state, and variables denoting the probability of a path subformula being satisfied at each state. Inequality constraints on these variables will ensure that they all have consistent values. We now describe this construction formally.

Let us fix an UMC $\mathcal{I} = (S, s_I, \check{\mathbf{P}}, \hat{\mathbf{P}}, L)$ and a PCTL formula ϕ. Let $\mathcal{M} = (S, s_I, \mathbf{P}, L)$ be an arbitrary Markov chain in $[\mathcal{I}]$.

For every pair of states $s, s' \in S$, let the variable $p_{ss'}$ denote the transition probability from s to s' in $\mathcal{M}$, i.e., $p_{ss'}$ denotes $\mathbf{P}(s, s')$. Since $\mathcal{M}$ is an arbitrary DTMC in $[\mathcal{I}]$, by the definition of UMC, the following constraints hold: For every state $s \in S$, $\sum_{s' \in S} p_{ss'} = 1$ and for every pair of states $s, s' \in S$, $\check{\mathbf{P}}(s, s') \leq p_{ss'} \leq \hat{\mathbf{P}}(s, s')$

Given any PCTL formula ϕ, let us define the set *subfS*(ϕ) (of state sub-formulas) recursively as follows:

$$\begin{array}{rlrl}
\textit{subfS}(a) & = \{a\} & \textit{subfS}(\neg\phi) & = \{\neg\phi\} \cup \textit{subfS}(\phi) \\
\textit{subfS}(\phi_1 \wedge \phi_2) & = \{\phi_1 \wedge \phi_2\} \cup \textit{subfS}(\phi_1) \cup \textit{subfS}(\phi_2) & \textit{subfS}(\mathcal{P}_{\bowtie p}(\psi)) & = \{\mathcal{P}_{\bowtie p}(\psi)\} \cup \textit{subfS}(\psi) \\
\textit{subfS}(\phi_1 \, \mathcal{U} \, \phi_2) & = \textit{subfS}(\phi_1 \wedge \neg\phi_2) & \textit{subfS}(\mathbf{X}\phi) & = \textit{subfS}(\phi)
\end{array}$$

Given a state $s \in S$ and any formula $\phi' \in$ *subfS*(ϕ), either $s \models_{\mathcal{M}} \phi'$ or $s \not\models_{\mathcal{M}} \phi'$. For each $s \in S$ and each $\phi' \in$ *subfS*(ϕ), let the variable $t_s^{\phi'}$ be such that $t_s^{\phi'} = 1$ iff $s \models_{\mathcal{M}} \phi'$; and, $t_s^{\phi'} = 0$ iff $s \not\models_{\mathcal{M}} \phi'$. Following the definition of the various logical operators in PCTL, we can set up a set of constraints among these variables such that for any $\mathcal{M} \in [\mathcal{I}]$, the values taken by these variables is consistent with their intended semantic interpretation. We introduce the following additional variables to aid in setting up these constraints. For every state $s \in S$ and $\phi' \in$ *subfS*(ϕ), let the auxiliary variables $f_s^{\phi'}$, and $u_s^{\phi'}$ be such that $t_s^{\phi'} = 1 \iff f_s^{\phi'} = 0 \iff u_s^{\phi'} = 1$ and $t_s^{\phi'} = 0 \iff f_s^{\phi'} = 1 \iff u_s^{\phi'} = -1$ Clearly, $t_s^{\phi'}$, $f_s^{\phi'}$, and $u_s^{\phi'}$ are related by the following set of constraints:

$$t_s^{\phi'} f_s^{\phi'} = 0 \qquad t_s^{\phi'} + f_s^{\phi'} = 1 \qquad 2t_s^{\phi'} = u_s^{\phi'} + 1$$

For every formula $\phi' \in$ *subfS*(ϕ) of the form $\mathcal{P}_{\bowtie p}(\psi)$ and for every state $s \in S$, let p_s^{ψ} be the variable such that p_s^{ψ} denotes $\mathit{Prob}\{\pi \in \mathit{Path}(s) \mid \pi \models_{\mathcal{M}} \psi\}$ in $\mathcal{M}$.

For each state $s \in S$ and for each $\phi' \in$ *subfS*(ϕ) exactly one of the following constraints hold depending on the form of ϕ':

$$\begin{array}{llll}
t_s^{\phi'} = 1 & \text{if } \phi' = a \in L(s) & t_s^{\phi'} = 0 & \text{if } \phi' = a \notin L(s) \\
t_s^{\phi'} = 1 - t_s^{\phi_1} & \text{if } \phi' = \neg\phi_1 & t_s^{\phi_1} t_s^{\phi_2} = t_s^{\phi'} & \text{if } \phi' = \phi_1 \wedge \phi_2 \\
u_s^{\phi'} p_s^{\psi} \geq u_s^{\phi'} p + \delta f_s^{\phi'} & \text{if } \phi' = \mathcal{P}_{\geq p}(\psi) & u_s^{\phi'} p_s^{\psi} \geq u_s^{\phi'} p + \delta t_s^{\phi'} & \text{if } \phi' = \mathcal{P}_{> p}(\psi) \\
u_s^{\phi'} p_s^{\psi} + \delta f_s^{\phi'} \leq u_s^{\phi'} p & \text{if } \phi' = \mathcal{P}_{\leq p}(\psi) & u_s^{\phi'} p_s^{\psi} + \delta t_s^{\phi'} \leq u_s^{\phi'} p & \text{if } \phi' = \mathcal{P}_{< p}(\psi)
\end{array}$$

where δ is slack variable that is required to be strictly greater than 0.

Note that the above constraints do not reflect the fact that for each $\phi' \in$ *subfS*(ϕ) of the form $\mathcal{P}_{\bowtie p}(\psi)$, p_s^{ψ} denotes $\mathit{Prob}\{\pi \in \mathit{Path}(s) \mid \pi \models_{\mathcal{M}} \psi\}$. To set up such constraints, we introduce the set *subfP*(ϕ) (of path sub-formulas) as follows:

$$\begin{array}{rlrl}
\textit{subfP}(a) & = \emptyset & \textit{subfP}(\neg\phi) & = \textit{subfP}(\phi) \\
\textit{subfP}(\phi_1 \wedge \phi_2) & = \textit{subfP}(\phi_1) \cup \textit{subfP}(\phi_2) & \textit{subfP}(\mathcal{P}_{\bowtie p}(\psi)) & = \{\psi\} \cup \textit{subfP}(\psi) \\
\textit{subfP}(\phi_1 \, \mathcal{U} \, \phi_2) & = \textit{subfP}(\phi_1) \cup \textit{subfP}(\phi_2) & \textit{subfP}(\mathbf{X}\phi) & = \textit{subfP}(\phi)
\end{array}$$

Thus for all sub-formula of ϕ of the form $\mathcal{P}_{\bowtie p}(\psi)$, *subfP*$(\phi)$ contains ψ.

For any $\psi \in$ *subfP*(ϕ) of the form $\mathbf{X}\phi_1$ and for each $s \in S$ the following constraint holds:

$$p_s^{\psi} = \sum_{s' \in S} p_{ss'} t_{s'}^{\phi_1}$$

For each $\psi \in \textit{subfS}(\phi)$ of the form $\phi_1 \, \mathcal{U} \, \phi_2$ and $s \in S$ the following constraints hold.

$$p_s^{\psi} = t_s^{\phi_2} + t_s^{\phi_1 \wedge \neg \phi_2} w_s^{\psi} \qquad\qquad w_s^{\psi} = \sum_{s' \in S} p_{ss'} p_s^{\psi}$$

As in simple DTMC, if we consider the above constraints only, then we may not have unique solution for certain p_s^{ψ}. Therefore, we fix a γ such that $0 < \gamma < 1$. Then, as in simple DTMC model-checking, for each $\psi \in \textit{subfP}(\phi)$ of the form $\phi_1 \, \mathcal{U} \, \phi_2$ and $s \in S$, we introduce the variables $p_s^{'\psi}$ and $w_s^{'\psi}$, such that the following constraints hold.

$$p_s^{'\psi} = t_s^{\phi_2} + t_s^{\phi_1 \wedge \neg \phi_2} w_s^{'\psi} \qquad\qquad w_s^{'\psi} = \gamma \sum_{s' \in S} p_{ss'} p_s^{'\psi}$$

We want $p_s^{\psi} = 0$ if $p_s^{'\psi} = 0$. To ensure this, for each $\psi \in \textit{subfP}(\phi)$ of the form $\phi_1 \, \mathcal{U} \, \phi_2$ and $s \in S$, we introduce the auxiliary variable δ_s^{ψ} and ensure that the following constraint hold.

$$\delta_s^{\psi} > 0 \qquad p_s^{\psi} = \delta_s^{\psi} p_s^{'\psi}$$

Let $V(\mathcal{I}, \phi) = \{\delta\} \cup \bigcup_{s,s' \in S} \{p_{ss'}\} \cup \bigcup_{s \in S, \phi' \in \textit{subfS}(\phi)} \{t_s^{\phi'}, f_s^{\phi'}, u_s^{\phi'}\} \cup \bigcup_{s \in S, \psi \in \textit{subfP}(\phi)} \{p_s^{\psi}, w_s^{\psi}, p_s^{'\psi}, w_s^{'\psi}, \delta_s^{\psi}\}$ denote the set of variables over which the above constraints are described and let $C(\mathcal{I}, \phi)$ denote the above set of constraints.

Lemma 5. *For every solution $I \colon V(\mathcal{I}, \phi) \to \mathbb{R}$ of $C(\mathcal{I}, \phi)$, there exists a DTMC $\mathcal{M} = (S, s_I, \mathbf{P}, L) \in [\mathcal{I}]$ such that the following holds:*

1. *$I(p_{ss'}) = \mathbf{P}(s, s')$ for any $s, s' \in S$*
2. *$t_s^{\phi'}, f_s^{\phi'} \in \{0, 1\}$ and $u_s^{\phi'} \in \{-1, 1\}$ for any $s \in S$ and $\phi' \in \textit{subfS}(\phi)$*
3. *$t_s^{\phi'} = 1 \wedge f_s^{\phi'} = 0 \wedge u_s^{\phi'} = 1$ iff $s \models_{\mathcal{M}} \phi'$ for any $s \in S$ and $\phi' \in \textit{subfS}(\phi)$*
4. *$t_s^{\phi'} = 0 \wedge f_s^{\phi'} = 1 \wedge u_s^{\phi'} = -1$ iff $s \models_{\mathcal{M}} \phi'$ for any $s \in S$ and $\phi' \in \textit{subfS}(\phi)$*
5. *$p_s^{\psi} = \textit{Prob}\{\pi \in \textit{Path}(s) \mid \pi \models_{\mathcal{M}} \psi\}$ for any $\psi \in \textit{subfP}(\phi)$*

The proof follows from the observations made while setting up the constraints. An immediate consequence of the Lemma 5 is the following theorem.

Theorem 1. *If there exists a solution I of $C(\mathcal{I}, \phi)$ such that $I(t_{s_I}^{\phi}) = 1$, then there exists an $\mathcal{M} \in [\mathcal{I}]$ such that $\mathcal{M} \models \phi$.*

In order to check if $\mathcal{I} \models \phi$, the model checking algorithm sets up the constraints $C(\mathcal{I}, \neg\phi)$ and checks its feasibility. Clearly, checking the feasibility of $C(\mathcal{I}, \neg\phi)$ is equivalent to checking if a sentence with existential quantifiers is valid for the reals; the size of the sentence is polynomial in the size of the UMC. However, the constraints $C(\mathcal{I}, \neg\phi)$ are bilinear constraints, and we need to satisfy the conjunction of all these constraints (not an arbitrary boolean function). The feasibility of such constraints can be more efficiently checked viewing them as *bilinear matrix inequalities* (BMIs) for which algorithms [10, 9] and tools [18] have been developed. (More details about bilinear matrix inequality can be found in Appendix.) We also observe that to prove that the model checking problem can be solved in PSPACE, we could have constructed a simpler set of constraints by first guessing the values of the variables $t_s^{\phi'}, u_s^{\phi'}$, and $f_s^{\phi'}$ for the subformulas ϕ', and then solving the constraints resulting from those guesses; since NPSPACE = PSPACE, we can obtain a deterministic algorithm from this. However, we believe that in practice solving this single BMI presented here will be more efficient than solving the exponentially many simpler BMIs that this alternative approach would yield.

5.1 Complexity of Model-checking UMC

We showed that the model-checking problem for UMC can be reduced to checking the validity of a formula in the existential theory of the reals. Therefore, the model-checking problem of UMC is in PSPACE.

We next demonstrate the intractability of the model checking problem for UMC by reducing the satisfiability and validity of propositional boolean formulas to the model checking problem. Consider a propositional boolean formula φ over the propositions $\{p_1, \ldots, p_m\}$.

We consider the UMC $\mathcal{I} = (S, s_I, \check{\mathbf{P}}, \hat{\mathbf{P}}, L)$ where

- $S = \{s_I, s_1, \ldots, s_m, s_\perp\}$
- $L(s_I) = L(s_\perp) = \{\}$, $L(s_i) = \{p_i\}$ for each $1 \leq i \leq m$
- $\check{\mathbf{P}}(s_I, s_i) = 0$ and $\hat{\mathbf{P}}(s_I, s_i) = 1/m$ for all $1 \leq i \leq m$
- $\check{\mathbf{P}}(s_I, s_\perp) = 0$ and $\hat{\mathbf{P}}(s_I, s_\perp) = 1$
- $\check{\mathbf{P}}(s_i, s_i) = \hat{\mathbf{P}}(s_i, s_i) = 1$ for all $1 \leq i \leq m$
- $\check{\mathbf{P}}(s_i, s_j) = \hat{\mathbf{P}}(s_i, s_j) = 0$ for all $1 \leq i \leq m$ and $1 \leq j \leq m$ and $i \neq j$
- $\check{\mathbf{P}}(s_\perp, s_\perp) = \hat{\mathbf{P}}(s_\perp, s_\perp) = 1$

We consider the PCTL formula ϕ' obtained from ϕ by syntactically replacing every occurrence of p_i in ϕ by $\mathcal{P}_{>\frac{1}{2m}}(\mathbf{X}p_i)$ for $1 < i < m$.

Lemma 6. *φ is satisfiable iff $\mathcal{I} \not\models \neg\phi$; φ is valid iff $\mathcal{I} \models \phi$.*

Proof. Suppose φ is satisfiable and let a be the satisfying assignment. Consider the DTMC $\mathcal{M}^a$, where $\mathbf{P}(s_I, s_i) = \frac{1}{2m}$ if $a(p_i) = \text{false}$ and $\mathbf{P}(s_I, s_i) = \frac{1}{m+1}$ if $a(p_i) = \text{true}$; $\mathbf{P}(s_I, s_\perp)$ is thus determined by this assignment. It is easy to see that $\mathcal{M}^a \in [\mathcal{I}]$ and $\mathcal{M}^a \models \phi$. Similarly, if $\mathcal{M} \in [\mathcal{I}]$ such that $\mathcal{M} \models \phi$, then we can construct a satisfying assignment for φ: $a(p_i) = \text{false}$ if $\mathbf{P}(s_I, s_i) \leq \frac{1}{2m}$ and $a(p_i) = \text{true}$ if $\mathbf{P}(s_I, s_i) > \frac{1}{2m}$. These observations also imply that φ is valid iff $\mathcal{I} \models \phi$.

Since the satisfiability of general propositional boolean formulas is NP-hard and the validity of general propositional boolean formulas is co-NP-hard [13], the lower bounds follow immediately from Lemma 6.

Theorem 2. *The model checking problem for UMC with respect to PCTL is NP-hard and co-NP-hard.*

6 Model-checking IMDP

We consider the problem of model checking IMDPs in this section. We will solve the problem by showing that we can reduce IMDP model checking to model checking (classical) a Markov Decision Process (MDP) [4, 23]. Before presenting this reduction we recall some basic properties of the feasible solutions of a linear program and the definition of an MDP.

6.1 Linear Programming

Consider an IMDP $\mathcal{I} = (S, s_I, \check{\mathbf{P}}, \hat{\mathbf{P}}, L)$. For a given $s \in S$, let *IE*(s) be the following set of inequalities over the variables $\{p_{ss'} \mid s' \in S\}$:

$$\sum_{s' \in S} p_{ss'} = 1 \qquad \check{\mathbf{P}}(s, s') \leq p_{ss'} \leq \hat{\mathbf{P}}(s, s') \text{ for all } s' \in S$$

Definition 4. *A map $\theta^s \colon S \to [0, 1]$ is called a* basic feasible solution *(BFS) to the above set of inequalities IE(s) iff $\{p_{ss'} = \theta^s(s') \mid s' \in S\}$ is a solution of IE(s) and there exists a set $S' \subseteq S$ such that $|S'| \geq |S| - 1$ and for all $s' \in S'$ either $\theta^s(s') = \check{\mathbf{P}}(s, s')$ or $\theta^s(s') = \hat{\mathbf{P}}(s, s')$.*

Let Θ^s be the set of all BFS of *IE*(s). The set of BFS of linear program have the special property that every other feasible solution can be expressed as a linear combination of basic feasible solutions. This is the content of the next proposition.

Proposition 1. *Let $\{p_{ss'} = \bar{p}_{ss'} \mid s' \in S\}$ be some solution of IE(s). There there are $0 \leq \alpha_{\theta^s} \leq 1$ for all $\theta^s \in \Theta^s$, such that*

$$\bar{p}_{ss'} = \textstyle\sum_{\theta^s \in \Theta^s} \alpha_{\theta^s} \theta^s(s') \textit{ for all } s' \in S \qquad \textit{and} \qquad \textstyle\sum_{s \in S} \alpha_{\theta^s} = 1$$

Lemma 7. *The number of basic feasible solutions of IE(s) in the worst case can be $O(|S| 2^{|S|-1})$.*

6.2 Markov Decision Processes (MDP)

A Markov decision process (MDP) is a Markov chain that has non-deterministic transitions, in addition to the probabilistic ones. In this section we formally introduce this model along with some well-known observations about them.

Definition 5. *If S is the set of states of a system, a* next-state probability distribution *is a function $\mu : S \to [0, 1]$ such that $\sum_{s \in S} \mu(s) = 1$. For $s \in S$, $p(s)$ represents the probability of making a direct transition to s from the current state.*

Definition 6. *A Markov decision Process (MDP) is a 4-tuple $\mathcal{D} = (S, s_I, \tau, L)$, where*

1. *S is a finite set of states,*
2. *$s_I \in S$ is the initial state,*
3. *$L \colon S \to 2^{\mathrm{AP}}$ is a labeling function that maps states to sets of atomic propositions from a set* AP*,*
4. *τ is a function which associates to each $s \in S$ a finite set $\tau(s) = \{\mu_1^s, \ldots, \mu_{k_s}^s\}$ of* next-state probability distributions *for transitions from s.*

A *path* π in an MDP $\mathcal{D} = (S, s_I, \tau, L)$ is a non-empty sequence of the form $s_0 \xrightarrow{\mu_1} s_1 \xrightarrow{\mu_2} \ldots$, where $s_i \in S$, $\mu_{i+1} \in \tau(s_i)$, and $\mu_{i+1}(s_{i+1}) > 0$ for all $i \geq 0$. A path can be either finite or infinite. We use π_{fin} to denote a finite path. Let *last*(π_{fin}) be the last state in the finite path π_{fin}. As in DTMC, we denote the i^{th} state in a path π by $\pi[i] = s_i$. We let *Path*(s) and *Path*$_{\mathrm{fin}}(s)$ be the set of all infinite and finite paths, respectively, starting at state s. To associate a probability measure with the paths, we resolve the non-deterministic choices by a randomized *adversary*, which is defined as follows:

Definition 7. *A randomized* adversary A *of an MDP* $\mathcal{D}$ *is a function mapping every finite path* π_{fin} *of* $\mathcal{D}$ *and an element of the set* $\tau(\mathit{last}(\pi_{\text{fin}}))$ *to* $[0,1]$*, such that for a given finite path* π_{fin} *of* $\mathcal{D}$*,* $\sum_{\mu \in \tau(\mathit{last}(\pi_{\text{fin}}))} A(\pi_{\text{fin}}, \mu) = 1$*. Let* $\mathcal{A}_{\mathcal{D}}$ *denote the set of all possible randomized adversaries of the MDP* $\mathcal{D}$*. Let* $\mathit{Path}^A(s)$ *denote the subset of* $\mathit{Path}(s)$ *which corresponds to an adversary* A*.*

The behavior of an MDP under a given randomized adversary is purely probabilistic. If an MDP has evolved to the state s after starting from the state s_I and following the finite path π_{fin}, then it chooses the next-state distribution $\mu^s \in \tau(s)$ with probability $A(\pi_{\text{fin}}, \mu^s)$. Then it chooses the next state s' with probability $\mu^s(s')$. Thus the probability that a direct transition to s' takes place is $\sum_{\mu^s \in \tau(s)} A(\pi_{\text{fin}}, \mu^s)\mu^s(s')$. Thus as for IMDPs, one can define DTMC $\mathcal{D}^A$ that captures the probabilistic behavior of MDP $\mathcal{D}$ under adversary A and also associate a probability measure on execution paths. Given a MDP $\mathcal{D}$ and a PCTL formula φ, we can define when $\mathcal{D} \models \varphi$ in a way analogous to the IMDPs (see Figure 2).

6.3 The Reduction

We are now ready to describe the model checking algorithm for IMDPs. Consider an IMDP $\mathcal{I} = (S, s_I, \check{\mathbf{P}}, \hat{\mathbf{P}}, L)$. Recall from Section 6.1, we can describe the transition probability distributions from state s that satisfy the range constraints as the feasible solutions of the linear program $\mathit{IE}(s)$. Furthermore, we denote by Θ^s is the set of all BFS of $\mathit{IE}(s)$. Define the following MDP $\mathcal{D} = (S', s'_I, \tau, L')$ where $S' = S$, $s'_I = s_I$, $L' = L$, and for all $s \in S$, $\tau(s) = \Theta^s$. Observe that $\mathcal{D}$ is exponentially sized in $\mathcal{I}$, since $\tau(s)$ is exponential (see Lemma 7).

The main observation behind the reduction is that the MDP $\mathcal{D}$ “captures” all the possible behaviors of the IMDP $\mathcal{I}$. This is the formal content of the next proposition.

Proposition 2. *For any adversary* A *for* $\mathcal{I}$*, we can define a randomized adversary* A' *such that* $\mathit{Prob}_s^{\mathcal{I}^A} = \mathit{Prob}_s^{\mathcal{D}^{A'}}$ *for every* s*, where* $\mathit{Prob}_s^{X^A}$ *is measure on paths from* s *defined by machine* X *under* A*. Similarly for every adversary* A *for* $\mathcal{D}$*, we can find an adversary* A' *for* $\mathcal{I}$ *that defines the same probability measure on paths.*

Proof. Consider an adversary A for $\mathcal{I}$. For a path π_{fin} let $A(\pi_{\text{fin}}) = \mu \in \mathit{Steps}(\mathit{last}(\pi_{\text{fin}}))$. We know from Proposition 1, that there are α_{θ^s} for $\theta^s \in \Theta^s$ such that

$$\mu(s') = \textstyle\sum_{\theta^s \in \Theta^s} \alpha_{\theta^s} \theta^s(s') \text{ for all } s' \in S \qquad \text{and} \qquad \textstyle\sum_{s \in S} \alpha_{\theta^s} = 1$$

We now define $A'(\pi_{\text{fin}}, \theta^s) = \alpha_{\theta^s}$. It is straightforward to see that $\mathit{Prob}_s^{\mathcal{I}^A} = \mathit{Prob}_s^{\mathcal{D}^{A'}}$. The converse direction also can be proved similarly. □

An important consequence of the above observation is the following main theorem.

Theorem 3. *For any PCTL formula* φ*,* $\mathcal{I} \models \varphi$ *iff* $\mathcal{D} \models \varphi$*.*

Thus, in order to model check IMDP $\mathcal{I}$, we can model check the MDP $\mathcal{D}$ for which algorithms are known [4, 23]. The algorithms for MDP run in time (and space) which is polynomial in the size of the MDP. Thus, if we directly model check $\mathcal{D}$ we get an EXPTIME model checking algorithm for $\mathcal{I}$. However, we can improve this to get a PSPACE algorithm. The reason for this is that it is known that as far as model checking MDPs is concerned, we can restrict our attention to *deterministic, memoryless* adversaries, i.e., adversaries that always pick the same single non-deterministic choice whenever a state is visited.

Proposition 3 ([4, 23]). *Let $\mathcal{A}_{\rm det}$ be the set of deterministic, memoryless adversaries for MDP $\mathcal{D}$, i.e., for all $A \in \mathcal{A}_{\rm det}$, $A(s, \mu) = 1$ for exactly one $\mu \in \tau(s)$. Consider a PCTL formula $\varphi = \mathcal{P}_{\bowtie p}(\psi)$ such that the truth or falsity of every subformula of ψ in every state of $\mathcal{D}$ is already determined. Then $\mathcal{D} \models \varphi$ iff $\mathcal{D}^A \models \varphi$ for all $A \in \mathcal{A}_{\rm det}$.*

For every subformula of the form $\mathcal{P}_{\bowtie p}(\psi)$, our model checking algorithm, will model check each of the DTMCs $\mathcal{D}^A$, where A is a deterministic, memoryless adversary. This will give us the desired PSPACE algorithm.

Theorem 4. *The model-checking algorithm for IMDP is in PSPACE.*

Proof. From Lemma 7, we know that the total number of BFSs is $O(|S|2^{|S|-1})$. Hence the total number of DTMCs $\mathcal{D}^A$ for $A \in \mathcal{A}_{\rm det}$ is $O(|S|^{|S|}2^{|S|^2-|S|})$. By reusing space for every subformula $\mathcal{P}_{\bowtie p}(\psi)$, all of these model checking problems can be solved in PSPACE. □

6.4 Iterative Algorithm

The above PSPACE algorithm is computationally expensive for large IMDPs. Therefore, we propose an alternative iterative algorithm motivated by a similar algorithm in [2].

The iterative model checking algorithm for PCTL over IMDPs works exactly as for DTMCs with the exception of handling of $\mathcal{P}_{\bowtie p}(\psi)$. For these, we need to check if $p_s^A(\psi) = \mathit{Prob}_s^A(\{\pi \in \mathit{Path}^A(s) \mid \pi \models \psi\})$ satisfies the bound $\bowtie p$ for all adversaries $A \in \mathcal{A}_{\mathcal{I}}$. Let $p_s^{\max}(\psi)$ and $p_s^{\min}(\psi)$ be the *minimum* or *maximum* probability, respectively, for all adversaries $A \in \mathcal{A}_{\mathcal{I}}$, i.e.,

$$p_s^{\max}(\psi) \stackrel{\rm def}{=} \sup\nolimits_{A \in \mathcal{A}_{\mathcal{I}}}[p_s^A(\psi)], \quad p_s^{\min}(\psi) \stackrel{\rm def}{=} \inf\nolimits_{A \in \mathcal{A}_{\mathcal{I}}}[p_s^A(\psi)].$$

Then if $\bowtie \in \{<, \leq\}$,

$$\text{Sat}(\mathcal{P}_{\bowtie p}(\psi)) = \{s \in S \mid p_s^{\max}(\psi) \bowtie p\}$$

and if $\bowtie \in \{>, \geq\}$,

$$\text{Sat}(\mathcal{P}_{\bowtie p}(\psi)) = \{s \in S \mid p_s^{\min}(\psi) \bowtie p\}$$

We next describe how to compute the values $p_s^{\max}(\psi)$ and $p_s^{\min}(\psi)$ for $\psi = \mathbf{X}\phi$ and $\psi = \phi_1 \mathcal{U} \phi_2$. Recall that Θ^s is the set of all BFS of *IE*(s). It can be shown following [2] that $p_s^{\max} = \lim_{n\to\infty} p_s^{\max(n)}$ where:

$$p_s^{\max(n)} = \begin{cases} 1 & \text{if } s \in S^{\rm yes} \\ 0 & \text{if } s \in S^{\rm no} \\ 0 & \text{if } s \in S^{?} \text{ and } n = 0 \\ \max_{\{\bar{p}_{ss'} \mid s' \in S\} \in \Theta^s} \left\{ \sum_{s' \in S} \bar{p}_{ss'} . p_{s'}^{\max(n-1)} \right\} & \\ & \text{if } s \in S^{?} \text{ and } n > 0 \end{cases}$$

and $p_s^{\min} = \lim_{n\to\infty} p_s^{\min(n)}$ where:

$$p_s^{\min(n)} = \begin{cases} 1 & \text{if } s \in S^{\text{yes}} \\ 0 & \text{if } s \in S^{\text{no}} \\ 0 & \text{if } s \in S^{?} \text{ and } n = 0 \\ \min_{\{\bar{p}_{ss'} | s' \in S\} \in \Theta^s} & \left\{ \sum_{s' \in S} \bar{p}_{ss'}.p_{s'}^{\min(n-1)} \right\} \\ & \text{if } s \in S^{?} \text{ and } n > 0 \end{cases}$$

Note that although the size of Θ^s can be $O(|S|2^{|S|-1})$ (by Lemma 7), the computation of the expressions

$$\max_{\{\bar{p}_{ss'} | s' \in S\} \in \Theta^s} \left\{ \sum_{s' \in S} \bar{p}_{ss'}.p_{s'}^{\max(n-1)} \right\} \text{ or } \min_{\{\bar{p}_{ss'} | s' \in S\} \in \Theta^s} \left\{ \sum_{s' \in S} \bar{p}_{ss'}.p_{s'}^{\min(n-1)} \right\} \tag{6}$$

can be done in $O(|S|)$ time as follows:

We consider the ordering $s_1, s_2, \ldots, s_{|S|}$ of the states of S such that $p_{s_1}^{\max(n-1)}, p_{s_2}^{\max(n-1)}, \ldots, p_{s_{|S|}}^{\max(n-1)}$ is in descending order. Then the following result holds.

Lemma 8.

a) There exists an $1 \le i \le |S|$ *such that* $\{\hat{\mathbf{P}}(s, s_1), \ldots, \hat{\mathbf{P}}(s, s_{i-1}), q, \check{\mathbf{P}}(s, s_{i+1}), \ldots, \check{\mathbf{P}}(s, s_{|S|})\}$ *is a BFS of IE*(s)*, where* $q = 1 - \sum_{1 \le j \le (i-1)} \hat{\mathbf{P}}(s, s_j) - \sum_{(i+1) \le j \le |S|} \check{\mathbf{P}}(s, s_j)$.

b) and for that i

$$\max_{\{\bar{p}_{ss'} | s' \in S\} \in \Theta^s} \left\{ \sum_{s' \in S} \bar{p}_{ss'}.p_{s'}^{\max(n-1)} \right\} = p_{s_i}^{\max(n-1)}.q$$
$$+ \sum_{1 \le j \le (i-1)} p_{s_j}^{\max(n-1)}.\hat{\mathbf{P}}(s, s_j) + \sum_{(i+1) \le j \le |S|} p_{s_j}^{\max(n-1)}.\check{\mathbf{P}}(s, s_j)$$

Proof.
a) Let i_0 be defined as follows:

$$i_0 = \min\{i \mid \sum_{j=1}^{i} \hat{\mathbf{P}}(s, s_j) + \sum_{j=i+1}^{|S|} \check{\mathbf{P}}(s, s_j) \ge 1\}$$

Observe that such an i_0 must exist if the IMDP is well-defined. Consider the solution $\{\hat{\mathbf{P}}(s, s_1), \ldots, \hat{\mathbf{P}}(s, s_{i_0-1}), q, \check{\mathbf{P}}(s, s_{i_0+1}), \ldots, \check{\mathbf{P}}(s, s_{|S|})\}$ where $q = 1 - \sum_{1 \le j \le (i_0-1)} \hat{\mathbf{P}}(s, s_j) - \sum_{(i_0+1) \le j \le |S|} \check{\mathbf{P}}(s, s_j)$. This solution is a BFS of *IE*(s).

b) Let $\{\bar{p}_{ss_1}, \ldots, \bar{p}_{ss_{|S|}}\}$ be any solution (it may be BFS or not) of *IE*(s). Then by simple algebraic simplification it can be shown that

$$\sum_{1 \le j \le (i-1)} p_{s_j}^{\max(n-1)}.\hat{\mathbf{P}}(s, s_j) + p_{s_i}^{\max(n-1)}.q + \sum_{(i+1) \le j \le |S|} p_{s_j}^{\max(n-1)}.\check{\mathbf{P}}(s, s_j) \ge \sum_{s' \in S} \bar{p}_{ss'}.p_{s'}^{\max(n-1)}$$

given the fact that $p_{s_1}^{\max(n-1)} \ge p_{s_2}^{\max(n-1)} \ge \ldots \ge p_{s_{|S|}}^{\max(n-1)}$, and $\check{\mathbf{P}}(s, s') \le \bar{p}_{ss'} \le \hat{\mathbf{P}}(s, s')$ for all $s' \in S$. □

Similarly, if we consider the ordering $s_1, s_2, \ldots, s_{|S|}$ of the states of S such that $p_{s_1}^{\min(n-1)}, p_{s_2}^{\min(n-1)}, \ldots, p_{s_{|S|}}^{\min(n-1)}$ is in ascending order, then the above Lemma holds with max replaced by min.

The expressions (6) can be computed in $O(|S|)$ time by finding an i as in Lemma 8.

6.5 Lower Bound for IMDP model-checking

We will show that the model checking problem for DTMCs is P-hard. Since DTMCs are a special case of IMDPs the P-time lower bound will follows.

To show this we will reduce 3-CNF value, which known to be P-hard [13], to the problem of model checking DTMCs. Recall that 3-CNF value is the problem where we are given a 3-CNF formula φ and an assignment a to each of the variables, and are asked whether φ evaluates to true or false under the assignment. The reduction is very similar to the one given Section 5.1.

Consider $\varphi = \bigwedge_{i\in[1,n]}(l_{1i} \vee l_{2i} \vee l_{3i})$ over the propositions $\{p_1, \ldots, p_m\}$, where each $l_{jj} \in \{p_1, \neg p_1, \ldots, p_m, \neg p_m\}$ for $1 \le i \le 3$ and $1 \le j \le n$. Construct the DTMC $\mathcal{M} = (S, s_I, \mathbf{P}, L)$ where

- $S = \{s_I, s_1, \ldots, s_m, s_\perp\}$
- $L(s_I) = L(s_\perp) = \{\}$, $L(s_i) = \{p_i\}$ for each $1 \le i \le m$
- $\mathbf{P}(s_i, s_i) = \mathbf{P}(s_\perp, s_\perp) = 1$ for all $1 \le i \le m$
- $\mathbf{P}(s_I, s_i) = \frac{1}{2m}$ if $a(p_i) = \text{false}$ and $\mathbf{P}(s_I, s_i) = \frac{1}{m}$ if $a(p_i) = \text{true}$.
- $\mathbf{P}(s_I, s_\perp) = 1 - \sum_i \mathbf{P}(s_I, s_i)$

Let $\phi = \bigwedge_{i\in[1,n]}(\phi_{1i} \vee \phi_{2i} \vee \phi_{3i})$, where if $l_{ji} = p_k$ then $\phi_{ji} = \mathcal{P}_{>\frac{1}{2m}}(\mathbf{X} p_k)$ and if $l_{ji} = \neg p_k$ then $\phi_{ji} = \mathcal{P}_{\le\frac{1}{2m}}(\mathbf{X} p_k)$. Analogous to Lemma 6, one can see that $\mathcal{M}$ satisfies ϕ if and only φ is true under the assignment a. The formal proof is skipped.

7 Conclusion

We have investigated the PCTL model checking problem for two semantic interpretations of IDTMCs, namely UMC and IMDP. We proved the upper bounds and the lower bounds on the complexity of the model checking problem for these models. Our bounds however are not tight. Finding tight lower and upper bounds for these model-checking problems is an interesting open problem.

Acknowledgment

We would like to thank anonymous referees and Timo Latvala for providing valuable comments. This work is supported in part by the ONR Grant N00014-02-1-0715, the NSF Grants NSF CNS 05-09321, NSF CCF 04-29639, NSF CCF 04-48178, and the Motorola Grant Motorola RPF #23.

References

1. A. Aziz, V. Singhal, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. In *Proc. of Computer Aided Verification*, volume 939, pages 155–165, 1995.
2. C. Baier. On algorithmic verification methods for probabilistic systems. Habilitation Thesis. Fakultät für Mathematik and Informatik, Universität Mannheim, 1998.

3. C. Baier and M. Z. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.
4. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proceedings of 15th Conference on the Foundations of Software Technology and Theoretical Computer Science (FSTTCS'95)*, volume 1026 of *LNCS*.
5. L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21:1–46, 1989.
6. J. Canny. Some algebraic and geometric computations in PSPACE. In *20th ACM Symposium on Theory of Computing (STOC'88)*, pages 460–467, 1988.
7. C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. In *Proceedings of the seventeenth international colloquium on Automata, languages and programming*, pages 336–349, 1990.
8. C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of ACM*, 42(4):857–907, 1995.
9. M. Fukuda and M. Kojima. Branch-and-cut algorithms for the bilinear matrix inequality eigenvalue problem. *Comput. Optim. Appl.*, 19(1):79–105, 2001.
10. K. C. Goh, M. G. Safonov, and G. P. Papavassilopoulos. Global optimization for the biaffine matrix inequality problem. *Journal of Global Optimization*, 7:365–380, 1995.
11. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
12. R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics*. Macmillan, New York, NY, USA, fourth edition, 1978.
13. J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, Reading, MA, 1979.
14. B. Jonsson and K. G. Larsen. Specification and refinement of probabilistic processes. In *Proceedings of the IEEE Symposium on Logic in Computer Science*, pages 266–277, 1991.
15. J. Kemeny, J. Snell, and A. Knapp. *Denumerable Markov chains*. Springer, 1976.
16. I. O. Kozine and L. V. Utkin. Interval-valued finite markov chains. *Reliable Computing*, 8(2):97–113, 2002.
17. V. P. Kuznetsov. Interval statistical models. *Radio and Communication*, 1991.
18. PENbmi. `http://www.penopt.com/`.
19. M. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley, New York, 1994.
20. J. Renegar. A faster pspace algorithm for deciding the existential theory of the reals. In *29th Annual IEEE Symposium on Foundations of Computer Science*, pages 291–295, 1988.
21. J. Rutten, M. Kwiatkowska, G. Norman, and D. Parker. *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems*, volume 23 of *CRM Monograph Series*. American Mathematical Society, 2004.
22. M. G. Safonov, K. C. Goh, and J. H. Ly. Controller synthesis via bilinear matrix inequalities. In *Proc. of American Control Conference*, pages 45–49. IEEE, 1994.
23. R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, MIT, 1995.
24. R. Segala and N. A. Lynch. Probabilistic simulations for probabilistic processes. In *International Conference on Concurrency Theory*, pages 481–496, 1994.
25. K. Sen, M. Viswanathan, and G. Agha. Statistical model checking of black-box probabilistic systems. In *16th conference on Computer Aided Verification (CAV'04)*, volume 3114 of *LNCS*, pages 202–215, 2004.
26. O. Toker and H. Özbay. On the NP-hardness of solving bilinear matrix in equalities and simultaneous stabilization with static output feedback. In *Proc. of American Control Conference*, 1995.
27. M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *26th Annual Symposium on Foundations of Computer Science*, pages 327–338. IEEE, 1985.
28. P. Walley. Measures of uncertainty in expert systems. *Artificial Intelligence*, 83:1–58, 1996.

A Motivation

We provide examples to show that UMC and IMDP arise as natural models in many realistic situations.

UMC in practice. Consider an Internet router having a finite buffer of size b where it queues up packets received from the Internet. If the buffer is full, the router drops any received packet. The router processes and transmits packets from the buffer at some rate which depends on the configuration of the router. Let us assume that the time is discretized into tiny periods (say 1 μsec). At each time period, with probability p there is a new arrival. At each time period, with probability q a packet is processed (if there is one in the buffer) and transmitted by the router. Note that during a time period, we might have both an arrival and a transmission, or neither.

Given such a router in the Internet, the arrival rate solely depends on the traffic in the Internet and can be determined exactly under given traffic conditions. However, the rate (i.e. q) at which the packets leave the router depends on the configuration (e.g. say security configuration) of the router itself. Suppose, the manufacturer of the router specifies that q always lie in the range $[q_{\min}, q_{\max}]$, where q is equal to the lower bound $q_{\min}$ if all the security measures are active, and is equal to the upper bound $q_{\max}$ if none of the security measures are active. However, the exact value of q when certain number of security measures are active cannot be determined exactly.

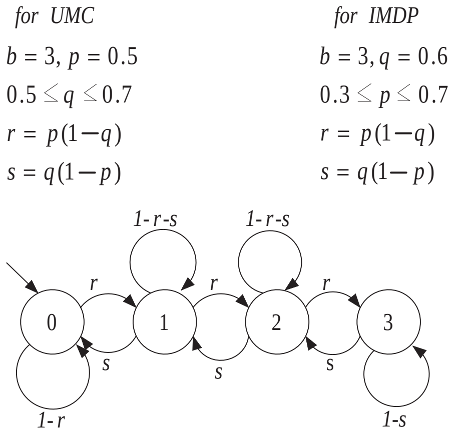

Fig. 3. Model of Router with Buffer Size 3

Suppose we want to model check a property such as “the probability that the buffer of a router eventually becomes full is less than 0.01.” For the router in the Internet, we exactly know the size of the buffer b and the arrival probability p. However, the departure probability is uncertain and is known to lie in the range $[q_{\min}, q_{\max}]$. A natural way to model such a system is using UMCs. A UMC model of the router with buffer size 3 is given in Figure 3. The arrival probability p is assumed to be known, say 0.5;

the departure probability q lies in the range $[0.5, 0.7]$ as provided by the manufacturer of the router. The label on a state gives the number of packets in the buffer of the router.

Yet another situation in which UMC models arise is in "black-box model-checking" [25]. In black-box model-checking, we assume that the transition probabilities of a DTMC model is not known; rather, we are allowed to estimate the probabilities using Monte-Carlo simulation. For example, through Monte-Carlo simulation, if we observe that out of total n transitions from a given state s to any other state there are only m transitions from s to s', then we can estimate the transition probability from s to s' by m/n. However, this estimation of the transition probability from s to s' by m/n is not statistically sound. Instead, we should consider a range of probabilities within which the actual probability must lie with high probability. Such a range is called *a confidence interval* [12]. For example, a 99% confidence interval for the transition probability would be the range $[p_1, p_2]$, if the probability that our observed transition probability is m/n given that the actual probability of transition lies in $[p_1, p_2]$ is 0.99. A confidence interval for a given confidence level and a given observation can be calculated by standard techniques. Thus for black-box models the estimated values for the various transition probabilities are better represented as UMCs.

IMDP in practice. There may be situations where the system cannot be modeled as an UMC. For example, in the router example above, the arrival rate of packets may vary from time to time depending on the Internet traffic. Therefore, we cannot assume that p is an exact probability; rather, it lies in a range. At every transition the environment chooses a p from the range non-deterministically and then decides to send a packet to the router with the chosen probability p.

Such situations can be naturally modeled as an IMDP, in which every time a state is visited, a probability distribution respecting certain range constraints is non-deterministically (possibly even adversarially) chosen, and then a transition is taken according to the chosen distribution. Thus, in IMDPs the non-deterministic choice is made over a set of (possibly) uncountably many choices. Note that this is different from MDPs (Markov Decision Processes) [7, 4] where the number of possible non-deterministic choices is finite.

For example, Figure 3 gives the IMDP model of a router (with buffer size 3) where, for simplicity, we assume that the departure probability q is fixed number, say 0.6. The arrival probability, however, lies in the range $[0.3, 0.7]$ depending on the Internet traffic.

B Bilinear Matrix Inequalities (BMI)

Recall that a $k \times k$ matrix A, over the reals, is said to be *positive semi-definite* if A is symmetric (i.e., $A = A^T$) and for every $z \in \mathbb{R}^k$, $z^T A z \geq 0$. We will denote A is positive semi-definite by $A \succeq 0$.

Optimization programs with bilinear matrix inequalities (BMIs) [22] are of the form

$$\text{maximize/minimize } C^T x$$

$$\text{subject to}$$

$$F(x, y) = F_0 + \sum_{i=0}^{m} x_i F_i + \sum_{j=0}^{n} y_j G_j + \sum_{i=0}^{m} \sum_{j=0}^{n} x_i y_j H_{ij} \succeq 0 \qquad (7)$$

where for every i and j, F_i, G_j and H_{ij} are symmetric matrices of the same dimension (say k), and $C, x \in \mathbb{R}^m, y \in \mathbb{R}^n$. Thus, the symmetric matrix $F(x, y)$ is an affine function of the elements of x and y and is required to be positive semidefinite.

Solving such optimization problems is known to be NP-hard [26], but is decidable. Efficient algorithms [10, 9] and tools [18] have been developed for solving optimization problems with BMI.

All the optimization problems that are solved during the model checking of UMC can be written as a single BMI. This follows from some simple observations. Our first observation says that a set of bilinear matrix inequality constraints can be rewritten as a single BMI of the form given in Equation (7).

Lemma 9. *A set of matrix inequalities*

$$F_0^k + \sum_{i=0}^{m} x_i F_i^k + \sum_{j=0}^{n} y_j G_j^k + \sum_{i=0}^{m}\sum_{j=0}^{n} x_i y_j H_{ij}^k \succeq 0$$

for $k = 1, \dots \ell$ can be written as a single BMI constraint.

Proof. The single BMI will be of the form

$$F_0 + \sum_{i=0}^{m} x_i F_i + \sum_{j=0}^{n} y_j G_j + \sum_{i=0}^{m}\sum_{j=0}^{n} x_i y_j H_{ij} \succeq 0$$

where F_0 is a block diagonal matrix with the matrices F_0^k along the diagonal; similarly F_i, G_j, and H_{ij} be will block diagonal matrices with F_i^k, G_j^k and H_{ij}^k along the diagonal [18]. □

The model checking problems that we investigate in this paper, will require us to optimize a simple linear function subject to certain constraints. The constraints that arise in the context of model checking will be of special forms. The next two lemmas show that these special constraints can be viewed as BMI constraints.

Lemma 10. *For $f_0, f_i, g_j, h_{ij} \in \mathbb{R}$, the (scalar) inequality*

$$f_0 + \sum_{i=0}^{m} x_i f_i + \sum_{j=0}^{n} y_j g_j + \sum_{i=0}^{m}\sum_{j=0}^{n} x_i y_j h_{ij} \geq 0 \tag{8}$$

can be written as a bilinear matrix inequality.

Proof. Let F_0, F_i, G_j, H_{ij} be matrices of dimension 1×1 whose entries are f_0, f_i, g_j, and h_{ij}, respectively. Consider

$$F(x, y) = F_0 + \sum_{i=0}^{m} x_i F_i + \sum_{j=0}^{n} y_j G_j + \sum_{i=0}^{m}\sum_{j=0}^{n} x_i y_j H_{ij}$$

It is easy to see that $F(x, y)$ is positive semi-definite if and only if the inequality (8) holds. □

Lemma 11. *The (strict) inequality $\delta > 0$ can be expressed as a BMI.*

Proof. Observe that $\delta > 0$ if and only if $x\delta \geq 1$, and $\delta \geq 0$. Thus the observation follows from Lemma 9 and Lemma 10. □

Our last observation is that the BMI requirement that the variables be partitioned into disjoint sets X and Y, such that the product terms only involve one variable from X with one from Y can be easily achieved by adding more variables and constraints.

Lemma 12. *A set of inequalities over the variables V of the form*

$$a_0^k + \sum_{i=1}^{n} a_i^k v_i + \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^k v_i v_j \geq 0$$

for $k = 1, \ldots \ell$ can written as a BMI.

Proof. For each variable $v_i \in V$ consider two variables: an "x-copy" v_i^x and a "y-copy" v_i^y. Replace a constraint of the form

$$a_0^k + \sum_{i=1}^{n} a_i^k v_i + \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^k v_i v_j \geq 0 \quad \text{with} \quad a_0^k + \sum_{i=1}^{n} a_i^k v_i^x + \sum_{i=1}^{n}\sum_{j=1}^{n} b_{ij}^k v_i^x v_j^y \geq 0$$

Also, add the constraints $v_i^x = v_i^y$ for each i. Observe that by Lemma 10, each constraint can be written as a BMI, where the variables in X are the x-copies of each variable, and those in Y are the y-copies of each variable. Thus, by Lemma 9, the resulting set of inequalities can be written as a BMI.