Statistical Model Checking of Incomplete Stochastic Systems

Shiraj Arora, Axel Legay, Tania Richmond, Louis-Marie Traonouez

Statistical Model Checking of Incomplete Stochastic Systems

Shiraj Arora[1], Axel Legay[2], Tania Richmond[2], and Louis-Marie Traonouez[2]

[1] Indian Institute of Technology Hyderabad, India
[2] Inria, Rennes, France

Abstract. We study incomplete stochastic systems that are missing some parts of their design, or are lacking information about some components. It is interesting to get early analysis results of the requirements of these systems, in order to adequately refine their design. In previous works, models for incomplete systems are analysed using model checking techniques for three-valued temporal logics. In this paper, we propose statistical model checking algorithms for these logics. We illustrate our approach on a case-study of a network system that is refined after the analysis of early designs.

1 Introduction

Stochastic systems comprise a wide range of applications that use probability distributions to describe the behaviour of a system. For instance, probabilities naturally arise during the execution of many cyber-physical systems to account for the variability of the physical processes connected to computer systems. They are also useful to model uncertainty in communication systems or protocols. The classical formalisms for modelling stochastic systems are Discrete and Continuous Time Markov Chains (DTMC and CTMC).

Formal specification requirements on stochastic systems can be formulated in temporal logics like the Linear Temporal Logic (LTL), the Computational Tree Logic (CTL) or a quantitative extension like the Probabilistic Computation Tree Logic (PCTL) [13]. Many works have studied the probabilistic model checking problem to formally verify these systems [23,13,9]. These works combine classical model checking algorithms with a numerical analysis to compute exact probabilities. These algorithms are however very expensive and often cannot scale to real-life problems.

To circumvent these limitations, one can resort to approximated techniques like Statistical Model Checking (SMC) [18,14,25,11]. These techniques rely on simulations and statistical analysis to evaluate the satisfaction of formal requirements. Requirements are usually expressed in the Bounded Linear Temporal Logic (BLTL). Statistical model checking can solve either quantitative estimation problems that evaluate probabilities, or qualitative analysis problems that perform hypothesis testing. It provides a trade-off between speed and accuracy that can be controlled by the number of simulations. These simulations can be

easily distributed on computing grids to increase the speed and accuracy of the analysis.

While probability distributions account for the variability and uncertainty in the system, like the variability of measurements, a precise specification of these distributions is required to analyse a system. This requirement is often hard to meet, especially in the early stages of the design of the system or in highly dynamic systems. Unknown specifications may also arise from the simulation process of the system as some components may be difficult to simulate. It is however of deep interest to have some early evaluation of the requirements of the system, even if it is not completely defined. In particular, a designer would want to know which of the following hypothesis holds: 1. the requirement is satisfied in the current design and in any of the subsequent designs that can be obtained by replacing unknown specifications, 2. the requirement may be satisfied in at least one subsequent design, and 3. the requirement will not be satisfied in any subsequent designs.

Related Works. Solutions for handling unknown specifications in stochastic systems usually imply switching from Boolean logics to three-valued or multi-valued logics. There is a rich theory on multi-valued logics and applications in multiple domains.

Our work is based on the works of Arora et al. [1,2] that introduces a three-valued PCTL logic. In this qPCTL logic *unknown* values are added to the atomic propositions of a DTMC and a qPCTL model checking algorithm is proposed.

Multi-valued extensions of temporal logics have also been proposed by Chechik et al. [5,6]. They introduce three-valued logics for atomic propositions and transitions and they perform model checking of multi-valued CTL.

In probabilistic model checking, multi-valued logics have been used for abstraction of Markov chains models to reduce the complexity of analysis [16,12,15]. Multiple states are combined in order to yield a reduced model. This process may lead to a loss of information that is represented with an *unknown* value. Abstracted models are then often analysed with probabilistic model checker for Markov Decision Processes (MDP).

Finally, Bauer et al. [3] introduced a four-valued semantics for LTL over finite traces. It is used in runtime verification to determine if a property is already satisfied, or if it may be satisfied in the future.

Our contribution. We extend the works in [1,2] to propose statistical model checking analysis of discrete time Markov chains with unknown values (qDTMC). We address both the quantitative estimation problem and the qualitative analysis problem. For the qualitative analysis problem, we adapt the model checking algorithm of [2] to perform a three hypotheses test and provide bounds on the probability of errors of this test. We finally propose an experiment in which we show how qDTMC and SMC can be used in a refinement process.

Organisation of the paper. Section 2 gives the basic definitions concerning DTMC and formal logics. Section 3 introduces unknown values in DTMC and extends

the semantics of BLTL to a three-valued logic. Section 4 proposes an estimation algorithm for BLTL with unknown values, while Section 5 studies the qualitative analysis problem and proposes a three hypotheses testing algorithm. Section 6 presents the implementation of these algorithms in the tool Plasma Lab, and Section 7 applies it to a network case-study. Finally, Section 8 concludes the paper.

2 Preliminaries

2.1 Discrete Time Markov Chains

Discrete Time Markov Chains (DTMC) are finite automata with a transition probability matrix.

Definition 1 (DTMC). *A DTMC is a tuple* $M = (S, \mathbb{P}, s_{init}, AP, L)$ *where:*

- S *is a set of states,*
- $\mathbb{P} : S \times S \to [0,1]$ *is a transition probability matrix, such that* $\forall s \in S : \sum_{s' \in S} \mathbb{P}(s, s') = 1$,
- $s_{init} \in S$ *is the initial state,*
- AP *is a set of atomic proposition,*
- $L : S \to AP$ *is a labelling function.*

Definition 2. *A* ***path*** π *in a DTMC* M *is a sequence of states* $s_0, s_1, s_2...$ *such that* $\forall i \in \{0,\ 1,\ 2, ...\}$, $\mathbb{P}(s_i, s_{i+1}) > 0$. *Let* π^i *denotes the suffix of* π *starting at state* s_i, *that is to say the path* $s_i, s_{i+1}, s_{i+2}....$

2.2 Bounded Linear Temporal Logic

The Linear Temporal Logic (LTL) allows expressing properties over the paths of a finite state system, such as a DTMC. It extends classical Boolean logic with *temporal operators* that allow reasoning on the temporal dimension of an execution path. These temporal operators express properties over the future of a path, with an unbounded number of states.

The Bounded Linear Temporal Logic (BLTL) is a restriction of LTL that specifies bounds on the temporal operators such that the properties can always be decided on finite executions. This characteristic allows verifying these properties using only simulation based approaches, such as statistical model checking.

Definition 3 (BLTL). *Bounded Linear Temporal Logic (BLTL) is used to express linear-time properties of the system. The syntax for BLTL is as follows:*

$$\Phi \ ::= \ T \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \Phi_1 \to \Phi_2 \mid X\Phi \mid F^{\leq k}\Phi \mid G^{\leq k}\Phi \mid \Phi_1 U^{\leq k}\ \Phi_2 \mid \Phi_1 W^{\leq k}\ \Phi_2$$

where Φ, Φ_1, *and* Φ_2 *are BLTL formulae,* a *is an atomic proposition, and* $k \in \mathbb{N}$ *is the time bound.*

Definition 4 (Semantics of BLTL). *Let $\mathcal{M} : (S, \mathbb{P}, s_{init}, AP, L)$ be a DTMC. Let $\pi = s_0\ s_1\ s_2 ...$ be a path in $\mathcal{M}$ and Φ, Φ_1, Φ_2 be BLTL formulae. Then, Φ is said to be satisfied in path π, i.e. $(\pi, \Phi) = \mathsf{T}$, if one of the following conditions is satisfied. Otherwise, the property is said to be not satisfied and denoted as $(\pi, \Phi) = \mathsf{F}$.*

1. $(\pi, T) = \mathsf{T}$,
2. $(\pi, a) = \mathsf{T}$ *iff* $a \in L(s_0)$,
3. $(\pi, \neg\Phi) = \mathsf{T}$ *iff* $(\pi, \Phi) = \mathsf{F}$,
4. $(\pi, \Phi_1 \wedge \Phi_2) = \mathsf{T}$ *iff* $(\pi, \Phi_1) = \mathsf{T} \wedge (\pi, \Phi_2) = \mathsf{T}$,
5. $(\pi, \Phi_1 \rightarrow \Phi_2) = \mathsf{T}$ *iff* $(\pi, \Phi_2) = \mathsf{T}$ *whenever* $(\pi, \Phi_1) = \mathsf{T}$,
6. $(\pi, X\Phi) = \mathsf{T}$ *iff* $(\pi^1, \Phi) = \mathsf{T}$,
7. $(\pi, F^{\leq k}\Phi) = \mathsf{T}$ *iff* $\exists i \leq k, (\pi^i, \Phi) = \mathsf{T}$,
8. $(\pi, G^{\leq k}\Phi) = \mathsf{T}$ *iff* $\forall i \leq k, (\pi^i, \Phi) = \mathsf{T}$,
9. $(\pi, (\Phi_1\ U^{\leq k}\ \Phi_2)) = \mathsf{T}$ *iff* $\exists i \leq k, (\pi^i, \Phi_2) = \mathsf{T} \wedge\ \forall j < i, (\pi^j, \Phi_1) = \mathsf{T}$,
10. $(\pi, (\Phi_1\ W^{\leq k}\ \Phi_2)) = \mathsf{T}$ *iff* $[\exists i \leq k, (\pi^i, \Phi_2) = \mathsf{T} \wedge\ \forall j < i, (\pi^j, \Phi_1) = \mathsf{T}]\ \vee\ [\forall j \leq k, (\pi^j, \Phi_1) = \mathsf{T}]$.

For a BLTL property Φ and a path π, we will write $\pi \models \Phi$ if $(\pi, \Phi) = \mathsf{T}$ and $\pi \not\models \Phi$ if $(\pi, \Phi) = \mathsf{F}$. However, these notations will not be applicable to the three-valued logic that we will present in the next section.

2.3 Statistical Model Checking

Statistical Model Checking (SMC) is an alternative to probabilistic model checking that evaluates the satisfaction of formal properties on any stochastic systems. It combines the formal analysis of linear temporal properties on finite simulations of the system with statistical methods. Contrary to an exhaustive exploration, statistical model checking does not store the state of the system and therefore can even be applied to infinite state systems. It returns approximated results, either quantitative results with confidence intervals or qualitative results with bounds on the probability of error.

Statistical model checking relies on generating a finite number of independent simulations either from a formal model, like DTMC, or even directly from a system simulator. These simulations are formally analysed using a monitor of a linear temporal property like BLTL. The results obtained for each simulation are combined by a statistical algorithm.

Quantitative estimation. Considering a random path π from a stochastic system, the satisfaction of a BLTL property Φ defines a random variable with a Bernoulli distribution. The goal of a quantitative analysis is to estimate the parameter of this distribution, that is to say the probability $\gamma = Prob(\Phi \models \pi)$. This estimation can be done using a Monte Carlo approach that consists in generating a set of n executions and computes an estimate of the probability using the following formula:

$$\overline{\gamma} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(\pi_i \models \Phi)$$

where $\mathbf{1}$ is an indicator function that returns 1 if $\pi_i \models \varphi$ and 0 otherwise.

The number of simulations allows to control the precision of the analysis. For instance the Chernoff bound [8] can be used to relate the number of simulations to the absolute error ϵ and the confidence δ with the following formula:

$$Prob(|\overline{\gamma} - \gamma| \geq \epsilon) \leq \delta \quad \text{if} \quad \delta = 2e^{-2n\epsilon^2}$$

Qualitative analysis. To test if the probability γ is greater (or lower) than a given bound θ, then the problem can be formulated as a hypothesis test with two hypotheses: $H_0 : \gamma \geq \theta$ against $H_1 : \gamma < \theta$. Statistical techniques can then be used to estimate the true hypothesis from the results of the simulations. These techniques do not guarantee a correct result but usually come with two bounds on the probability of making errors. These two bounds are α and β, such that the probability of accepting H_1 (resp. H_0) when H_0 (resp. H_1) holds, called a Type-I error (resp. a Type-II error) is less or equal to α (resp. β).

Statistical tests cannot guarantee a low probability for both types of error (see [25] for details). A solution is to relax the problem with an indifference region that can be defined with a parameter δ, such that the two hypotheses become $H_0 : \gamma \geq \theta + \delta = p_0$ and $H_1 : \gamma \leq \theta - \delta = p_1$. If the true probability γ is between $[p_1, p_0]$ then we are indifferent to which hypothesis is accepted.

A common test to analyse the qualitative problem is the Sequential Probability Ratio Test (SPRT) [24]. This test is based on a variable number of simulations that allows giving results in an online manner, as soon as enough simulations guarantee a decision. The SPRT test computes a ratio based on the last m simulations generated and decide after each new simulation whether a decision can be made or more simulations are needed. The ratio is given by the following formula:

$$ratio = \prod_{i=1}^{m} \frac{p_1^{\mathbf{1}(\pi_i \models \mathbf{\Phi})}(1-p_1)^{\mathbf{1}(\pi_i \not\models \mathbf{\Phi})}}{p_0^{\mathbf{1}(\pi_i \models \mathbf{\Phi})}(1-p_0)^{\mathbf{1}(\pi_i \not\models \mathbf{\Phi})}}$$

After each new simulation is generated, the ratio is updated. H_1 is accepted if $ratio \geq (1-\beta)/\alpha$ whereas H_0 is accepted if $ratio \geq \beta/(1-\alpha)$. Otherwise, a new simulation is generated.

3 Adding Uncertainty in DTMC and BLTL

To add uncertainty in DTMC models, authors in [1] modified the labelling function of the DTMC such that it may return three values (true, false or unknown) instead of two. This labelling function is then considered as a parameter of the DTMC.

Definition 5 (qDTMC [1]). *A qDTMC $\mathcal{M}(L)$ is a DTMC $\mathcal{M}$ parametrized with a labelling function L allowing unknown values. $\mathcal{M}(L) = (S, \mathbb{P}, s_{init}, AP, L)$ where:*

- *S is a set of states,*

– $\mathbb{P} : S \times S \to [0,1]$ *is a transition probability matrix, such that* $\forall s \in S : \sum_{s' \in S} \mathbb{P}(s,s') = 1$,
– $s_{init} \in S$ *is the initial state,*
– AP *is a set of atomic propositions,*
– $L : S \times AP \to \{\mathsf{T}, \mathsf{F}, ?\}$ *is a labelling function.*

We define a refinement relation (partial order) between the labelling functions with unknown values, such that a function L_2 refines L_1 if L_2 only replaces some unknown values from L_1 with T or F (and it does not change the values T and F form L_1). Formally,

Definition 6 (Refinement). *Given two labelling functions* L_1, L_2 *of a qDTMC* $\mathcal{M} = (S, \mathbb{P}, s_{init}, AP)$*, we say that* $L_2 \prec L_1$ *iff* $\forall s \in S, \forall p \in AP, (L_1(s,p) = \mathsf{T}) \Rightarrow (L_2(s,p) = \mathsf{T})$ *and* $(L_1(s,p) = \mathsf{F}) \Rightarrow (L_2(s,p) = \mathsf{F})$.

We now extend the semantics of BLTL properties to take care of the unknown information in the path of the qDTMC. The new logic qBLTL has the same syntax as BLTL. Its semantics however may return three values (T, F or ?) instead of a Boolean value. Also, the semantics for a path π of a qDTMC $\mathcal{M}$ now depend on its labelling function L.

Definition 7 (Semantics of qBLTL [1]). *Let* $\mathcal{M}(\mathcal{L}) : (S, \mathbb{P}, s_{init}, AP, L)$ *be a qDTMC. Let* $\pi = s_0\ s_1\ s_2 \ldots$ *be a path in* $\mathcal{M}(\mathcal{L})$ *and* Φ, Φ_1, Φ_2 *be qBLTL formulae. The semantics of qBLTL formulae are as follow:*

1. $(\pi, L, T) = \mathsf{T}$
2. $(\pi, L, a) = L(s_0, a)$
3. $(\pi, L, \neg\Phi) = \begin{cases} \mathsf{T} \text{ iff } (\pi, L, \Phi) = \mathsf{F} \\ \mathsf{F} \text{ iff } (\pi, L, \Phi) = \mathsf{T} \\ ? \text{ iff } (\pi, L, \Phi) = ? \end{cases}$
4. $(\pi, L, \Phi_1 \wedge \Phi_2) = \begin{cases} \mathsf{T} \text{ iff } (\pi, L, \Phi_1) = \mathsf{T} \ \wedge \ (\pi, L, \Phi_2) = \mathsf{T} \\ \mathsf{F} \text{ iff } (\pi, L, \Phi_1) = \mathsf{F} \ \vee \ (\pi, L, \Phi_2) = \mathsf{F} \\ ? \text{ otherwise} \end{cases}$
5. $(\pi, L, \Phi_1 \to \Phi_2) = \begin{cases} \mathsf{T} \text{ iff } (\pi, L, \Phi_1) = \mathsf{F} \ \vee \ (\pi, L, \Phi_2) = \mathsf{T} \\ \mathsf{F} \text{ iff } (\pi, L, \Phi_1) = \mathsf{T} \ \wedge \ (\pi, L, \Phi_2) = \mathsf{F} \\ ? \text{ otherwise} \end{cases}$
6. $(\pi, L, X\Phi) = (\pi^1, L, \Phi)$
7. $(\pi, L, F^{\leq k}\Phi) = \begin{cases} \mathsf{T} \text{ iff } \exists i \leq k, (\pi^i, L, \Phi) = \mathsf{T} \\ \mathsf{F} \text{ iff } \forall i \leq k, (\pi^i, L, \Phi) = \mathsf{F} \\ ? \text{ otherwise} \end{cases}$
8. $(\pi, L, G^{\leq k}\Phi) = \begin{cases} \mathsf{T} \text{ iff } \forall i \leq k, (\pi^i, L, \Phi) = \mathsf{T} \\ \mathsf{F} \text{ iff } \exists i \leq k, (\pi^i, L, \Phi) = \mathsf{F} \\ ? \text{ otherwise} \end{cases}$
9. $(\pi, L, (\Phi_1\ U^{\leq k}\ \Phi_2)) =$
$$\begin{cases} \mathsf{T} \text{ iff } \exists i \leq k, (\pi^i, L, \Phi_2) = \mathsf{T} \wedge \ \forall j < i, (\pi^j, L, \Phi_1) = \mathsf{T} \\ \mathsf{F} \text{ iff } [\forall i \leq k, (\pi^i, L, \Phi_2) = \mathsf{F}] \\ \qquad \vee [\exists i \leq k, (\pi^i, L, \Phi_2) = \mathsf{T} \wedge \exists j < i, (\pi^j, L, \Phi_1) = \mathsf{F}] \\ ? \text{ otherwise} \end{cases}$$

10. $(\pi, L, (\Phi_1\ W^{\leq k}\ \Phi_2)) =$

$$\begin{cases} \mathsf{T} \text{ \textit{iff} } [\exists i \leq k, (\pi^i, L, \Phi_2) = \mathsf{T}\ \wedge\ \forall j < i, (\pi^j, L, \Phi_1) = \mathsf{T}] \\ \qquad\quad \vee\ [\forall j \leq k, (\pi^j, L, \Phi_1) = \mathsf{T}] \\ \mathsf{F} \text{ \textit{iff} } \exists i \leq k, (\pi^i, L, \Phi_1) = \mathsf{F}\ \wedge\ \forall j < i, (\pi^j, L, \Phi_2) = \mathsf{F} \\ ?\ \textit{otherwise} \end{cases}$$

The following proposition shows that every qBLTL formula satisfied (resp. unsatisfied) by a qDTMC with a labelling function L_1 is also satisfied (resp. unsatisfied) with any refinement $L_2 \prec L_1$.

Proposition 1. *Given two labelling functions L_1, L_2 of a qDTMC $\mathcal{M}$ such that $L_2 \prec L_1$, and a qPCTL formula Φ. Let π be a path from $\mathcal{M}$. Then,*

$$(\pi, L_1, \Phi) = \mathsf{T} \Rightarrow (\pi, L_2, \Phi) = \mathsf{T}$$

$$(\pi, L_1, \Phi) = \mathsf{F} \Rightarrow (\pi, L_2, \Phi) = \mathsf{F}$$

Proof. This proposition can be easily proved using an induction on the shape of qBLTL formulae. We assume that the proposition holds for sub-formulae Φ, Φ_1 and Φ_2 and we prove that it also holds for any of the formulae that can be built using equations 1. to 10. in Definition 7. We prove below the two base cases, 1 and 2, and the induction for equation 9. Other equations are similar.

1. It trivially holds for basic formula 1.
2. For formula 2, if $(\pi, L_1, a) = L_1(s_0, a) = \mathsf{T}$ then by definition of $\prec$ $(\pi, L_2, a) = L_2(s_0, a) = \mathsf{T}$. Conversely, if $(\pi, L_1, a) = L_1(s_0, a) = \mathsf{F}$ also by definition of $\prec$ $(\pi, L_2, a) = L_2(s_0, a) = \mathsf{F}$.
9. For formula 9, if $(\pi, L_1, (\Phi_1\ U^{\leq k}\ \Phi_2)) = \mathsf{T}$, then there exists $i < k$ with $(\pi^i, L_1, \Phi_2) = \mathsf{T}$ and for all $j < i$, $(\pi^j, L_1, \Phi_1) = \mathsf{T}$. By induction hypothesis, $(\pi^i, L_2, \Phi_2) = \mathsf{T}$ and $(\pi^j, L_2, \Phi_1) = \mathsf{T}$. This proves that $(\pi, L_2, (\Phi_1\ U^{\leq k}\ \Phi_2)) = \mathsf{T}$. Conversely, if $(\pi, L_1, (\Phi_1\ U^{\leq k}\ \Phi_2)) = \mathsf{F}$, then either $\forall i \leq k, (\pi^i, L_1, \Phi_2) = \mathsf{F}$. By induction hypothesis, $(\pi^i, L_2, \Phi_2) = \mathsf{F}$ and thus $(\pi, L_2, (\Phi_1\ U^{\leq k}\ \Phi_2)) = \mathsf{F}$. Or there exists $i \leq k$, $(\pi^i, L_1, \Phi_2) = \mathsf{T}$ and $j < i$, $(\pi^j, L_1, \Phi_1) = \mathsf{F}$. Again, by induction hypothesis, $(\pi^i, L_2, \Phi_2) = \mathsf{T}$ and $(\pi^j, L_2, \Phi_1) = \mathsf{F}$, which proves that $(\pi, L_2, (\Phi_1\ U^{\leq k}\ \Phi_2)) = \mathsf{F}$. □

4 Quantitative Estimation of BLTL with Unknown

In [14], Hérault et al. provided a quantitative estimation of the probability that a BLTL property is satisfied with high confidence by a stochastic system (c.f. Subsection 2.3). In this section, we propose an efficient extension to qBLTL.

4.1 Three-valued estimation algorithm

In order to estimate the probability that a qDTMC $\mathcal{M}(L)$ satisfies a property Φ in the qBLTL logic, we generate random paths π in the probabilistic space

underlying the qDTMC structure of depth k. Let n be the total number of simulations, t the number of T, f the number of F and u the number of ?. Notice that $n = t+f+u$. Then in order to estimate the probability $\gamma = \text{Prob}((\pi, L, \Phi) = \mathsf{T})$, we test if the property Φ holds on each path π_i (for $i = 1, \ldots, n$), and compute a random value t/n, otherwise from the number of paths that do not hold Φ, we compute a random value f/n. Finally we deduce the probability for unknown values from the others, i.e. $u/n = 1-(t+f)/n$. This is described in Algorithm 1, called three-valued estimation algorithm for True-False-Unknown.

Algorithm 1: Three-valued estimation algorithm

Data: $\mathcal{M}(L)$ qDTMC model, Φ the property to verify, n the number of simulations
Result: P_T probability of true values, P_F probability of false values and P_U probability of unknown values
begin
  $nb_T = 0$; $nb_F = 0$;
  for $i = 1$ *to* n *do* **do**
    Generate a random path π of length k from $\mathcal{M}(L)$;
    if Φ *is true on* π **then**
      $nb_T = nb_T + 1$;
    else if Φ *is false on* π **then**
      $nb_F = nb_F + 1$;
    end
  end
  $P_T = nb_T/n$; $P_F = nb_F/n$; $P_U = 1 - (P_T + P_F)$;
  return P_T, P_F, P_U
end

Remark 1. Algorithm 1 can also be used to verify qualitative statements of the form $\text{Prob}((\Phi \models \pi) \geqslant \theta$. We test whether $t/n > \theta-\varepsilon$. Our decision is correct with confidence $(1 - \delta)$ after a number of samples polynomial in $1/\varepsilon$ and $\log(1/\delta)$. In Section 5 we will also present a sequential algorithm for these qualitative statements.

4.2 Correctness

Proposition 2. *The three-valued estimation algorithm is a randomized approximation for the probability* $\gamma = \text{Prob}((\Phi, L, \pi) = \mathsf{T})$ *for a qBLTL formula* Φ*, with* $\gamma \in [0, 1]$.

Proof. Let $X_1, X_2, \ldots, X_k$ be k independent random variables with a multinomial distribution. So each X_i takes value true (let say 1) with probability p_1, false (let say 0) with probability p_2, unknown (let say -1) with probability p_3 in our case. Let X be the random vector $[X_1, X_2, \ldots, X_k]^T$ and μ its mean.

Then $\mu = [\mu_1, \mu_2, \ldots, \mu_k]$, with μ_i the mean of X_i, for $i = 1, \ldots, k$. In our case, $k = 3$. Let z_i, for $i = 1, \ldots, k$ be positive integers such that $\sum_{i=1}^{k} z_i = n$. Let z be the vector $[z_1, z_2, \ldots, z_k]^T$. Then the multinomial generalization of the Chernoff-Hoeffding bound [7] gives:

$$Prob(X \geqslant z) \leqslant \prod_{i=1}^{k} \left(\frac{\mu_i}{z_i}\right)^{z_i} \quad \text{provided that } z \geqslant \mu.$$

In our case, $z_1 = nb_T$, $z_2 = nb_F$ and $z_3 = nb_U$, so $n = nb_T + nb_F + nb_U$. Using Stirling's approximation, we can easily check that the minimum bound for n is $4\log(\frac{2}{\delta})/\varepsilon^2$ to get the estimation:

$$Prob(|X - \gamma| \leqslant \varepsilon) \geqslant 1 - \delta.$$

5 Qualitative Analysis of BLTL with Unknown

5.1 Problem

Let Φ be a qBLTL property and $M(L)$ be a qDTMC with labelling function L. Let π be a random path from M. The qualitative analysis problem consists in determining whether one of the two following hypotheses holds: $H_0 : \text{Prob}((\pi, L, \Phi) = \mathsf{T}) \geq \theta$ and $H_1 : \text{Prob}((\pi, L, \Phi) = \mathsf{F}) > 1 - \theta$. This problem defines a probabilistic qBLTL property $\text{Prob}_{\geq\theta}((\pi, L, \Phi) = \mathsf{T})$. It is a subclass of the qPCTL logic presented in [1].

If H_0 holds then Φ is satisfied with at least probability θ using the labelling function L. Consequently using Proposition 1, Φ is also satisfied with at least probability θ for any refinement of L. Conversely, if H_1 holds, Φ can be disproved with at least probability $1 - \theta$ using the labelling function L or any refinement of L.

Contrary to the two-valued qualitative analysis problem presented in Subsection 2.3, we cannot use a two-valued statistical test to distinguish these two hypotheses, since our simulations results are three-valued. We will therefore combine two statistical tests to design an hypothesis testing algorithm with three potential outcomes.

Remark 2. We present the case for the probabilistic qBLTL property $\text{Prob}_{\geq\theta}(\Phi = \mathsf{T})$. To check properties $\text{Prob}_{\leq\theta}(\Phi = \mathsf{T})$ we would consider the reverse problem that is checking $\text{Prob}_{\geq 1-\theta}(\neg\Phi = \mathsf{F})$ and then use the same algorithm.

5.2 Hypothesis Testing Algorithm

To solve the qualitative analysis problem of a qBLTL property with SMC we propose an hypothesis testing algorithm that may return three values:

- T if the hypothesis $H_0 : \text{Prob}((\pi, L, \Phi) = \mathsf{T}) \geq \theta$ is accepted,

– F if the hypothesis $H_1 : \text{Prob}((\pi, L, \Phi) = \mathsf{F}) > 1 - \theta$ is accepted,
– ? otherwise (we call this hypothesis H_2).

Algorithm 2 is an adaptation of the model checking algorithm qMC presented in [1]. It follows the same principles. It involves three subroutines. The two subroutines SetToFalse and SetToTrue consist in modifying the original labelling function of the qDTMC in order to replace all uncertainty by F or T, respectively. The results of these transformations are normal DTMC that can be checked with classical SMC algorithms. That is what the subroutine BSMC performs, that is to say it uses the SMC hypothesis test that determines whether a DTMC satisfies a BLTL formula, using the SPRT algorithm presented in Subsection. 2.3, This subroutine returns either T or F.

Algorithm 2: qSMC

Input:
$\mathcal{M}(L)$: qDTMC with labelling function L
φ: probabilistic BLTL formula

$L_F \leftarrow \text{SetToFalse}(L)$
if $\text{BSMC}(\mathcal{M}(L_F), \varphi) = \mathsf{T}$ **then**
  return T
else
  $L_T \leftarrow \text{SetToTrue}(L)$
  if $\text{BSMC}(\mathcal{M}(L_T), \varphi) = \mathsf{F}$ **then**
    return F
  else
    return ?

5.3 Error Bounds for Three Hypotheses Testing

The qSMC algorithm can be seen as a three hypotheses testing algorithm. Therefore the classical error bounds that are used to evaluate the precision of the results in a two hypotheses case, such as the error bounds for the SPRT algorithm, must be extended.

Instead of two types of errors we define three types of errors, one for each hypothesis being wrongly accepted. We say that the qSMC algorithm returns a wrong value when the result is different from the qMC algorithm. In the following we simply write $\text{qSMC} = *$ (resp. $\text{qMC} = *$) if the qSMC (resp. qMC) algorithm returns the value $* \in \{\mathsf{T}, \mathsf{F}, ?\}$. The three types of errors are:

1. **Type-I error**: H_1 is wrongly accepted: $\text{qSMC} = \mathsf{F}$ while $\text{qMC} \neq \mathsf{F}$
2. **Type-II error**: H_0 is wrongly accepted: $\text{qSMC} = \mathsf{T}$ while $\text{qMC} \neq \mathsf{T}$
3. **Type-III error**: H_2 is wrongly accepted: qSMC =? while $\text{qMC} \neq ?$

We will bound the probability of these errors using the parameters of the binary SMC tests used in the qSMC algorithm. We will write BSMC_T (resp. BSMC_F) to denote the results of the binary SMC algorithm on the qDTMC $\mathcal{M}(L_T)$ (resp. $\mathcal{M}(L_F)$), and similarly BMC_T and BMC_F to denote the true results of these tests. The bounds on the probability of errors of these tests are:

$$\text{Prob}(\text{BSMC}_F = \mathsf{F} \mid \text{BMC}_F = \mathsf{T}) \leq \alpha_1 \tag{1}$$
$$\text{Prob}(\text{BSMC}_F = \mathsf{T} \mid \text{BMC}_F = \mathsf{F}) \leq \beta_1 \tag{2}$$
$$\text{Prob}(\text{BSMC}_T = \mathsf{F} \mid \text{BMC}_T = \mathsf{T}) \leq \alpha_2 \tag{3}$$
$$\text{Prob}(\text{BSMC}_T = \mathsf{T} \mid \text{BMC}_T = \mathsf{F}) \leq \beta_2 \tag{4}$$

where α_1 and β_1 are the error bounds for the BSMC_F test and α_2 and β_2 the error bounds for the BSMC_T test.

Proposition 3 (Error bounds). *Given a qDTMC $\mathcal{M}(L)$ and a probabilistic BLTL formula φ, the probabilities that the* qSMC *algorithm returns a wrong answer compared to the* qMC *algorithm [2] is bounded:*

1. **Type-I error***:* $\text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} \neq \mathsf{F}) \leq \max(\alpha_1, \alpha_2)$.
2. **Type-II error***:* $\text{Prob}(\text{qSMC} = \mathsf{T} \mid \text{qMC} \neq \mathsf{T}) \leq \beta_1$.
3. **Type-III error***:* $\text{Prob}(\text{qSMC} = ? \mid \text{qMC} \neq ?) \leq \max(\alpha_1, \beta_2)$.

Proof. To compute bounds on the probability of error, we use the following properties on the probabilities between three events A, B and C:

$$\text{Prob}(A \mid B \vee C) \leq \max(\text{Prob}(A \mid B), \text{Prob}(A \mid C)) \quad \text{if } B \text{ and } C \text{ are disjoint events.} \tag{5}$$
$$\text{Prob}(A \wedge B) \leq \min(\text{Prob}(A), \text{Prob}(B)). \tag{6}$$
$$\text{Prob}(A \vee B) = \text{Prob}(A) + \text{Prob}(B) - \text{Prob}(A \wedge B) \leq \text{Prob}(A) + \text{Prob}(B). \tag{7}$$

Bound on Type-I error
$\text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} \neq \mathsf{F}) = \text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} = \mathsf{T} \vee \text{qMC} = ?)$
Then according to Equation 5:
$\leq \max(\text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} = \mathsf{T}), \text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} = ?))$
We bound these two probabilities:

$\text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} = \mathsf{T}) = \text{Prob}(\text{BSMC}_F = \mathsf{F} \wedge \text{BSMC}_T = \mathsf{F} \mid \text{BMC}_F = \mathsf{T})$
According to Equation6 :
$\leq \min(\text{Prob}(\text{BSMC}_F = \mathsf{F} \mid \text{BMC}_F = \mathsf{T}) \ , \ \text{Prob}(\text{BSMC}_T = \mathsf{F} \mid \text{BMC}_F = \mathsf{T}))$
and according to Equation1 : $\text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} = \mathsf{T}) \leq \alpha_1$

$\text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} =?)$
$= \text{Prob}(\text{BSMC}_F = \mathsf{F} \wedge \text{BSMC}_T = \mathsf{F} \mid \text{BMC}_F = \mathsf{F} \wedge \text{BMC}_T = \mathsf{T})$
According to Equation6 :
$\leq \min(\text{Prob}(\text{BSMC}_F = \mathsf{F} \mid \text{BMC}_F = \mathsf{F} \wedge \text{BMC}_T = \mathsf{T}),$
$\quad \text{Prob}(\text{BSMC}_T = \mathsf{F} \mid \text{BMC}_F = \mathsf{F} \wedge \text{BMC}_T = \mathsf{T}))$
and according to Equation3 : $\text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} =?) \leq \alpha_2$

Thus we get $\boxed{\text{Prob}(\text{qSMC} = \mathsf{F} \mid \text{qMC} \neq \mathsf{F}) \leq \max(\alpha_1, \alpha_2)}$.

Bound on Type-II error
$\text{Prob}(\text{qSMC} = \mathsf{T} \mid \text{qMC} \neq \mathsf{T}) = \text{Prob}(\text{qSMC} = \mathsf{T} \mid \text{qMC} = \mathsf{F} \vee \text{qMC} =?)$
$= \text{Prob}(\text{BSMC}_F = \mathsf{T} \mid \text{BMC}_F = \mathsf{F} \wedge (\text{BMC}_T = \mathsf{F} \vee \text{BMC}_T = \mathsf{T}))$
$= \text{Prob}(\text{BSMC}_F = \mathsf{T} \mid \text{BMC}_F = \mathsf{F})$ since $\text{BMC}_T = \mathsf{F}$ and $\text{BMC}_T = \mathsf{T}$ form a total partition of the probability space.
Finally according to Equation 2 we get $\boxed{\text{Prob}(\text{qSMC} = \mathsf{T} \mid \text{qMC} \neq \mathsf{T}) \leq \beta_1}$.

Bound on Type-III error
$\text{Prob}(\text{qSMC} =? \mid \text{qMC} \neq?) = \text{Prob}(\text{qSMC} =? \mid \text{qMC} = \mathsf{T} \vee \text{qMC} = \mathsf{F})$
$= \text{Prob}(\text{qSMC} =? \mid \text{BMC}_F = \mathsf{T} \vee (\text{BMC}_F = \mathsf{F} \wedge \text{BMC}_T = \mathsf{F}))$
$= \text{Prob}(\text{qSMC} =? \mid \text{BMC}_F = \mathsf{T} \vee \text{BMC}_T = \mathsf{F})$ since $\text{BMC}_F = \mathsf{T}$ and $\text{BMC}_F = \mathsf{F}$ form a total partition of the probability space. Then according to Equation 5:
$\leq \max(\text{Prob}(\text{qSMC} =? \mid \text{BMC}_F = \mathsf{T}), \text{Prob}(\text{qSMC} =? \mid \text{BMC}_T = \mathsf{F}))$
We bound these two probabilities:

$\text{Prob}(\text{qSMC} =? \mid \text{BMC}_F = \mathsf{T}) = \text{Prob}(\text{BSMC}_F = \mathsf{F} \wedge \text{BSMC}_T = \mathsf{T} \mid \text{BMC}_F = \mathsf{T})$
according to Equation6 and 1 :
$\text{Prob}(\text{qSMC} =? \mid \text{BMC}_F = \mathsf{T}) \leq \text{Prob}(\text{BSMC}_F = \mathsf{F} \mid \text{BMC}_F = \mathsf{T}) \leq \alpha_1$

$\text{Prob}(\text{qSMC} =? \mid \text{BMC}_T = \mathsf{F}) = \text{Prob}(\text{BSMC}_F = \mathsf{F} \wedge \text{BSMC}_T = \mathsf{T} \mid \text{BMC}_T = \mathsf{F})$
according to Equation6 and 4 :
$\text{Prob}(\text{qSMC} =? \mid \text{BMC}_T = \mathsf{F}) \leq \text{Prob}(\text{BSMC}_T = \mathsf{T} \mid \text{BMC}_T = \mathsf{F}) \leq \beta_2$

Thus we get $\boxed{\text{Prob}(\text{qSMC} =? \mid \text{qMC} \neq?) \leq \max(\alpha_1, \beta_2)}$. □

6 Implementation using Plasma Lab

We have implemented our three-valued estimation algorithm and the qSMC hypothesis testing algorithm in the tool Plasma Lab [4,19]. Plasma Lab is a generic platform for performing statistical model checking of stochastic systems. It provides several SMC algorithms that can be applied to different types of systems and properties using a plugin system. Plasma Lab's algorithms include

estimation algorithm with Monte Carlo method, hypothesis testing with SPRT, estimation of rare events with importance splitting and importance sampling [20], and algorithms for Markov decision processes [10]. Plasma Lab includes a simulator for the Reactive Module Language (RML) of the probabilistic model-checker Prism [17] that allows to specify discrete and continuous time Markov chains, as well as Markov decision processes. It also includes interfaces to external simulators such as SystemC [22], LLVM bytecode, or MATLAB/Simulink models [21].

To specify qDTMC in Plasma Lab, we use RML and the simulator plugin implemented in Plasma Lab. RML allows to write DTMC in a compact textual format by writing a set of concurrent modules, each having a set of local variables. These local variables define both the state of the system and its atomic propositions. The system switches from one state to another according to guarded transitions, either involving a single module or a synchronization between several modules. To write qDTMC in RML we add variables that will only be used as atomic propositions. Their values can be 1, for true, 0, for false, or -1, for unknown. With this representation, we do not need to implement a new language and a new simulator.

Plasma Lab also provides several formal logics for analysing paths of stochastic systems. To implement the qBLTL logic we have adapted the BLTL checker of Plasma Lab. The new plugin uses the same parser. It assumes that formulae only use three-valued atomic propositions defined in qDTMC. It implements a new semantics with the rules presented in Definition 7. The analysis of a path by this checker returns a value in $\{-1, 0, 1\}$.

We have implemented the three-valued estimation algorithm and the qSMC algorithm in two new algorithm plugins. The qSMC algorithm implements the subroutines SetToFalse and SetToTrue as syntactic transformations to remove unknowns from the model. It uses the SPRT algorithm and the BLTL checker of Plasma Lab to analyse the RML model after transformation. We summarize the components used in this implementation in Fig. 1.

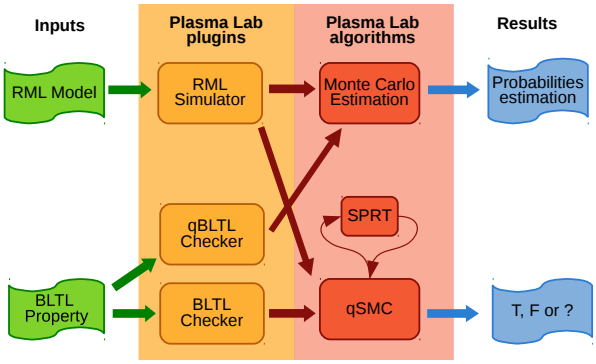
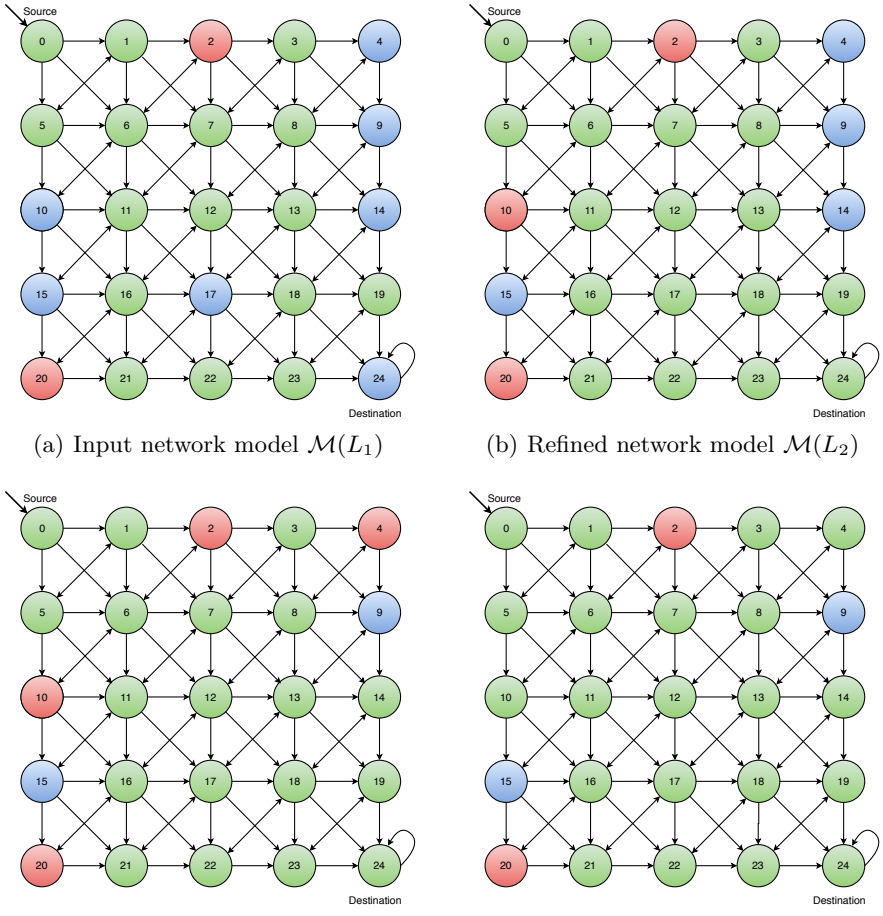


Fig. 1. Plasma Lab implementation of SMC algorithms for qDTMC

7 Experiments

In this section, we illustrate an example scenario wherein a set of nodes are connected to each other in a fixed network topology. A node, called source, wants to deliver a message to another node, called destination, via its neighbouring nodes. The message transfer occurs at discrete time steps. Each node forwards the message to one of its neighbouring nodes with some probability. However, it is possible that the message gets changed or corrupted at one or more intermediary nodes. The information about the nodes that can corrupt the message is incomplete. Thus, the aim is to analyze the probability of delivering an uncorrupted message from the source to the destination.



(a) Input network model $\mathcal{M}(L_1)$

(b) Refined network model $\mathcal{M}(L_2)$

(c) Refined network model $\mathcal{M}(L_3)$

(d) Refined network model $\mathcal{M}(L_4)$

Fig. 2. Example network models containing different number of unknowns

We model this network using a qDTMC $\mathcal{M}(L) = (S, \mathbb{P}, s_{init}, AP, L)$. The set of nodes in the network are represented as states in $\mathcal{M}(L)$ and the probability of sending messages between two nodes is governed by the transition probability matrix $\mathbb{P}$. We use an atomic proposition *notCorrupt* for each state $s \in S$. If the node corresponding to state s does not change the message, the labelling function L assigns true (T) to *notCorrupt* at s. Similarly, if the node changes the message, then *notCorrupt* is labelled as false (F) in s. However, if the behaviour of the node is not known, then the atomic proposition *notCorrupt* is labelled as unknown (?) in the state s. The qDTMC $\mathcal{M}(L)$ can be analyzed to estimate the probability that the message is delivered to the destination within 100 time units and it remains uncorrupted until it is delivered. We formally express this using qBLTL property Φ : $notCorrupt\ U^{\leq 100} delivered$. The qBLTL property Φ is analyzed using both qualitative and quantitative methods. If the analysis of Φ returns a large ratio of unknown results then the model $\mathcal{M}(L)$ is refined to $\mathcal{M}(L_1)$ with $L_1 \prec L$. The refinement from L to L_1 ensures that the ratio of unknowns in the model is reduced. The results of the experiments are discussed in the next subsection.

7.1 Results

The input topology of the network is shown in Figure 2(a). There are 25 nodes in the network, and each node has a uniform probability of forwarding the message to its allowed neighbours. The atomic proposition *notCorrupt* is true for the nodes coloured green, false for the nodes coloured red, and unknown for the nodes coloured blue. The node 0 is the source of the message and node 24 is the destination. This network topology is thus represented by a qDTMC $\mathcal{M}(L_1)$. The qualitative analysis of $\mathcal{M}(L_1)$ is done using the hypothesis testing Algorithm 2 for probabilistic BLTL query: $\text{Prob}_{\geq\theta}[notCorrupt\ U^{\leq 100} delivered]$. For different values of θ, the result generated is shown in Table 1. Similarly, by using the estimation algorithm for quantitative analysis, the probabilities for the BLTL query Φ : $[notCorrupt\ U^{\leq 100} delivered]$ to be true, false or unknown, is shown in Table 2.

It is evident from the results that for a large ratio of paths in $\mathcal{M}(L_1)$, the BLTL query is evaluated to unknown. For instance, Table 1 illustrates that the qualitative analysis of $\mathcal{M}(L_1)$ with $\theta = 0.5$ generates an unknown result. In particular, the probability of the property being unknown is estimated to 0.634, as shown in Table 2. Thus, we propose a new refined qDTMC $\mathcal{M}(L_2)$ with $L_2 \prec L_1$ wherein few nodes with atomic proposition *notCorrupt* as ? and the highest connectivity to neighbouring nodes are refined to a known truth value (either T or F). The network for the refined qDTMC is shown in Figure 2(b). $\mathcal{M}(L_2)$ is then again analyzed using both hypothesis testing and estimation method for Φ.

Indeed by refining $\mathcal{M}(L_1)$ to $\mathcal{M}(L_2)$, we can see in Table 2 that the probability of unknown result is reduced to 0.273. The qualitative analysis also results in a known result (T or F) for a number of different θ values. To further reduce the uncertainty in the model, we can again refine $\mathcal{M}(L_2)$ to $\mathcal{M}(L_3)$ such that

θ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$\mathcal{M}(L_1)$	T	?	?	?	?	?	F	F	F
$\mathcal{M}(L_2)$	T	T	?	?	?	F	F	F	F
$\mathcal{M}(L_3)$	T	T	T	?	F	F	F	F	F
$\mathcal{M}(L_4)$	T	T	T	T	T	?	F	F	F

Table 1. Hypothesis Testing results for various values of θ for probabilistic qBLTL property $Pr(\Phi) \geq \theta$

$\mathcal{M}(L_i)$	$Pr((\mathcal{M}(L_i), \Phi) = T)$	$Pr((\mathcal{M}(L_i), \Phi) = F)$	$Pr((\mathcal{M}(L_i), \Phi) =?)$
$\mathcal{M}(L_1)$	0.122	0.244	0.634
$\mathcal{M}(L_2)$	0.260	0.467	0.273
$\mathcal{M}(L_3)$	0.361	0.507	0.132
$\mathcal{M}(L_4)$	0.597	0.273	0.130

Table 2. Estimated probabilities of qBLTL property Φ being T, F or ?

$L_3 \prec L_2$. The network for the refined qDTMC $\mathcal{M}(L_3)$ is shown in Figure 2(c). The analysis results for $\mathcal{M}(L_3)$ for Φ are also shown in the same tables.

It is evident from the results of analysis of $\mathcal{M}(L_3)$ that the probability of the property being unknown has reduced significantly, thus allowing to make conclusive comments about the behaviour of the network. However, with these refinements $L_3 \prec L_2 \prec L_1$, the probability of Φ being false has also increased to 0.507. Thus, $\mathcal{M}(L_1)$ can alternately be refined to $\mathcal{M}(L_4)$, with $L_4 \prec L_1$, such that we ensure that the nodes with unknown get refined to true only. The network for the qDTMC $\mathcal{M}(L_4)$ is shown in Figure 2(d). Thus, it can be easily concluded from these results that while properties with low required probabilities can still be verified in incomplete models, the models may need refinements to satisfy properties demanding a higher success rate.

8 Conclusion

In this paper, we proposed a statistical analysis of stochastic systems with incomplete information. These incomplete systems are modelled using discrete time Markov chains with unknowns (qDTMC), and the required behaviour was formalized using qBLTL logic. By doing both quantitative and qualitative analysis of such systems using statistical model checking, we also proposed refinement on the qDTMCs. These refined qDTMCs depict a decrease in the probability of unknown behaviour in the system. The algorithms for both qualitative and quantitative analysis of qDTMC were implemented in the tool Plasma Lab. We demonstrated the working of these algorithms on a case study of a network with unknown information. We plan to extend this work to analyse the behaviour of other stochastic models like Markov decision processes and abstract Markov chains, with incomplete information.

References

1. Arora, S., Rao, M.V.P.: Probabilistic model checking of incomplete models. In: Proceedings of ISoLA. LNCS, vol. 9952, pp. 62–76 (2016)
2. Arora, S., Rao, M.V.P.: Probabilistic model checking of incomplete models. CoRR abs/1706.05082 (2017)
3. Bauer, A., Leucker, M., Schallhart, C.: The good, the bad, and the ugly, but how ugly is ugly? In: Proceedings of Runtime Verification. pp. 126–138. Springer (2007)
4. Boyer, B., Corre, K., Legay, A., Sedwards, S.: PLASMA-lab: A Flexible, Distributable Statistical Model Checking Library. In: Proceedings of QEST. LNCS, vol. 8054, pp. 160–164. Springer (2013)
5. Chechik, M., Easterbrook, S., Devereux, B.: Model checking with multi-valued temporal logics. In: Proceedings of the 31st IEEE International Symposium on Multiple-Valued Logic. pp. 187–192 (2001)
6. Chechik, M., Easterbrook, S., Petrovykh, V.: Model-checking over multi-valued logics. In: Formal Methods for Increasing Software Productivity. pp. 72–98. Springer (2001)
7. Chen, X.: Concentration inequalities for bounded random vectors. arXiv preprint arXiv:1309.0003 (2013)
8. Chernoff, H.: A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. Ann. Math. Statist. 23(4), 493–507 (1952)
9. Courcoubetis, C., Yannakakis, M.: The complexity of probabilistic verification. J. ACM 42(4), 857–907 (1995)
10. D'Argenio, P., Legay, A., Sedwards, S., Traonouez, L.: Smart sampling for lightweight verification of markov decision processes. STTT 17(4), 469–484 (2015)
11. David, A., Larsen, K.G., Legay, A., Mikucionis, M., Poulsen, D.B., van Vliet, J., Wang, Z.: Statistical model checking for networks of priced timed automata. In: Proceedings of FORMATS. LNCS, vol. 6919, pp. 80–96. Springer (2011)
12. Fecher, H., Leucker, M., Wolf, V.: Don't know in probabilistic systems. In: Model Checking Software. pp. 71–88. Springer (2006)
13. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. Formal Aspects of Computing 6(5), 512–535 (1994)
14. Hérault, T., Lassaigne, R., Magniette, F., Peyronnet, S.: Approximate Probabilistic Model Checking. In: Proceedings of VMCAI, LNCS, vol. 2937, pp. 73–84. Springer (2004)
15. Huth, M., Piterman, N., Wagner, D.: Three-valued abstractions of markov chains: Completeness for a sizeable fragment of pctl. In: Fundamentals of Computation Theory. pp. 205–216. Springer (2009)
16. Katoen, J.P., Klink, D., Leucker, M., Wolf, V.: Three-valued abstraction for probabilistic systems. Journal of Logic and Algebraic Programming 81(4), 356 – 389 (2012), special Issue: NWPT 2009
17. Kwiatkowska, M.Z., Norman, G., Parker, D.: PRISM 4.0: Verification of Probabilistic Real-Time Systems. In: Proceedings of CAV. LNCS, vol. 6806, pp. 585–591. Springer (2011)
18. Legay, A., Delahaye, B., Bensalem, S.: Statistical Model Checking: An Overview. In: Proceedings of Runtime Verification. LNCS, vol. 6418, pp. 122–135. Springer (2010)
19. Legay, A., Sedwards, S., Traonouez, L.: Plasma lab: A modular statistical model checking platform. In: ISoLA. LNCS, vol. 9952, pp. 77–93 (2016)

20. Legay, A., Sedwards, S., Traonouez, L.: Rare events for statistical model checking an overview. In: Reachability Problems. LNCS, vol. 9899, pp. 23–35. Springer (2016)
21. Legay, A., Traonouez, L.: Statistical model checking of simulink models with plasma lab. In: Proceedings of FTSCS. Communications in Computer and Information Science, vol. 596, pp. 259–264. Springer (2015)
22. Ngo, V.C., Legay, A., Quilbeuf, J.: Statistical model checking for systemc models. In: Proceedings of HASE. pp. 197–204. IEEE Computer Society (2016)
23. Vardi, M.Y.: Automatic verification of probabilistic concurrent finite state programs. In: Proceedings of SFCS. pp. 327–338. IEEE Computer Society (1985)
24. Wald, A.: Sequential tests of statistical hypotheses. The Annals of Mathematical Statistics 16(2), 117–186 (1945)
25. Younes, H.L.S.: Verification and Planning for Stochastic Processes with Asynchronous Events. Ph.D. thesis, Carnegie Mellon (2005)