Importance Sampling for Model Checking of Time-Bounded Until

Benoît Barbot, Serge Haddad, Claudine Picaronny

February 2012

Research report LSV-12-04

Laboratoire Spécification & Vérification

École Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex France

Importance Sampling for Model Checking of Time-Bounded Until

Benoît Barbot, Serge Haddad, Claudine Picaronny
LSV, ENS Cachan & CNRS & INRIA Cachan, France
{*barbot,haddad,picaronny*}*@lsv.ens-cachan.fr*

***Abstract*—Statistical model-checking is an alternative verification technique applied on stochastic systems whose size is beyond numerical analysis ability. Given a model (most often a Markov chain) and a formula, it provides a confidence interval for the probability that the model satisfies the formula. In a previous contribution, we have overtaken the main limitation of the statistical approach, i.e. the computation time explosion associated with the evaluation of very small probabilities. This method was valid only for the standard "Until" of temporal logics. We establish a similar validity condition which applies to the "Bounded Until", using more elaborate arguments. We also address the problem of additional memory requirements necessary to apply the method and we design several algorithms depending on the intended trade-off between time and memory. The corresponding algorithms have been implemented in our tool COSMOS. We present experimentations on several relevant systems, with drastic time reductions w.r.t. standard statistical model checking.**

***Keywords*-statistical model checking, rare events, importance sampling, time-bounded until**

I. INTRODUCTION

Probabilistic systems. Probabilistic systems have been intensively introduced and studied as they are used in a broad range of domains from analysing communication protocols to biological systems. They can model inherent random behaviour (e.g. an algorithm tossing a coin) as well as partial information (e.g. an open system whose probabilities are related to the uncertainity of the environment behaviour).

From performance evaluation to model checking. Model checking temporal formulas is a natural way to verify properties of a system behaviour [1]. Thanks to it algorithmic simplicity, it has been successfully implemented in a variety of tools. The system is described as a formal model and required properties are expressed by some temporal logic. Although a method initially dedicated to discrete event systems, it has been adapted both to timed systems [2] (e.g. modelled by timed automata) and stochastic ones [3] (e.g. modelled by Markov chains). For instance, in a vehicular system one would verify that "the delay between a shock and the airbag deploying is less than 10^{-2} s". Similarly if the system is prone to random perturbations, a relevant property could be "the probability that the airbag does not deploy after a shock is at most 10^{-6}".

Statistical model checking. Analysis of probabilistic systems may be undertaken using *numerical* or *statistical* techniques. Numerical methods give exact results (up to numerical approximations) but are subject to state explosion which significantly restricts the class of analysable systems (manageable size, Markov properties, etc.). Instead, for bigger systems, statistical method may be used. By simulating a big sample of trajectories of the system and computing the ratio of these trajectories that satisfy a given property, it produces a probabilistic framing of the expected value. To generate the sample we only need to have an operational stochastic semantic of the system. This usually requires a very small state space compared to the numerical method and allows to deal with huge models [4].

Rare events. The main drawback of the statistical model checking is its inefficiency in dealing with very small probabilities. The size of the sample of simulations required to estimate these small probabilities exceeds achievable capacities. This difficulty is known as the *rare event* problem. Several methods have been developed to cope with this problem whose main one is *importance sampling*. Importance sampling consists in modifying the model and in substituting to the indicator random variable related to the satisfaction of the formula, another variable with same mean and, in the favorable cases, reduced variance. Most of the techniques related to importance sampling are based on heuristics and cannot provide any confidence interval for the estimated probability. In [5], we proposed an efficient method based on importance sampling to estimate in a reliable way (the first one with a true, and not an approximate, confidence interval) the very small probability of a standard "Until" property (aUb) using coupling theory and we applied it on a large variety of case studies, modelled by discrete time Markov chains (DTMC).

Our contribution. The standard unbounded "Until" is sufficient to express logical properties on a probabilistic system but does not allow to handle the random nature of delays as in the requirement "the probability that the delay between a shock and the airbag deploying is more than 10^{-2} s is at most 10^{-6}". So here more generally, we handle the evaluation of a tiny probability associated with a "time-bounded until" formula in a DTMC. Extending our previous method, we propose a specific theoretical framework in order to perform an efficient importance sampling that guarantees a confidence interval in this context. Managing the time bound in our framework has required more elaborate technics both for proving its soundness and designing efficient algorithms. On the one hand, the soundness is established by a careful

study of the recursive equations defining the time-bounded until. On the other hand, since importance sampling now depends on the current time of the execution, it yields a space explosion when the horizon (i.e the upper bound of the interval) is far. This requires a trade-off between space and time for which we provide three different solutions depending on the reduction of the memory requirements. We implemented our method with the three options in the statistical model checker COSMOS [6]. We tested our tool on a classical relevant model getting impressive time or memory reductions.

Organisation. In section II, we motivate this work and we give a state of the art related to rare event handling. Then we develop our method in section III. Afterwards we present and discuss experimentation in section IV. Finally in section V, we conclude and give some perspectives to this work.

II. MOTIVATION AND STATE OF THE ART

Temporal logics for probabilistic systems include both the qualitative and quantitative aspects of the systems. They make possible to evaluate the probability that a random path fulfills some property (in CSL [7]). In a more general setting, they allow to estimate the conditional expectation of a path random variable whose condition is the satisfaction of some property by the random path (in HASL [6]).

Model checking of "timed-bounded until" formulas has been studied and used to express relevant properties of numerous modelings since the founding introduction of logic PCTL [8]. Theoretical analysis and specific studies are essential because of the exponentially increasing complexity in terms of the horizon of such a property.

Basically, there are two approaches to perform model checking of these logics: numerical or statistical. The first one builds the underlying stochastic process of the model and then computes probabilities or expectations using direct or iterative methods. Such methods have been implemented efficiently in tools like PRISM [9], LiQuor [10] or MRMC [11]. These methods have two drawbacks. On the one hand, they rely on strong assumptions about the stochastic process that must be a Markov chain (see for instance [7]) or at least a regenerative process (see for instance [12]). On the other hand they suffer from the combinatorial explosion of the size of the stochastic process w.r.t. the size of the model.

Models with huge stochastic process are handled by statistical model checking. The corresponding methods randomly generate a (large) set of execution paths and check whether the paths fulfill the formula. The result is a probabilistic estimation of the satisfaction given by a confidence interval [13]. In principle, it only requires to maintain a current state (and some numerical values in case of a non Markovian process). Furthermore no regenerative assumption is required and it is easier to parallelize the methods. Several tools include statistical model checking: COSMOS [6], GREATSPN [14], PRISM [9], UPPAAL [15], VESTA [16], YMER [17].

Model checking of probabilistic systems is particularly important for events which have disastrous consequences (loss of human life, financial ruin, etc.), but occur with very small probability. Unfortunately statistical model checking of *rare events* triggers a computation time explosion, forbidding its use. To illustrate this point, suppose one wants to estimate an unknown probability $p = 10^{-13}$ and one chooses to generate 10^{10} paths (which is already a large number) for such an estimation. With probability larger than 0.999 the result is 0, giving no information on the value of p. With probability smaller than 0.001 the result will be greater or equal than 10^{-10} which is a very crude estimation.

Thus *acceleration* techniques [18] have been introduced to cope with this problem. The two main families of methods are *splitting* and *importance sampling*. Splitting methods [19] are by nature heuristics and model dependent. Importance samplings methods [20] are more robust as they possess an optimal result generally impossible to compute, but allowing to design efficient heuristics for some classes of models. The goal is to substitute to the Bernoulli random variable corresponding to the occurrence of the rare event, another one with same mean value (the probability of event occurrence) but smaller variance. In Markov chains, an optimal change of distribution exists leading to a zero variance but it requires more information than the searched value!

The modification of the distribution can be performed at the model level (called *static*) or at the Markov chain level (called *dynamic*). The static importance sampling requires no additional memory but in general provides a smaller reduction of variance than the dynamic importance sampling. More precisely, it is proved in [21] that asymptotic optimality (a weaker requirement than optimality) cannot be obtained even for very simple classes of models by static importance sampling. In full generality, the dynamic importance sampling [22] requires to maintain a memory whose size is proportional to the size of the Markov chain which is exactly what one wants to avoid. To deal with this problem, in [23] the authors develop the following method: (1) the possible distributions belong to the convex hull of a finite number of distributions, (2) the state space is partitioned and (3) a distribution is selected for each subset of this partition. They prove that for a simple class of models their method is asymptotically optimal. Other empirical approaches turn out to be efficient [24].

Summarizing, theoretical results (reduction of variance, asymptotical optimality, etc.) have been obtained for importance sampling but does not provide any reliable confidence interval [1] for the mean value since the distribution of the

[1] In contrast to the empirical confidence interval based on approximations by the normal distribution.

modified random variable is unknown.

In [5], we proposed an efficient method based on importance sampling to estimate the tiny probability of a standard "Until" property (aUb). We constrained the importance sampling method to be *with guaranteed variance*, a property that ensures a true confidence interval framing this probability under some conditions. We established a theoretical framework based on coupling theory in which only structural analysis is required to verify those hypotheses. We implemented the whole method in our tool COSMOS and applied it successfully on numerous examples. To our knowledge, the more general case of model checking "time-bounded until" formulas corresponding to rare events has never been studied.

III. GENERAL APPROACH

A. Preliminaries

Markov Chains and Model Checking

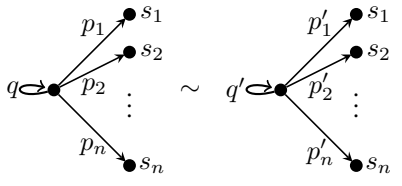
Definition 1: A discrete time Markov chain (DTMC) $\mathcal{C}$ is defined as a set of states S, an initial state s_0, and a transition probability matrix $\mathbf{P}$ of size $S \times S$. The state of the chain at time n is a random variable X_n defined inductively by $\Pr(X_0 = s_0) = 1$ and $\Pr(X_{n+1} = s' \mid X_n = s, X_{n-1} = s_{n-1}, \ldots, X_0 = s_0) = \Pr(X_{n+1} = s' \mid X_n = s) = \mathbf{P}(s, s')$.

For our purpose, we enrich the usual definition of Markov chains with labels on transitions. When these labels are not useful, we simply omit them.

Definition 2: An enriched discrete time Markov chain $\mathcal{C}$ is defined by a set of states S, an initial state s_0, a finite set of events E, a successor function $\delta : S \times E \to S$, and a function $p : S \times E \to [0; 1]$ with the property that for all $s \in S$, $\sum_{e \in E} p(s, e) = 1$. We define the transition probability matrix $\mathbf{P}$ of size $S \times S$ by:

$$\forall s, s' \in S,\ \mathbf{P}(s, s') = \sum_{\delta(s,e)=s'} p(s, e)$$

Two Markov chains are *equivalent* if they have the same set of states S, respective probability distribution matrices $\mathbf{P}$, $\mathbf{P}'$ and for all state s with $q = \mathbf{P}(s, s)$, $q' = \mathbf{P}'(s, s)$ we have the following equalities:



$$\forall s' \neq s\ \frac{\mathbf{P}(s, s')}{1 - q} = \frac{\mathbf{P}'(s, s')}{1 - q'}$$

This equivalence is used in an implicit way in the proofs. We will often omit self-loops: we consider that a self-loop always exists with a probability such that the sum of all outgoing transitions probability is equal to 1.

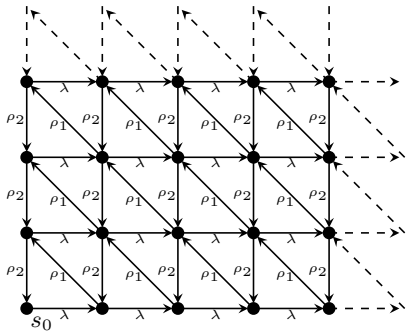


Figure 1. DTMC for the tandem queues

Example The figure III-A represents a Markov chain of a tandem queue system. This system contains two queues, the number of clients in the first queue is represented on the horizontal axis and the number of clients in the second one is represented on the vertical axis. In the initial state s_0, the two queues are empty. Given some state, a new client comes in the first queue with probability λ, a client leaves the first queue for the second one with probability ρ_1 and a client leaves the second queue and exits with probability ρ_2 ($\lambda + \rho_1 + \rho_2 = 1$). An impossible event (due to the emptiness of some queue) corresponds to an event leaving unchanged the state. These loops are not represented in the figure.

Usually the modeller does not specify its system with a Markov chain. He rather defines a higher level model $\mathcal{M}$ (a queueing network, a stochastic Petri net, etc.), whose operational semantic is a Markov chain $\mathcal{C}$.

In the context of model checking, the states of chain $\mathcal{C}$ are labelled with atomic propositions that they fulfill. Given state s, $\alpha(s)$ denotes the set of propositions satisfied by s. We denote $S_x = \{s \in S \mid x \in \alpha(s)\}$, $S_{\overline{x}} = \{s \in S \mid x \notin \alpha(s)\}$, $S_{xy} = \{s \in S \mid x \in \alpha(s) \wedge y \in \alpha(s)\}$, etc.

The problem we address here is the computation of the probability that a random path starting from a fixed state s (and in particular from the initial state) satisfies a formula $aU^I b$ where U is the *Until* operator, a, b are atomic propositions and I is an integer interval. We note $[l, \infty]$ the unbounded intervals and we adopt the convention $\infty - 1 = \infty$. Then these probabilities, denoted $\mu_I(s)$, can be shown to be the smallest solution of the following system of equations ($\mathbf{1}_E$ denotes the indicator function of set E).

$$\begin{cases} \mu_I(s) = 0 & \forall s \in S_{\overline{a \vee b}} \\ \mu_{[0,0]} = \mathbf{1}_{S_b} & \\ \mu_{[0,u]}(s) = 1 & \forall u > 0\ \forall s \in S_b \\ \mu_{[0,u]}(s) = \sum_{s' \in S} \mathbf{P}(s, s') \mu_{[0,u-1]}(s') & \forall u > 0\ \forall s \in S_{a\overline{b}} \\ \mu_{[l,u]}(s) = 0 & \forall l > 0\ \forall s \in S_{\overline{a}} \\ \mu_{[l,u]}(s) = \sum_{s' \in S} \mathbf{P}(s, s') \mu_{[l-1,u-1]}(s') & \forall l > 0\ \forall s \in S_a \end{cases}$$

Coupling for Model Checking

The coupling method [25] is a classical method for comparing two stochastic processes, applied in different contexts (establishing ergodicity of a chain, stochastic ordering, bounds, etc.). In the sequel we will develop a new

application for coupling. A coupling between two Markov chains is a chain whose space is a subset of the product of the two spaces which satisfies: (1) the projection of the product chain on any of its components behaves like the original corresponding chain, (2) an additional constraint which depends on the property to be proved (here related to the until formula). In this paper, we only need to define the coupling of a chain with itself.

Definition 3: Let $\mathcal{C} = (S, \mathbf{P})$ be a labelled chain and $\varphi \equiv aU^{[l,u]}b$ be a formula. A coupling of $\mathcal{C}$ w.r.t. φ is a DTMC $\mathcal{C}^{\otimes} = (S^{\otimes}, \mathbf{P}^{\otimes})$ such that :

- $S^{\otimes} \subseteq S \times S$
- $\forall s \neq s_1 \in S\ ,\ \forall (s, s') \in S^{\otimes}$, $\mathbf{P}\ (s\ , s_1) = \sum_{(s_1, s_1') \in S^{\otimes}} \mathbf{P}^{\otimes}((s, s'), (s_1, s_1'))$ and $\forall s' \neq s_1' \in S,\ \forall (s, s') \in S^{\otimes}$, $\mathbf{P}(s', s_1') = \sum_{(s_1, s_1') \in S^{\otimes}} \mathbf{P}^{\otimes}((s, s'), (s_1, s_1'))$
- $\forall (s, s') \in S^{\otimes}\ s' \in S_b \Rightarrow s \in S_b$
- $\forall (s, s') \in S^{\otimes}\ s \in S_{\overline{ab}} \Rightarrow s' \in S_{\overline{ab}}$
- If $l > 0$ then $\forall (s, s') \in S^{\otimes}\ s \in S_{\overline{a}} \Rightarrow s' \in S_{\overline{a}}$

The set $S^{\otimes}$ defines a coupling relation of the chain.

The following proposition allows to compare probabilities without any numerical computation. As before, $\mu_I(s)$ denotes the probability that a random path starting from s satisfies $aU^I b$.

Proposition 1: Let $\mathcal{C}^{\otimes}$ be a coupling of $\mathcal{C}$ related to $aU^I b$ Then for all $(s, s') \in S^{\otimes}$, we have:

$$\mu_I(s) \geq \mu_I(s')$$

Proof

We first observe that the property on the coupling only depends on a, b and whether $l > 0$. So we prove the property by induction on the intervals.

Let $(s, s') \in S^{\otimes}$, since $\mu_{[0,0]} = \mathbf{1}_{S_b}$ and $s' \in S_b \Rightarrow s \in S_b$, $\mu_{[0,0]}(s) \geq \mu_{[0,0]}(s')$.

Now we prove the property for interval $[0, u]$ with finite u by induction on $u > 0$ with basis case $u = 0$ already proved.
If $s \in S_b$ then $\mu_{[0,u]}(s) = 1 \geq \mu_{[0,u]}(s')$
If $s' \in S_b$ then $s \in S_b$ and $\mu_{[0,u]}(s) = \mu_{[0,u]}(s') = 1$
If $s \in S_{\overline{ab}}$ then $s' \in S_{\overline{ab}}$ and $\mu_{[0,u]}(s) = \mu_{[0,u]}(s') = 0$
If $s' \in S_{\overline{ab}}$ then $\mu_{[0,u]}(s') = 0 \leq \mu_{[0,u]}(s)$
The last case to consider is $s, s' \in S_{a\overline{b}}$.
$\mu_{[0,u]}(s) = \sum_{s_1 \in S} \mathbf{P}(s, s_1)\mu_{[0,u-1]}(s_1)$
$= \sum_{s_1 \in S} \sum_{(s_1, s_1') \in S^{\otimes}} \mathbf{P}^{\otimes}((s, s'), (s_1, s_1'))\mu_{[0,u-1]}(s_1)$
$\geq \sum_{s_1 \in S} \sum_{(s_1, s_1') \in S^{\otimes}} \mathbf{P}^{\otimes}((s, s'), (s_1, s_1'))\mu_{[0,u-1]}(s_1')$
$= \sum_{s_1' \in S} \sum_{(s_1, s_1') \in S^{\otimes}} \mathbf{P}^{\otimes}((s, s'), (s_1, s_1'))\mu_{[0,u-1]}(s_1')$
$= \sum_{s_1' \in S} \mathbf{P}(s', s_1')\mu_{[0,u-1]}(s_1') = \mu_{[0,u]}(s')$

Since $\mu_{[0,\infty]}(s) = \lim_{u \to \infty} \mu_{[0,u]}(s)$, this proves the result for interval $[0, \infty]$.

Now we prove the property for interval $[l, u]$ with finite or infinite u by induction on $l > 0$ with basis case $l = 0$ already proved.
If $s \in S_{\overline{a}}$ then $s' \in S_{\overline{a}}$ and $\mu_{[l,u]}(s) = \mu_{[l,u]}(s') = 0$
If $s' \in S_{\overline{a}}$ then $\mu_{[l,u]}(s') = 0 \leq \mu_{[l,u]}(s)$

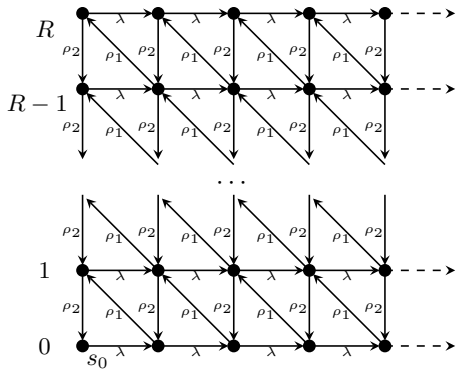


Figure 2. Reduced DTMC $\mathcal{C}^{\bullet}$

The last case to consider is $s, s' \in S_a$.
$\mu_{[l,u]}(s) = \sum_{s_1 \in S} \mathbf{P}(s, s_1)\mu_{[l-1,u-1]}(s_1)$
$= \sum_{s_1 \in S} \sum_{(s_1, s_1') \in S^{\otimes}} \mathbf{P}^{\otimes}((s, s'), (s_1, s_1'))\mu_{[l-1,u-1]}(s_1)$
$\geq \sum_{s_1 \in S} \sum_{(s_1, s_1') \in S^{\otimes}} \mathbf{P}^{\otimes}((s, s'), (s_1, s_1'))\mu_{[l-1,u-1]}(s_1')$
$= \sum_{s_1' \in S} \sum_{(s_1, s_1') \in S^{\otimes}} \mathbf{P}^{\otimes}((s, s'), (s_1, s_1'))\mu_{[l-1,u-1]}(s_1')$
$= \sum_{s_1' \in S} \mathbf{P}(s', s_1')\mu_{[l-1,u-1]}(s_1') = \mu_{[l,u]}(s')$

c.q.f.d. $\Diamond\Diamond\Diamond$

Example *Let us illustrate coupling for the Markov chain represented in figure 2 and called $\mathcal{C}^{\bullet}$. This chain is obtained from the tandem queues by lumping together states which have the same number of clients and at least R clients in the second queue. Its set of state is $S^{\bullet} = \mathbb{N} \times [0..R]$. Consider the coupling of this chain with itself defined by $S^{\otimes} = \{((n_1, n_2), (n_1', n_2')) \mid n_1 + n_2 \geq n_1' + n_2' \wedge n_1 \geq n_1'\}$. Consider now proposition a "There is at least one client in some queue" and proposition b "The sum of the number of clients in both queues is at least 5". (we consider that the initial state s_0 is the state with one client in the first queue to avoid trivial evaluation).*

Lemma 1: $S^{\otimes}$ is a coupling relation w.r.t. $aU^I b$ for any interval I.

Proof

We recall the coupling relation
a) $n_1 + n_2 \geq n_1' + n_2'$
b) $n_1 \geq n_1'$
We need to check that for all couples $((n_1, n_2), (n_1', n_2'))$ in the relation, all successors are also in the relation. We have three different types of transitions in the system:

1) *For transition λ, the successor of $((n_1, n_2), (n_1', n_2'))$ is $((n_1 + 1, n_2), (n_1' + 1, n_2'))$ which is inside the relation.*
2) *For transition ρ_1, the successor of $((n_1, n_2), (n_1', n_2'))$ is*
   $((n_1 - \mathbf{1}_{\{n_1 > 0 \wedge n_2 < R\}}, n_2 + \mathbf{1}_{\{n_1 > 0 \wedge n_2 < R\}}),$
   $(n_1' - \mathbf{1}_{\{n_1' > 0 \wedge n_2' < R\}}, n_2' + \mathbf{1}_{\{n_1' > 0 \wedge n_2' < R\}}))$
   the condition a) is satisfied as the sum is not modified.
   - *If $n_1 > n_1'$ then $n_1 - \mathbf{1}_{\{n_1 > 0 \wedge n_2 < R\}} \geq n_1' - \mathbf{1}_{\{n_1' > 0 \wedge n_2' < R\}}$ and condition b) is satisfied.*

- *Else* $n_1 = n'_1$ *and with the condition a) we have* $n_2 \geq n'_2$ *; then*
  $n_1 > 0 \wedge n_2 < R \Rightarrow n'_1 > 0 \wedge n'_2 < R$ *which implies :*
  $n_1 - \mathbf{1}_{\{n_1>0 \wedge n_2<R\}} \geq n'_1 - \mathbf{1}_{\{n'_1>0 \wedge n'_2<R\}}$ *then conditionb) holds.*

3) *For transition* ρ_2*, the successor of* $((n_1, n_2), (n'_1, n'_2))$ *is*
$((n_1, n_2 - \mathbf{1}_{\{n_2>0\}}), (n'_1, n'_2 - \mathbf{1}_{\{n'_2>0\}}))$.
As the first component is not modified, the condition b) holds.
   - *If* $n_1 + n_2 > n'_1 + n'_2$ *then* $n_1 + n_2 - \mathbf{1}_{\{n_2>0\}} \geq n'_1 + n'_2 - \mathbf{1}_{\{n'_2>0\}}$
   - *Else* $n_1 + n_2 = n'_1 + n'_2$ *and with condition b)* $n_2 \leq n'_2$ *then* $n_2 - \mathbf{1}_{\{n_2>0\}} \geq n'_2 - \mathbf{1}_{\{n'_2>0\}}$*, the condition a) holds.*

We now have to check the properties related to atomic propositions a *and* b*.*

- *Let* $(n'_1, n'_2) \in S_b$ *which means that* $n'_1 + n'_2 \geq 5$*. Since* $n_1 + n_2 \geq n'_1 + n'_2 \geq 5$*,* $(n_1, n_2) \in S_b$*.*
- *Let* $(n_1, n_2) \in S_{\overline{a}}$ *which means that* $n_1 + n_2 = 0$*, Since* $0 = n_1 + n_2 \geq n'_1 + n'_2$*,* $(n'_1, n'_2) \in S_{\overline{a}}$*.*
- *Combining the two first items,* $(n_1, n_2) \in S_{\overline{ab}}$ *implies* $(n'_1, n'_2) \in S_{\overline{ab}}$*.*

Then $S^{\otimes}$ *is a coupling relation.*

c.q.f.d. $\Diamond\Diamond\Diamond$

Importance Sampling for Reachability Analysis

We consider a Markov chain $\mathcal{C}$ with two *absorbing* states s_+ or s_-, i.e. $\mathbf{P}(s_-, s_-) = \mathbf{P}(s_+, s_+) = 1$. We require that the probability to reach s_+ or s_- from any state is equal to 1.

The statistical approach consists in generating K paths of the Markov chain $\mathcal{C}$ which ends in an absorbing state.

Let K_+ be the number of paths ending in the s_+ state. The random variable K_+ follows a binomial distribution with parameters p and K. Thus the random variable $\frac{K_+}{K}$ has a mean value p and a variance $\frac{p-p^2}{K}$. When K goes to infinity the variance goes to 0. In order to be more precise on the estimation, we introduce the notion of confidence interval.

Definition 4: Let $X_1, \ldots, X_n$ be independent random variables following a common distribution including a parameter θ. Let $0 < \gamma < 1$ be a confidence level. Then a confidence interval for θ with level at least γ is given by two random variables $m(X_1, \ldots, X_n)$ and $M(X_1, \ldots, X_n)$ such that for all θ:

$$\Pr\left(m(X_1, \ldots, X_n) \leq \theta \leq M(X_1, \ldots, X_n)\right) \geq \gamma$$

For standard parametrized distributions like the normal or the Bernoulli ones, it is possible to compute confidence intervals [13]. Thus, given a number of paths K and a confidence level $1 - \varepsilon$, the method produces a confidence interval. As discussed before when $p \ll 1$, the number of paths required for a small confidence interval is too large to be simulated.

The importance sampling method uses a modified transition matrix $\mathbf{P}'$ during the generation of paths. $\mathbf{P}'$ must satisfy:

$$\mathbf{P}(s, s') > 0 \Rightarrow \mathbf{P}'(s, s') > 0 \vee s' = s_- \tag{1}$$

which means that this modification cannot remove transitions that have not s_- as target, but can add new transitions. The method maintains a correction factor called L initialized to 1; this factor represents the *likelihood* of the path. When a path crosses a transition $s \to s'$ with $s' \neq s_-$, L is updated by $L \leftarrow L\frac{\mathbf{P}(s,s')}{\mathbf{P}'(s,s')}$. When a path reaches s_-, L is set to zero. If $\mathbf{P}' = \mathbf{P}$ (i.e. no modification of the chain), the value of L when the path reaches s_+ (resp. s_-) is 1 (resp. 0).

Let V_s (resp. W_s) be the random variable associated with the final value of L for a path starting in x in the original model $\mathcal{C}$ (resp. in $\mathcal{C}'$). By definition, $\mathbf{E}(V_{s_0}) = p$. The following proposition establishes the correctness of the method.

Proposition 2: $\mathbf{E}(W_{s_0}) = p$.

Proof

In all states, the probability to reach s_- or s_+ is equal to 1. Then thanks to a classic result on Markov chains the expected value of the r.v. V_s is the unique solution of the following system of equations:

$$\begin{aligned} \mathbf{E}(V_{s_-}) = 0 \ \wedge \ \mathbf{E}(V_{s_+}) = 1 \ \wedge \ \forall s \notin \{s_-, s_+\} \\ \mathbf{E}(V_s) = \sum_{s' \neq s_-} \mathbf{P}(s, s')\mathbf{E}(V_{s'}) \end{aligned} \tag{2}$$

We now write the corresponding system for $\mathbf{P}'$ with correction factor:

$$\begin{aligned} \mathbf{E}(W_{s_-}) = 0 \ \wedge \ \mathbf{E}(W_{s_+}) = 1 \wedge \ \forall s \notin \{s_-, s_+\} \\ \mathbf{E}(W_s) = \sum_{s' \neq s_- \wedge \mathbf{P}'(s,s') > 0} \mathbf{P}'(s, s') \left(\frac{\mathbf{P}(s, s')}{\mathbf{P}'(s, s')}\right) \mathbf{E}(W_{s'}) \end{aligned} \tag{3}$$

Thanks to the restriction of equation 1, the two systems are equal after simplification, and we have $\mathbf{E}(W_{s_0}) = \mathbf{E}(V_{s_0}) = p$.

c.q.f.d. $\Diamond\Diamond\Diamond$

Proof of proposition 3

If $\mu(s) = 0$ then all trajectories starting in s end in s_-. Therefore the variance is null.

If $\mu(s) \neq 0$, thanks to the equation

$$\mu(s) = \sum_{s' | \mu(s') > 0} \mathbf{P}(s, s')\mu(s')$$

$\mathbf{P}'(s, -)$ is a distribution. A trajectory starting from a state s with $\mu(s) > 0$ visits only states s' with $\mu(s') > 0$, so it ends in s_+. Denoting by $s =$

$u_0, \ldots, u_l = s_+$ such a trajectory, the value L is equal to $(\mu(u_0)/\mu(u_1)) \ldots (\mu(u_{l-1})/\mu(u_l)) = \mu(s)$.

c.q.f.d. $\Diamond\Diamond\Diamond$

A good choice of $\mathbf{P}'$ should reduce the variance of W_{s_0} w.r.t. to variance of V_{s_0}. The following proposition shows that there exists a matrix $\mathbf{P}'$ which leads to a null variance. We denote the probability to reach s_+ starting from s by $\mu(s)$.

Proposition 3: Let $\mathbf{P}'$ be defined by

- $\forall s$ such that $\mu(s) \neq 0$, $\mathbf{P}'(s,s') = \frac{\mu(s')}{\mu(s)}\mathbf{P}_u(s,s')$
- $\forall s$ such that $\mu(s) = 0$, $\mathbf{P}'(s,s') = \mathbf{P}(s,s')$

Then for all s, we have $\mathbf{V}(W_s) = 0$.

This result has a priori no practical application since it requires the knowledge of μ for all states, whereas we only want to estimate $\mu(s_0)$!

From Model Checking to Reachability

We now relate the computation of $\mu_I(s)$ in a Markov chain $\mathcal{C}$ to a reachability problem in a Markov chain $\mathcal{C}_I$ which depends both on $\mathcal{C}$ and on I. From now on, we focus on the case $I = [0,u]$ with $0 < u < \infty$ (and $s \in S_{a\bar{b}}$) and we postpone to paragraph III-D the management of the other cases. So we simplify the notations with μ_u (resp. $\mathcal{C}_u$) instead of $\mu_{[0,u]}$ (resp. $\mathcal{C}_{[0,u]}$).

The Markov chain C_u is defined by:

- $S_u = S_{a\bar{b}} \times [1,u] \cup \{s_-, s_+\}$
- s_-, s_+ are *absorbing* states:
  $\mathbf{P}_u(s_-,s_-) = \mathbf{P}_u(s_+,s_+) = 1$
- $\forall s,s'\ \forall \tau > 1\ \mathbf{P}_u((s,\tau),(s',\tau-1)) = \mathbf{P}(s,s')$,
  $\mathbf{P}_u((s,\tau),s_-) = \sum_{s' \in S_{\overline{a\bar{b}}}} \mathbf{P}(s,s')$,
  $\mathbf{P}_u((s,\tau),s_+) = \sum_{s' \in S_b} \mathbf{P}(s,s')$.
- $\forall s\ \mathbf{P}_u((s,1),s_+) = \sum_{s' \in S_b} \mathbf{P}(s,s')$,
  $\mathbf{P}_u((s,1),s_-) = 1 - \mathbf{P}_u((s,1),s_+)$,
- The other transition probabilities are null.

Example Figure III-A describes chain $\mathcal{C}_u$ associated with the example.

First observe that the probability to reach s_+ or s_- from any state is equal to 1. Moreover by construction, $\mu(s,\tau) = \mu_\tau(s)$ where $\mu(s,\tau)$ is a reachability property in $\mathcal{C}_u$ and $\mu_\tau(s)$ is the probability of satisfying a formula in $\mathcal{C}$.

B. An Importance Sampling Method with Variance Reduction and Confidence Interval

The proposed method performs statistical model checking on $\mathcal{C}$ by statistically computing a reachability probability in $\mathcal{C}_u$ using importance sampling with associated matrix obtained by numerical model checking on a reduced chain whose formal definition is given below.

Definition 5: Let $\mathcal{C}$ be a DTMC, a DTMC $\mathcal{C}^\bullet$ is called a *reduction* of $\mathcal{C}$ by a function f that maps S to $S^\bullet$, the state space of $\mathcal{C}^\bullet$, if for all $s \in S$:

- $a \in \alpha(s)$ (resp. $b \in \alpha(s)$) iff $a \in \alpha^\bullet(f(s))$ (resp. $b \in \alpha^\bullet(f(s))$)
- $\forall 0 < \tau \leq u\ \mu^\bullet_\tau(f(s)) = 0 \Rightarrow \mu_\tau(s) = 0$
  where $\mu^\bullet_\tau(s^\bullet)$ denotes the probability that a random path in $\mathcal{C}^\bullet$ starting from $s^\bullet$ satisfies $aU^{[0,\tau]}b$.

Two states s and s' are *equivalent* if $f(s) = f(s')$, in other words f^{-1} define equivalence classes for this reduction.

Example In the example of tandem queues, the reduced chain $\mathcal{C}^\bullet$ is obtained from the original chain by applying the following function to the state space.

$$f(n_1,n_2) = \begin{cases} (n_1,n_2) & \text{if } n_2 \leq R \\ (n_1+n_2-R,R) & \text{otherwise} \end{cases}$$

The intuition behind this reduction is to block clients in the first queue when there are R clients in the second one, thus increasing the probability of a global overflow.

Given some reduced chain $\mathcal{C}^\bullet$, our goal is to replace the random variable (r.v.) V_{s_0} which takes value in $\{0,1\}$ by a r.v. W_{s_0} which takes value in $\{0, \mu^\bullet(f(s_0))\}$. This requires that $\mu_u(s_0) \leq \mu^\bullet_u(f(s_0))$. By applying an homogeneity principle, we get the stronger requirement $\forall s \in S, \forall 0 \leq \tau \leq u, \mu_\tau(s) \leq \mu^\bullet_\tau(f(s))$. In fact, the appropriate requirement which implies the previous one (see later proposition 4) is expressed by the next definition.

Definition 6: Let $\mathcal{C}$ be a DTMC and $\mathcal{C}^\bullet$ a reduction of $\mathcal{C}$ by f. $\mathcal{C}^\bullet$ is a *reduction with guaranteed variance* if for all $s \in S$ such that $\mu^\bullet(f(s)) > 0$ and $\forall 0 < \tau \leq u$ we have :

$$\sum_{s' \in S} \mu^\bullet_{\tau-1}(f(s')) \cdot \mathbf{P}(s,s') \leq \mu^\bullet_\tau(f(s)) \tag{4}$$

Given $s \in S$ and $0 < \tau \leq u$, let $h_\tau(s)$ be defined by $h_\tau(s) = \sum_{s' \in S} \frac{\mu^\bullet_{\tau-1}(f(s'))}{\mu^\bullet_\tau(f(s))}\mathbf{P}(s,s')$. We can now construct an efficient important sampling based on a reduced chain with guaranteed variance.

Definition 7: Let $\mathcal{C}$ be a DTMC and $\mathcal{C}^\bullet$ be a reduction of $\mathcal{C}$ by f with guaranteed variance. Then $\mathbf{P}'_u$ is the transition matrix on S_u the state space of C_u defined by:
Let s be a state of $S_{a\bar{b}}$ and $0 < \tau \leq u$,

- if $\mu^\bullet_\tau(f(s)) = 0$ then for all $s' \in S_u$
  $\mathbf{P}'_u((s,\tau),s') = \mathbf{P}_u((s,\tau),s')$
- if $\mu^\bullet_\tau(f(s)) > 0$ and $\tau > 1$ then
  - $\forall s' \in S_{a\bar{b}}$
    $\mathbf{P}'_u((s,\tau),(s',\tau-1)) = \frac{\mu^\bullet_{\tau-1}(f(s'))}{\mu^\bullet_\tau(f(s))}\mathbf{P}(s,s')$
  - $\mathbf{P}'_u((s,\tau),s_+) = \mu^\bullet_\tau(f(s))^{-1} \sum_{s' \in S_b} \mathbf{P}(s,s')$
  - $\mathbf{P}'_u((s,\tau),s_-) = 1 - h_\tau(s)$.
- if $\mu^\bullet_1(f(s)) > 0$ then
  - $\mathbf{P}'_u((s,1),s_+) = \mu^\bullet_1(f(s))^{-1} \sum_{s' \in S_b} \mathbf{P}(s,s')$
  - $\mathbf{P}'_u((s,\tau),s_-) = 1 - h_1(s)$.

The following proposition justifies the definition of $\mathbf{P}'_u$.

Proposition 4: Let $\mathcal{C}$ be a DTMC and $\mathcal{C}^\bullet$ be a reduction with guaranteed variance. The importance sampling based on matrix of $\mathbf{P}'_u$ definition 7 has the following properties:

- For all s and all $0 < \tau \leq u$ such that $\mu(s,\tau) > 0$, $W_{(s,\tau)}$ is a random variable which has value in $\{0, \mu^\bullet_\tau(f(s))\}$.

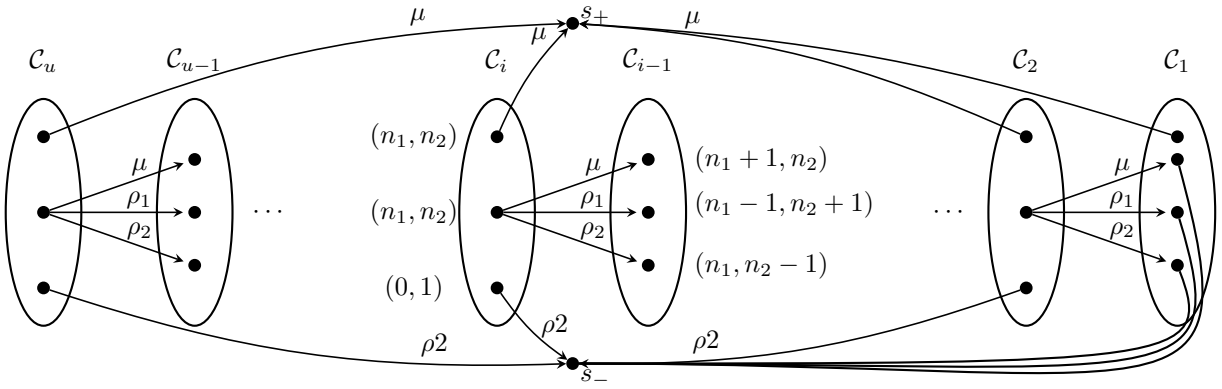


Figure 3. "Unfolding $\mathcal{C}_u$"

- $\mu_\tau(s) \leq \mu^\bullet_\tau(f(s))$ and $\mathbf{V}(W_{(s,\tau)}) = \mu_\tau(s)\mu^\bullet_\tau(f(s)) - \mu^2_\tau(s)$.
- One can compute a confidence interval for this importance sampling.

Proof

Let $s = (s, \tau) = u_0, \ldots, u_l = s_+$ be a trajectory starting in s ending in s_+. We observe that due to the construction of $\mathcal{C}_u$, any u_k can be written as $(s_k, \tau - k)$ with $0 \leq k < l$. As the trajectory avoids s_-, its value is:
$(\mu^\bullet_\tau(f(s))/\mu^\bullet_{\tau-1}(f(s_1))) \cdots (\mu^\bullet_{\tau-l+2}(f(s_{l-2}))/\mu^\bullet_{\tau-l+1}(f(s_{l-1})))$
$\mu^\bullet_{\tau-l+1}(f(u_{l-1})) = \mu^\bullet(f(s))$

We know that $\mathbf{E}(W_{(s,\tau)}) = \mu_\tau(s)$, then $\mathbf{P}(W_{(s,\tau)} = \mu^\bullet(f(s))) = \frac{\mu_\tau(s)}{\mu^\bullet_\tau(f(s))}$. This implies that $\mu_\tau(s) \leq \mu^\bullet_\tau(f(s))$ and $\mathbf{V}(W_{(s,\tau)}) = \mu_\tau(s)\mu^\bullet_\tau(f(s)) - \mu^2_\tau(s)$. As $W_{(s,\tau)}$ takes only two values, as for a Bernoulli law, it is possible to compute a confidence interval.

c.q.f.d. $\lozenge\lozenge\lozenge$

Since $\mu_u(s_0) \ll 1$, $\mathbf{V}(V_{s_0}) \approx \mu_u(s_0)$. If $\mu_u(s_0) \ll \mu^\bullet_u(f(s_0))$, we obtain $\mathbf{V}(W_{(s_0,u)}) \approx \mu_u(s_0)\mu^\bullet_u(f(s_0))$, so the variance is reduced by a factor $\mu^\bullet_u(f(s_0))$. In the case where $\mu_u(s_0)$ and $\mu^\bullet_u(f(s_0))$ have same magnitude order, the reduction of variance is even bigger.

Unfortunately, Inequation (4) requires to compute the functions $\mu^\bullet_\tau$ in order to check that $\mathcal{C}^\bullet$ is a reduction with guaranteed variance. We are looking for a structural requirement that does not involve the computation of $\mu^\bullet_\tau$.

Proposition 5: Let $\mathcal{C}$ be a DTMC, $\mathcal{C}^\bullet$ be a reduction of $\mathcal{C}$ by f. Assume there exists a family of functions $(g_s)_{s\in S}$, $g_s : \{t \mid \mathbf{P}(s,t) > 0\} \to S^\bullet$ such that:

1) $\forall s \in S,\ \forall t^\bullet \in S^\bullet$,
   $\mathbf{P}^\bullet(f(s), t^\bullet) = \sum_{s'|g_s(s')=t^\bullet} \mathbf{P}(s, s')$
2) $\forall s, t \in S$ such that $\mathbf{P}(s,t) > 0\ \forall 0 \leq \tau < u$
   $\mu^\bullet_\tau(f(t)) \leq \mu^\bullet_\tau(g_s(t))$

Then $\mathcal{C}^\bullet$ is a reduction of $\mathcal{C}$ with guaranteed variance.

Proof

Let s be a state of S and $\tau > 0$. We partition the terms of the sum of the inequation (4) according to their images by the function g_s:

$$\sum_{s'|\mathbf{P}(s,s')>0} \mu^\bullet_{\tau-1}(f(s')) \cdot \mathbf{P}(s,s') =$$

$$= \sum_{s^\bullet \in S^\bullet} \sum_{s'|g_s(s')=s^\bullet} \mu^\bullet_{\tau-1}(f(s')) \cdot \mathbf{P}(s,s')$$

We apply the second hypothesis:

$$\leq \sum_{s^\bullet \in S^\bullet} \sum_{s'|g_s(s')=s^\bullet} \mu^\bullet_{\tau-1}(s^\bullet) \cdot \mathbf{P}(s,s')$$

$$= \sum_{s^\bullet \in S^\bullet} \mu^\bullet_{\tau-1}(s^\bullet) \sum_{s'|g_s(s')=s^\bullet} \mathbf{P}(s,s')$$

then the first hypothesis yields:

$$= \sum_{s^\bullet \in S^\bullet} \mu^\bullet_{\tau-1}(s^\bullet)\mathbf{P}(f(s), s^\bullet)$$

This term is equal to $\mu^\bullet_\tau(f(s))$ thanks to the equation (2) applied to the Markov chain $\mathcal{C}^\bullet$.

c.q.f.d. $\lozenge\lozenge\lozenge$

The family of functions (g_s) assigns to each transition of $\mathcal{C}$ starting from s a transition of $\mathcal{C}^\bullet$ starting from $f(s)$. The first condition can be checked by straightforward examination of the probability transition matrices. The second condition still involves the mapping $\mu^\bullet$ but here there are only comparisons between its values. Thanks to proposition 1, it can be proved by exhibiting a coupling of $\mathcal{C}^\bullet$ with itself.

Example *To apply the method on the example it remains to specify the family of functions $(g_s)_{s\in S}$.*

$$\begin{aligned}
g_{(n_1,n_2)}(n_1, n_2) &= f(n_1, n_2) \\
g_{(n_1,n_2)}(n_1 + 1, n_2) &= f(n_1 + 1, n_2) \\
g_{(n_1,n_2)}(n_1 - 1, n_2 + 1) &= f(n_1 - 1, n_2 + 1) \\
g_{(n_1,n_2)}(n_1, n_2 - 1) &= \begin{cases} (n_1, n_2 - 1) & \text{if } n_2 \leq R \\ (n_1 + n_2 - R, R - 1) & \text{else} \end{cases}
\end{aligned}$$

The condition 2 always trivially holds except for the last case with $n_2 > R$. We have to check that $\mu^\bullet(n_1+n_2-1-R, R) \leq \mu^\bullet(n_1+n_2-R, R-1)$. As $(n_1+n_2-R, R-1), (n_1+n_2-1-R, R))$ belongs to the coupling relation the inequality holds.

C. An Importance Sampling Method with Variance Reduction and Confidence Interval

Based on the previous developments, we describe a methodology to perform statistical model checking using importance sampling to estimate the tiny probability $p = \mu_u(s_0)$ in four steps.

1) Exhibit a suitable reduced Markov chain $\mathcal{C}^\bullet$ (by a function f).
2) Specify a coupling satisfying the required properties in order to insure $\mathcal{C}^\bullet$ is a reduction with guaranteed variance (Proposition 5).
3) Compute the distributions $\{\mu^\bullet_\tau\}_{0<\tau\leq u}$ (numerical computations using equations III-A on $\mathcal{C}^\bullet$).
4) Use these distributions to perform importance sampling on the simulation of the initial model. We generate a large sample of trajectories using the transition system corresponding to matrix P'_u (definition 7) and compute along each path the likehood L in order to obtain an estimation of p with some confidence interval.

The first step requires some understanding of the system to design the appropriate reduced chain. The proof of coupling is done by hand but could be mechanized with a proof assistant. We now describe in detail the third of forth steps since they rise algorithmic problems.

We denote by n the number of states of the Markov chain $\mathcal{C}$ and by d the maximum of outdegrees of vertices of $\mathcal{C}$. Let us remark that in typical modellings, d is very small compared to n. A simulation takes at most u steps going through states $(s_u, u), \ldots, (s_1, 1), s_\pm$ where $s_u = s_0$ and $s_\pm \in \{s_+, s_-\}$. In state (s_τ, τ), we compute the distribution $P'_u((s_\tau, \tau), -)$ (cf. definition 7), which requires the values of $\mu^\bullet_\tau(f(s))$ and $\mu^\bullet_{\tau-1}(f(s'))$, for each possible target state s' from s_τ.

Algorithm 1:

`Precomputation`$(u, \mu^\bullet_0, P^\bullet_0)$
Result: L
`// List` L `fulfills` $L(i) = \mu^\bullet_i$
$v \leftarrow \mu^\bullet_0$
for $i = 1$ **to** u **do**
  $v \leftarrow P^\bullet_0 v$
  $L(i) \leftarrow v$

Thanks to equations III-A, the vectors $\{\mu^\bullet_\tau\}_{0<\tau\leq u}$ may be computed iteratively one from the other with complexity $\Theta(ndu)$. More precisely, we derive from $\mathbf{P}^\bullet$, matrix $\mathbf{P}^\bullet_0$, a square (substochastic) matrix, indexed by $S_{a\bar{b}} \cup s_+$ and defined by $\forall s, s' \in S_{a\bar{b}}$:
$\mathbf{P}^\bullet_0(s, s') = \mathbf{P}^\bullet(s, s'), \mathbf{P}^\bullet_0(s, s_+) = \sum_{s'' \in S_b} \mathbf{P}^\bullet(s, s'')$
$\mathbf{P}^\bullet_0(s_+, s_+) = 1, \mathbf{P}^\bullet_0(s_+, s') = 0$
Then $\mu^\bullet_\tau = \mathbf{P}^\bullet_0 \cdot \mu^\bullet_{\tau-1}$ and $\mu^\bullet_0$ is null except $\mu^\bullet_0(s_+) = 1$. But for large values of u, the space complexity to store them becomes intractable and the challenge is to obtain a space-time trade-off. So we propose three methods. The methods consist of a precomputation stage and a simulation stage. Their difference lies in the information stored during the first stage and the additional numerical computations during the second stage. In the precomputation, both methods compute iteratively the u vectors $\mu^\bullet_\tau = (P^\bullet_0)^\tau(\mu^\bullet_0)$ for τ from 1 to u.

1) The first method is the "natural" implementation. It consists in storing all these vectors during the precomputation stage and then proceeding to the simulation without any additional numerical computations. The precomputation stage is described in algorithm 1 where list L is the main memory requirement.
2) Let $l(< u)$ be an integer. In the precomputation stage, the second method only stores the $\lfloor \frac{u}{l} \rfloor + 1$ vectors $\mu^\bullet_\tau$ with τ multiple of l in list L and $\mu^\bullet_{l\lfloor \frac{u}{l} \rfloor+1}, \ldots, \mu^\bullet_u$ in list K (see the precomputation stage of algorithm 2). During the simulation stage, in a state (s, τ), with $\tau = ml$, the vector $\mu^\bullet_{\tau-1}$ is present neither in L nor in K. So the method uses the vector $\mu^\bullet_{l(m-1)}$ stored in L to compute iteratively all vectors $\mu^\bullet_{l(m-1)+i} = P^{\bullet i}(\mu^\bullet_{l(m-1)})$ for i from 1 to $l-1$ and store them in K (see the step computation stage of algorithm 2). Then it proceeds to l consecutive steps of simulation without anymore computations. We choose l close to $\sqrt{u}$ in order to minimize the space complexity of such a factorization of steps.
3) Let $k = \lfloor \log_2(u) \rfloor + 1$. In the precomputation stage, the third method only stores $k+1$ vectors in L. More precisely, initially using the binary decomposition of u ($u = \sum_{i=0}^k a_{u,i} 2^i$), the list L of $k+1$ vectors consists of $v_{i,\tau} = \mu^\bullet_{\sum_{j=i}^k a_{\tau,j} 2^j}$, for all $1 \leq i \leq k+1$ (see the precomputation step of algorithm 3). During the simulation stage in a state (s, τ), with the binary decomposition of τ ($\tau = \sum_{i=0}^k a_{\tau,i} 2^i$), the list L consists of $v_{i,\tau} = \mu^\bullet_{\sum_{j=i}^k a_{\tau,j} 2^j}$, for all $1 \leq i \leq k+1$. Observe that the first vector $v_{1,\tau}$ is equal to $\mu^\bullet_\tau$. We obtain $\mu^\bullet_{\tau-1}$ by updating L according to $\tau - 1$. Let us describe the updating of the list performed by the stepcomputation of algorithm 3. Let i_0 be the smallest index such that $a_{\tau,i_0} = 1$. Then for $i > i_0$, $a_{\tau-1,i} = a_{\tau,i}$, $a_{\tau-1,i_0} = 0$ and for $i < i_0$, $a_{\tau-1,i} = 1$. The new list L is then obtained as follows. For $i > i_0$ $v_{i,\tau-1} = v_{i,\tau}$, $v_{i_0,\tau-1} = v_{i_0-1,\tau}$. Then the vectors for $i_0 < i$, the vectors $v_{i,\tau-1}$ are stored along iterated $2^{i_0-1} - 1$ matrix-vector products starting from vector $v_{i_0,\tau-1}$: $v(j, \tau-1) = P_0^{\bullet 2^j} v(j+1, \tau-1)$.

The computation associated with τ requires $1+2+\cdots+2^{i_0-1}$ products matrix-vector , i.e. $\Theta(nd2^{i_0})$. Noting that the bit i is reset at most $u2^{-i}$ times, the complexity of the whole computation is $\sum_{i=1}^{k} 2^{k-i}\Theta(nd2^i) = \Theta(ndu\log(u))$.

The three methods are numbered according to their decreasing space complexity. The corresponding space-time trade-off is summarized by table I, where the space unit is the storage of a float.

Algorithm 2:

`Precomputation`$(u, \mu_0^\bullet, P_0^\bullet)$
Result: L, K
`// List` L `fulfills` $L(i) = \mu_{i\cdot l}^\bullet$
$l \leftarrow \lfloor\sqrt{u}\rfloor$
$v \leftarrow \mu_0^\bullet$
for i **from** 1 **to** $\lfloor\frac{u}{l}\rfloor l$ **do**
  $v \leftarrow P_0^\bullet v$
  if $i \mod l = 0$ **then**
    $L(\frac{i}{l}) \leftarrow v$
`// List` K `contains` $\mu^\bullet_{\lfloor\frac{u}{l}\rfloor l+1}, \ldots, \mu_u^\bullet$
for i **from** $\lfloor\frac{u}{l}\rfloor l+1$ **to** u **do**
  $v \leftarrow P_0^\bullet v$
  $K(i \mod l) \leftarrow v$

`Stepcomputation`$(\tau, l, P_0^\bullet, K, L)$
`// Updates` K `when needed`
if $\tau \mod l = 0$ **then**
  $v \leftarrow L(\frac{\tau}{l} - 1)$
  for i **from** $(\frac{\tau}{l} - 1)l + 1$ **to** $\tau - 1$ **do**
    $v \leftarrow P_0^\bullet v$
    $K(i \mod l) \leftarrow v$

Algorithm 3:

`Precomputation`$(u, \mu_0^\bullet, P_0^\bullet)$
Result: L
`// L fulfills` $L(i) = \mu^\bullet_{\sum_{j=i}^{k} a_{u,j}2^j}$
$k \leftarrow \lfloor\log_2(u)\rfloor + 1$
$v \leftarrow \mu_0^\bullet$
$L(k+1) \leftarrow v$
for i **from** k **downto** 0 **do**
  if $a_{u,i} = 1$ **then**
    for j **from** 1 **to** 2^i **do**
      $v \leftarrow P_0^\bullet v$
  $L(i) \leftarrow v$

`Stepcomputation`$(\tau, l, P_0^\bullet, L)$
`// L is updated accordingly to` $\tau - 1$
$i_0 \leftarrow \min(i \mid a_{\tau,i} = 1)$
$v \leftarrow L(i_0 + 1)$
$L(i_0) \leftarrow v$
for i **from** $i_0 - 1$ **downto** 0 **do**
  for $j = 1$ **to** 2^i **do**
    $v \leftarrow P_0^\bullet v$
  $L(i) \leftarrow v$

Complexity	Method 1	Method 2	Method 3
Space	nu	$2n\sqrt{u}$	$n\log u$
Time for the precomputation	$\Theta(ndu)$	$\Theta(ndu)$	$\Theta(ndu)$
Additional time for the simulation	0	$\Theta(ndu)$	$\Theta(ndu\log(u))$

Table I
COMPARED COMPLEXITIES

More on simulation. Simulating trajectories sequentially is really inefficient since the additional computations during the simulation stage are repeated during every simulation. Thus for methods 2 and 3, we proceed with a bunch of trajectories simulated in parallel step by step. Different sizes of bunches are possible but they cannot exceed the size required for the numerical computations. So based on the asymptotic time and space cost, we handle n^2 trajectories in parallel.

D. Handling all the intervals

Three other kinds of intervals must be handled:

- When interval $I = [0, \infty]$, which is the topic that we fully describe in [5]. So we refer the interested reader to this communication.
- When interval $I = [l, u]$ with $l > 0$ that we detail below.
- When interval $I = [l, \infty]$ with $l > 0$ which, roughly speaking is managed by combining the two previous cases.

When $I = [l, u]$ with $0 < l < u$, chain $\mathcal{C}_I$ is defined by:

- Its state space is:
  $S_I = S_a \times [u-l+1, u] \cup S_{a\bar{b}} \times [u-l, 1] \cup \{s_-, s_+\}$
  with s_-, s_+ absorbing states.
- The transition probabilities associated with
  $S_{a\bar{b}} \times [u-l, 1]$ are the same ones as those of $\mathcal{C}_{u-l}$.
- $\forall s, s' \in S_a \ \forall \tau > u-l+1$
  $\mathbf{P}_u((s,\tau),(s',\tau-1)) = \mathbf{P}(s,s')$,
  $\mathbf{P}_u((s,\tau), s_-) = \sum_{s'\in S_{\bar{a}}} \mathbf{P}(s,s')$.
- $\forall s \in S_a \ \forall s' \in S_{a\bar{b}}$
  $Pt_u((s,u-l+1),(s',u-l)) = \mathbf{P}(s,s')$,
  $\mathbf{P}_u((s,u-l+1), s_-) = \sum_{s'\in S_{\bar{a}}} \mathbf{P}(s,s')$,
  $\mathbf{P}_u((s,u-l+1), s_+) = \sum_{s'\in S_b} \mathbf{P}(s,s')$.
- The other transition probabilities are null.

When $I = [u, u]$ with $0 < u$, chain $\mathcal{C}_I$ is defined by:

- Its state space is $S_I = S_a \times [1,u] \cup \{s_-, s_+\}$ with s_-, s_+ absorbing states
- $\forall s, s' \in S_a \ \forall \tau > 1$
  $\mathbf{P}_I((s,\tau),(s',\tau-1)) = \mathbf{P}(s,s')$,
  $\mathbf{P}_I((s,\tau), s_-) = \sum_{s' \in S_{\bar{a}}} \mathbf{P}(s,s')$.
- $\forall s \in S_a$,
  $\mathbf{P}_I((s,1), s_+) = \sum_{s' \in S_b} \mathbf{P}(s,s')$
  $\mathbf{P}_I((s,1), s_-) = 1 - \mathbf{P}_I((s,1), s_+)$.
- The other transition probabilities are null.

In both cases, by construction, $\mu(s,u) = \mu_I(s)$ where $\mu(s,u)$ is a reachability property in $\mathcal{C}_I$ and $\mu_I(s)$ is the probability to satisfy $aU^I b$ in $\mathcal{C}$.

Furthermore definition 7 can be adapted straightforwardly to obtain an importance sampling distribution P'_I yielding the same conclusions as those of proposition 4. Below we only detail the main equation related to P'_I. with $s, s' \in S_a$ and $u - l + 1 < \tau \leq u$. Given $I = [l,u]$ we denote $I - k = [\max(l-k, 0), u-k]$. Then:

$$\mathbf{P}'_I((s,\tau),(s',\tau-1)) = \frac{\mu^\bullet_{I-(\tau-1-u)}(f(s'))}{\mu^\bullet_{I-(\tau-u)}(f(s))} \mathbf{P}(s,s')$$

It remains to explain how to numerically compute $\mu^\bullet_{I-k}$ for $0 \leq k < u$. For sake of readability, we only present the case $0 < l < u$ (the case $0 < l = u$ is simpler). This is performed by iterative matrix-vector products using three transformations of $\mathbf{P}^\bullet$. Let $I' = [l', u']$ be an interval.

- If $l' = 0$ then one only needs to compute $\mu^\bullet_{I'}(s)$ for $s \in S_{a\bar{b}}$. So we use matrix $\mathbf{P}^\bullet_0$ defined in subsection III-C and $\mu^\bullet_{I'} = \mathbf{P}^\bullet_0 \cdot \mu^\bullet_{I'-1}$
- If $l' = 1$ then one needs to compute $\mu^\bullet_{I'}(s)$ for $s \in S_a$. So matrix $\mathbf{P}^\bullet_1$ is a (possibly non square) matrix defined for $S_a \times (S_{a\bar{b}} \cup s_+)$ by $\forall s \in S_a \ \forall s' \in S_{a\bar{b}}$:
  $\mathbf{P}^\bullet_0(s,s') = \mathbf{P}^\bullet(s,s'), \mathbf{P}^\bullet_0(s,s_+) = \sum_{s'' \in S_b} \mathbf{P}^\bullet(s,s'')$
  and $\mu^\bullet_{I'} = \mathbf{P}^\bullet_1 \cdot \mu^\bullet_{I'-1}$
- If $l' > 1$ then one needs to compute $\mu^\bullet_{I'}(s)$ for $s \in S_a$. So matrix $\mathbf{P}^\bullet_>$ is a square (substochastic) matrix defined for S_a by $\forall s, s' \in S_a$: $\mathbf{P}^\bullet_>(s,s') = \mathbf{P}^\bullet(s,s')$
  and $\mu^\bullet_{I'} = \mathbf{P}^\bullet_> \cdot \mu^\bullet_{I'-1}$

The three methods that we have proposed can be adapted for this general case. Since method 1 does not perform additional computations during the simulation, it only has to select the appropriate matrix during the iterative steps of the precomputation. Methods 2 and 3 still apply partial storage with additional computations by splitting the simulation in two steps whose boundary is time l.

IV. Experimentation

A. Implementation

Tools. Our experiments have been performed on a modified version of Cosmos. Cosmos is a statistical model checker whose input model is a stochastic Petri net with general distributions and formulas are expressed by the Hybrid Automata Stochastic Logic [6]. We have also used the model checker Prism for comparisons with our method. All the experiments have been performed on a computer with a 2.6Ghz processor and 48Go of memory which allow us to perform benchmark on very large model.

Adaptation of Cosmos. To proceed with experimentation on the bounded case we had to perform major changes in the tool. We detail below the added features to Cosmos we have implemented w.r.t. the handling of the unbounded "Until" case[2]:

- Cosmos takes as input continuous time stochastic Petri nets, therefore we adjusted the tool to apply on discrete time systems.
- To compute the vector of probability $\mu^\bullet_\tau$, we implemented a state space generator which computes matrix $P^\bullet$.
- We implemented the three algorithms described in section: III-C. Theses algorithms achieve different space-time trade-off for the computation of $\mu^\bullet_\tau$, which is needed for the importance sampling. To manage sparse matrix computation we use the uBLAS/boost libraries.
- We modified the simulator in order to construct step by step a batch of trajectories instead of generating full trajectories one after the other.

B. Example: Global Overflow in Tandem Queues

Modelling considerations. This example is a classical benchmark for importance sampling. It has practical interest as a standard modelling of networks [26]. Such a modelling allows to accurately dimension a network for a given loadwork. Furthermore it illustrates the advantage of dynamic importance sampling w.r.t. the static importance sampling [21].

We choose the parameters of the system as follows: $\mu = 0.8$ and $\rho_1 = \rho_2 = 0.1$. These parameters correspond to an unstable system. Thus checking the property "What is the probability for the system to overflow (more than N clients in both queues) before returning to an empty state within u steps" can be interpreted as the probability of a network to fall in u seconds while facing a deny of service attack.

Preliminary observation. While the probability associated with the time-bounded until converges towards the probability associated with the unbounded until when $u \leftarrow \infty$, the rate of this convergence highly depends on the model. In our benchmarks, these probabilities are drastically different: the former probability is tiny while the latter is close to 1.

Analysis of numerical and statistical Prism. We compare our method to numerical and statistical model checking done by Prism. The Prism statistical model checker behaves as follows. For $N = 500$, the computation takes at least $85s$ to ensure the confidence interval that we obtain in at most $5s$ with Cosmos. For larger values of N, the event becomes

[2]The reader can refer to [5] for details about the modifying distribution performed by the importance sampling method

Table II
EXPERIMENTAL RESULTS FOR THE EXAMPLE

N	R	u	Size of $\mathcal{C}$	numerical PRISM			Cosmos											
									Method 1			Method 2			Method 3			
				T (s)	Mem	$\mu(s_0)$	$\mu(s_0)$	Conf. Int.	T_{pre}	T_{sim}	Mem	T_{pre}	T_{sim}	Mem	T_{pre}	T_{sim}	Mem
500	5	650	$\approx$ 250E3	5	5.4M	0.0105	0.0104	0.00129	≈ 0	2	18M	≈ 0	3	4.3M	1	4	3.5M
1E3	10	1300	$\approx$ 1E6	36	18.8M	1.924E-4	1.989E-4	1.523E-05	4	5	116M	3	9	10 M	3	24	6M
5E3	20	6500	$\approx$ 25E6	3439	426M	1.795E-18	1.797E-18	2.502E-19	178	200	5345M	185	385	157M	114	1165	38M
10E3	25	13E3	$\approx$ 100E6	28028	1.7G	3.752E-36	3.770E-36	7.072E-37	883	908	26G	919	1894	521M	544	5911	93M
20E3	30	26E3	$\approx$ 400E6	227812	6.7G	1.246E-71	1.219E-71	3.351E-72	-	-	$\approx$127G	4347	4366	1696M	2703	32393	225M
50E3	35	65E3	$\approx$ 2.5E9	-	-	-	2.73E-178	-	-	-	-	33961	32489	7.5G	16515	248882	672M

very rare and the PRISM statistical model checker is unable to evaluate the probability.

We collect our results with respective time and space consumption for the three algorithms and PRISM in table II. We observe that the empirical complexity of time and memory consumption of numerical model checker PRISM are respectively $\Theta(uN^2)$ and $\Theta(N^2)$. For $N = 20000$, PRISM is highly time-consuming ($> 48h$). We did not launch the computation for $N = 50000$ since, applying extrapolation, we estimate its execution time to be bigger than a month.

Comparison of the three methods. Given some fixed horizon, the time complexity of the numerical computation of the methods is linear w.r.t. the size of the system $\mathcal{C}^\bullet$ (here $\Theta(NR)$). For our methods to be efficient we need to choose a value of R small compared to N while being enough large to provide tight confidence interval. On this example, very small value of R (smaller than 35) are sufficient to provide a tight confidence interval with a thousand of simulations. The confidence interval width is always less than 30% of the estimated value which is very accurate for statistical estimation of a rare event.

As the results and confidence interval width are similar for the three algorithm they are reported in the table only once. For the three algorithms we show the precomputation time, the simulation time (time unit is the second) and the memory consumption.

We observe that empirical complexities of time and memory of the three methods follow our theoretical ones, obtained for the asymptotic case (table I).

Applying method 1 is possible until N reach 10000, but for $N = 20000$ the program crashes because of its memory consumption. We have estimated the required memory in this case to be approximately equal to $127Go$ which is to big for a present-day computer. For such a N, methods 2 and 3 terminate, only using respectively $521Mo$ and $93Mo$. Moreover the method 2 and 3 are always faster than PRISM and use less memory. For $N = 20000$ the method 2 is 28 times faster and uses 4 times less memory while method 3 is 6 times faster and uses 29 times less memory.

From our experimentations, we recommend to apply method 1 when time and space resources are available. Otherwise, if memory is the bottleneck methods 2 and 3 should be preferred. Finally observe that for a huge horizon, it is likely that the probability associated with an "unbounded until" could be a good approximation of the time-bounded case. Since the method we design in this case requires significantly less memory, applying it is a good alternative when other methods fail.

V. CONCLUSION

We proposed a method of statistical model checking in order to compute with accuracy a tiny probability associated with a "time-bounded until" formula in a DTMC. We obtain a reliable confidence interval bounding this value. We have developed a theoretical framework justifying the validity of a confidence interval and ensuring the reduction of the variance. This framework requires a structural analysis of the model using coupling theory in order to get an appropriate distribution for the importance sampling method. As the time dependance of a property specified by "time-bounded until" formula induces a space explosion, we propose three different trade-off between space and time complexities. We have implemented these three algorithms in the statistical model checker COSMOS and we have done experiments on a classical relevant model with impressive results.

We plan to go further in three directions. Our first goal is to deal with more complex infinite systems. Secondly we want to handle "bounded until" formulas in the context of CTMC. In the long run, we consider to mechanize the proofs of coupling using an assistant prover as COQ [27], since it consists in checking parametrized inequalities.

REFERENCES

[1] E. A. Emerson and E. M. Clarke, "Characterizing correctness properties of parallel programs using fixpoints," in *ICALP*, ser. LNCS 85, 1980.

[2] R. Alur and T. A. Henzinger, "Logics and models of real time: A survey," *Real Time: Theory in Practice*, vol. 600, pp. 74–106, 1992.

[3] M. Kwiatkowska, G. Norman, and D. Parker, "Stochastic model checking," in *SFM'07*, ser. LNCS), M. Bernardo and J. Hillston, Eds., vol. 4486. Springer, 2007, pp. 220–270.

[4] A. Legay, B. Delahaye, and S. Bensalem, "Statistical model checking: an overview," in *RV'10*. Springer-Verlag, 2010, pp. 122–135.

[5] B. Barbot, S. Haddad, and C. Picaronny, "Coupling and importance sampling for statistical model checking," in *(TACAS'12)*, ser. LNCS, C. Flanagan and B. König, Eds. Tallinn, Estonia: Springer, Mar. 2012, to appear.

[6] P. Ballarini, D. Hilal, M. Duflot, S. Haddad, and N. Pekergin, "HASL: An expressive language for statistical verification of stochastic models," in *(VALUETOOLS'11)*, Cachan, France, May 2011, to appear.

[7] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen, "Model-checking algorithms for continuous-time markov chains," *IEEE Trans. Software Eng.*, vol. 29, no. 6, pp. 524–541, 2003.

[8] H. Hansson and B. Jonsson, "A logic for reasoning about time and reliability," *Formal Aspects of Computing*, vol. 6, pp. 102–111, 1994.

[9] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM: Probabilistic symbolic model checker," in *Computer Performance Evaluation: Modelling Techniques and Tools*, ser. LNCS. Springer Berlin / Heidelberg, 2002, vol. 2324, pp. 113–140.

[10] F. Ciesinski and C. Baier, "LiQuor: A tool for qualitative and quantitative linear time analysis of reactive systems." in *QEST'06*, 2006, pp. 131–132.

[11] J.-P. Katoen, I. S. Zapreev, E. M. Hahn, H. Hermanns, and D. N. Jansen, "The ins and outs of the probabilistic model checker MRMC," *Quantitative Evaluation of Systems, International Conference on*, pp. 167–176, 2009.

[12] E. G. Amparore and S. Donatelli, "Model checking CSL^{TA} with deterministic and stochastic petri nets," in *DSN*, 2010, pp. 605–614.

[13] L. J. Bain and M. Engelhardt, *Introduction to Probability and Mathematical Statistics, Second Edition*. Duxbury Classic Series, 1991.

[14] G. Chiola, G. Franceschinis, R. Gaeta, and M. Ribaudo, "GreatSPN 1.7: Graphical editor and analyzer for timed and stochastic Petri nets," *Perform. Eval.*, vol. 24, no. 1-2, pp. 47–68, 1995.

[15] J. Bengtsson, K. G. Larsen, F. Larsson, P. Pettersson, and W. Yi, "UPPAAL - a tool suite for automatic verification of real-time systems," in *Hybrid Systems*, 1995, pp. 232–243.

[16] K. Sen, M. Viswanathan, and G. Agha, "VESTA: A statistical model-checker and analyzer for probabilistic systems," *QEST*, vol. 0, pp. 251–252, 2005.

[17] H. Younes, "Ymer: A statistical model checker," in *Computer Aided Verification*, ser. LNCS, K. Etessami and S. Rajamani, Eds. Springer Berlin / Heidelberg, 2005, vol. 3576, pp. 171–179.

[18] G. Rubino and B. Tuffin, *Rare Event Simulation using Monte Carlo Methods*. Wiley, 2009.

[19] P. L'Ecuyer, V. Demers, and B. Tuffin, "Splitting for rare-event simulation," in *Winter Simulation Conference*, 2006, pp. 137–148.

[20] P. W. Glynn and D. L. Iglehart, "Importance sampling for stochastic simulations," *Management Science*, 1989.

[21] P.-T. de Boer, "Analysis of state-independent importance-sampling measures for the two-node tandem queue," *ACM Trans. Model. Comput. Simul.*, vol. 16, no. 3, pp. 225–250, 2006.

[22] R. Srinivasan, *Importance sampling – Applications in communications and detection*. Berlin: Sringer Verlag, 2002.

[23] P. Dupuis, A. D. Sezer, and H. Wang, "Dynamic importance sampling for queueing networks," *Annals of Applied Probability*, vol. 17, pp. 1306–1346, 2007.

[24] P. E. Heegaard and W. Sandmann, "Ant-based approach for determining the change of measure in importance sampling," in *Winter Simulation Conference*, 2007, pp. 412–420.

[25] T. Lindvall, *Lectures on the coupling method*. Dover, 2002.

[26] L. Kleinrock, *Queueing Systems*. Wiley Interscience, 1976, vol. II: Computer Applications, (Published in Russian, 1979. Published in Japanese, 1979.).

[27] Y. Bertot and P. Castéran, *Interactive Theorem Proving and Program Development. Coq'Art: The Calculus of Inductive Constructions*, ser. Texts in Theoretical Computer Science. Springer Verlag, 2004.