6-2019

Importance sampling of Interval Markov Chains

Cyrille JEGOUREL

Jingyi WANG

Jun SUN
Singapore Management University, junsun@smu.edu.sg

2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks

Importance Sampling of Interval Markov Chains

1st Cyrille Jegourel
Information Systems Technology and Design
Singapore University of Technology and Design
Singapore, Singapore
cyrille.jegourel@gmailcom

2nd Jingyi Wang
Information Systems Technology and Design
Singapore University of Technology and Design
Singapore, Singapore
wangjyee@gmail.com

3rd Jun Sun
Information Systems Technology and Design
Singapore University of Technology and Design
Singapore, Singapore
sunjunhqq@gmail.com

***Abstract*—In real-world systems, rare events often characterize critical situations like the probability that a system fails within some time bound and they are used to model some potentially harmful scenarios in dependability of safety-critical systems. Probabilistic Model Checking has been used to verify dependability properties in various types of systems but is limited by the state space explosion problem. An alternative is the recourse to Statistical Model Checking (SMC) that relies on Monte Carlo simulations and provides estimates within predefined error and confidence bounds. However, rare properties require a large number of simulations before occurring at least once. To tackle the problem, Importance Sampling, a rare event simulation technique, has been proposed in SMC for different types of probabilistic systems. Importance Sampling requires the full knowledge of probabilistic measure of the system, e.g. Markov chains. In practice, however, we often have models with some uncertainty, e.g., Interval Markov Chains. In this work, we propose a method to apply importance sampling to Interval Markov Chains. We show promising results in applying our method to multiple case studies.**

***Index Terms*—Rare Events, Importance Sampling, Markov Chains, Interval Markov Chains, Dependability, Statistical Model Checking**

I. Introduction

Discrete Time Markov Chains (DTMC) are a standard formalism to model and reason about probabilistic systems [9], [27], well suited to dependability analysis of security protocols (e.g. [20]) or safety-critical systems. In particular, the reliability of a failure-repair process can be described by a Markovian structure based on stochastic failure and repair mechanisms of the system components and can be investigated by *reachability* or *mean time to failure* properties specified with an appropriate logic.

However, modelling real-world systems is a difficult task: individual probabilistic transitions are in general unknown or partially known and may be given with a margin of error. For this reason, many extension of Markov Chains have been proposed in the literature [16], [22], [30]. In particular, Interval Markov Chains (IMC) are a formalism in which the transition values of a DTMC are given within intervals. Algorithms for common implementation and consistency of IMC have been proposed [10]. In the original work [16], the IMC semantics allowed a transition to be taken with different values in their corresponding interval at each occurrence. In this work, we consider an alternative common interpretation for IMCs in which they represent all of the DTMCs such that the transition probabilities lie in their corresponding intervals. In this semantics, the transitions are fixed *once-and-for-all*[1].

Probabilistic Model Checking algorithms have been developed to analyse stochastic systems in the context of DTMCs (e.g. [12], [29]) and IMCs [3], [4] but they are limited by the state space explosion problem. This limitation prompted the development of simulation-based techniques like Statistical Model Checking (SMC) [29]. SMC requires the use of an executable model of the system and then estimates the probability of a property based on the simulations. One of the core ideas of SMC is to sample independent execution traces of the system and individually verify if they satisfy a property of interest. The probability that the system satisfies the property is estimated by the proportion of traces which satisfy the property. By modelling the executions of a system as a Bernoulli random variable, SMC provides rigorous bounds of the error of the estimator based on confidence intervals or Chernoff bounds [6]. Note that SMC is not limited to frequentist inference and may use alternative efficient techniques, such as Bayesian inference [15] and hypothesis testing [28], to decide with specified confidence whether the probability of a property exceeds a given threshold or not.

However, rare events pose a problem to SMC because they imply that a large number of simulations must be sampled in order to observe them. Hence SMC may still be computationally challenging. Several variance reduction techniques, such as Importance Splitting [13] and Importance Sampling (IS) [14], [23], have been applied to estimate rare dependable properties in Markov models. IS works by simulating a system under a weighted (IS) distribution that makes a property more likely to be seen. It then compensates the results by

[1]Note that the *once-and-for-all* semantics is not novel but, as far as we know, the terminology is recent. See for example [3].

2158-3927/18/$31.00 Â©2018 IEEE
DOI 10.1109/DSN.2018.00040

the weights, to estimate the probability under the original distribution. In order to perform IS and to evaluate the resulting estimator, it is necessary to know exactly the probability distribution of the original system. This limitation makes IS infeasible for probability estimation of an IMC since the probabilistic transitions are given in intervals.

The goal of this work is to overcome this problem by using an optimisation algorithm. Due to the potentially large number of observed transitions and the inherent number of constraints that must be fulfilled, standard numerical and statistical approaches fail to work. We thus propose a new algorithm which is shown to work effectively for IMC importance sampling (IMCIS). We implement our approach with a prototype tool and apply the algorithm to estimate rare dependable properties of failure-repair processes and a safety property of a secure water treatment system. The experiment results show empirically that our confidence intervals are correct with respect to the original system instead of an approximation of the system.

a) Structure of the article: Section II introduces the basic notions of DTMCs, IMCs and Monte Carlo integration. Section III introduces the IS framework for IMCs. Section IV addresses the optimisation problem raised by IMCIS. Our algorithm is fully described in Section V, with the results of applying it to some case studies given in Section VI. Section VII concludes the paper.

II. Background

In this section, we introduce the notions and notations used throughout the paper. A stochastic system $\mathcal{S}$ is interpreted as a set of interacting components in which the state is determined randomly with respect to a global probability distribution. Let $(\Omega, \mathcal{F}, \mu)$ be the probability space induced by the system with Ω a set of finite paths with respect to system's property ϕ, $\mathcal{F}$ a σ-algebra of Ω and μ the probability distribution defined over $\mathcal{F}$. We first recall the definitions of a Discrete Time Markov Chain (DTMC) and an Interval Markov Chains (IMC) and give the basics of Monte Carlo integration.

A. Discrete Time and Interval Markov Chains

DTMCs are a standard formalism, extensively used in the literature, to model probabilistic systems. Formally,

Definition 2.1: A DTMC is a tuple $\mathcal{M} = (S, s_0, A, G, V)$, where S is a finite set of states, $s_0 \in S$ is an initial state, G is a set of atomic propositions, $V : S \to 2^G$ is a labelling function and $A : S \times S \to [0, 1]$ is a probabilistic transition function such that $\forall s \in S$, $\sum_{t \in S} A(s, t) = 1$.

For convenience, we use a matrix notation for the transition function, that is $A = (a_{ij})_{0 \leq i,j \leq m}$ with $m + 1 = |S|$. Each a_{ij} corresponds to the probability to reach s_j from s_i in one step. We denote $a_i = (a_{i0}, \cdots, a_{im})$ the probabilistic state distribution from $s_i \in S$.

Given the transition matrix A of a DTMC, the probability of taking a path $\omega = \omega_0 \to \cdots \to \omega_l$ is defined by the product of the individual transition probabilities of the path, i.e., $P_A(\omega) = \prod_{i=1}^{l} A(\omega_{i-1}, \omega_i)$. The length of a path is denoted $|\omega|$ and is defined by its number of transitions.

For a given path $\omega \in \Omega$, we denote $n_{ij}(\omega)$ the number of times the transition from state s_i to state s_j occurred. Thus, we can write $P(\omega)$ as a product of the elements of A:

$$P_A(\omega) = \prod_{i=0}^{m} \prod_{j=0}^{m} a_{ij}^{n_{ij}(\omega)} \tag{1}$$

Note that $\sum_{i=0}^{m} \sum_{j=0}^{m} n_{ij}(\omega) = |\omega|$. Also, if $a_{ij} = 0$, $n_{ij}(\omega) = 0$ and then $a_{ij}^{n_{ij}(\omega)} = 1$.

Definition 2.2: An IMC is a tuple $\mathcal{M} = (S, s_0, A^-, A^+, G, V)$, where S, s_0, G and V are as for a DTMC and where the transition function is replaced by two functions $A^-, A^+ : S \times S \to [0, 1]$ such that (i) $A^- \leq A^+$, (ii) $\forall s \in S$, $\sum_{t \in S} A^-(s, t) \leq 1$ and (iii) $\forall s \in S$, $\sum_{t \in S} A^+(s, t) \geq 1$.

A^- and A^+ give respective lower and upper bounds on the transition probabilities. IMCs are then a natural extension of DTMCs since they allow us to specify intervals of possible probability transitions for each state of the Markov chain. We say that $B \in [A]$ if B is a DTMC that satisfies all the interval constraints of $[A]$ and that $b_i \in [a_i]$ if we restrict the DTMC and the IMC to state i.

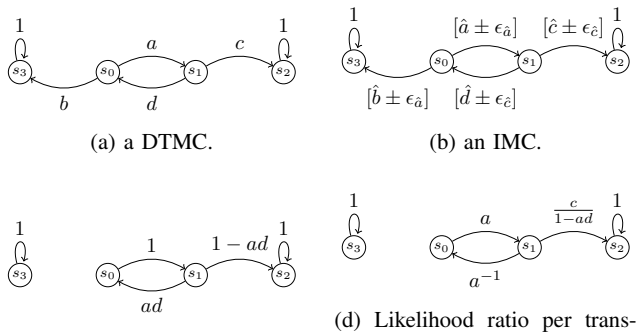
B. Learning a DTMC or an IMC

In practice, DTMCs are often obtained through some estimation based on belief, partial knowledge, learning process, etc. Therefore the transition probability is not precise. A common way to learn transition matrix A of Markov chain $\mathcal{M}$ is to use standard frequentist estimations based on a (long) sequence of random observations. An individual transition between state s_i and s_j can be estimated by $\hat{a}_{ij} = n_{ij}/n_i$ where n_{ij} is the number of times transition $s_i \to s_j$ occurred and n_i the number of times a transition has been taken from state s_i. However, this estimation lies within a confidence interval denoted I. For example, given confidence $1-\delta$ and n_i, one can determine absolute error ϵ such that $P\left(|\hat{a}_{ij} - a_{ij}| > \epsilon\right) \leq \delta$ using the Okamoto bound [21]. With $\delta = 10^{-5}$ and $n_i = 10^4$, $\epsilon \approx 0.025$ and $I = [\hat{a}_{ij} - \epsilon; \hat{a}_{ij} + \epsilon]$.

It is worth mentioning that if the state space is large, standard frequentist estimations are unlikely to be accurate for all transitions. But other methods have been proposed in the literature such as Laplace and Good-Turing's estimations [8], [11]. Moreover, large models are sometimes parametrised by global variables that may be learnt up to some precision. In the latter case, if the transitions are symbolic functions of the global variables, it is not necessary to observe all the transitions but to estimate directly the global variables and to deduce a DTMC or an IMC from it.

In this article, $\hat{A} = (\hat{a}_{ij})_{0 \leq i,j \leq m}$ denotes a learnt transition matrix of Markov chain $\mathcal{M}$. We assume that the DTMC is learnt up to some precision $\epsilon = (\epsilon_{ij})_{0 \leq i,j \leq m}$. Then, we denote $\hat{A}^- = \hat{A} - \epsilon$, $\hat{A}^+ = \hat{A} + \epsilon$ and $[\hat{A}]$ the corresponding IMC centred on DTMC $\hat{A}$. By construction, $\hat{A} \in [\hat{A}]$.

Fig. 1a illustrates a DTMC A with state space $S = \{s_0, \cdots, s_3\}$ and a probabilistic distribution μ parametrised by two individual transitions a and c. Fig. 1b illustrates an



(a) a DTMC. (b) an IMC.

(c) Perfect Importance Sampling. (d) Likelihood ratio per transition.

Figure 1: DTMC and IMC

IMC of A in which parameters a and c are supposed to be equal to $\hat{a}$ and $\hat{c}$ up to margins of error $\epsilon_{\hat{a}}$ and $\epsilon_{\hat{c}}$.

C. Monte Carlo estimation

Given a probabilistic model $\mathcal{S}$, a DTMC or an IMC, the goal is to estimate the probability that a random execution of $\mathcal{S}$ satisfies a property ϕ specified using bounded temporal logic formulae (see for example [12]). Let γ be this probability and z be the function that assigns 1 to a trace satisfying ϕ and 0 otherwise. By definition, γ is the sum of the probabilities of the paths ω such that $z(\omega) = 1$. In other words, γ is the expectation of function z over the set of traces Ω where z must be interpreted as a Bernoulli random variable Z:

$$\gamma = \sum_{\Omega} z(\omega) P_\mu(\omega) = \sum_{\omega \models \phi} P_\mu(\omega) = \mathbb{E}_\mu[Z], \tag{2}$$

with $P_\mu(\omega)$ the probability of path ω under the probability distribution μ.

In SMC, a set of N traces $(\omega_i)_{1 \leq i \leq N}$ is sampled randomly according to distribution μ and a Monte Carlo frequentist estimation $\hat{\gamma}_N$ of γ is given by:

$$\hat{\gamma}_N = \frac{1}{N} \sum_{i=1}^{N} z(\omega_i). \tag{3}$$

Note that $z(\omega_i)$ is effectively the realisation of a Bernoulli random variable with parameter γ. Hence $\mathrm{Var}(\hat{\gamma}_N) = \gamma(1-\gamma)/N$. Given the level of confidence $1-\delta$ and $\Phi^{-1}_{1-\delta/2}$ the $(1-\delta/2)$-quantile of the normal distribution, an approximate confidence interval is given by $I = \left[\hat{\gamma}_N \pm \Phi^{-1}_{1-\delta/2} \sqrt{\frac{\hat{\gamma}_N(1-\hat{\gamma}_N)}{N}}\right]$.

For clarity, we sometimes use the notation $\gamma(A)$ to refer to the probability of property ϕ when μ is a DTMC parametrised by the matrix of transitions A, $\hat{\gamma}_N(A)$ to denote an estimate of $\gamma(A)$ based on N samples and $\hat{\sigma}_N(A)$ the empirical standard deviation of the samples.

III. Importance Sampling in Markov Models

Given a set of N traces, the absolute error, defined as the half size of the confidence interval, decreases as the inverse square root of N. But for small probabilities, the accuracy of the estimation is better captured by the relative error, that is the absolute error divided by γ. However, the relative error explodes when γ tends to zero since it is inversely proportional to the square root of $N\gamma$. So, in practice, denoting the relative error RE, if we desired $RE = 10\%$, we would need to increase N as a proportion of $100 * \gamma^{-1}$. Rare events require too many samples to be observed at least once and prompted the recourse to advanced simulation techniques such as importance sampling [2], [7], [14], [23].

A. IS estimation

Let μ be absolutely continuous with respect to another probability measure μ' over Ω, then (2) can be written

$$\gamma = \sum_{\Omega} \frac{P_\mu(\omega)}{P_{\mu'}(\omega)} z(\omega)\, P_{\mu'}(\omega). \tag{4}$$

The function $L = P_\mu / P_{\mu'}$ is called the *likelihood ratio* function and γ can be then interpreted as the expectation of function z weighted by L under probabilistic measure μ':

$$\gamma = \sum_{\Omega} L(\omega) z(\omega) P_{\mu'}(\omega) = \mathbb{E}_{\mu'}[ZL]. \tag{5}$$

Note that in a DTMC, the likelihood ratio L of a path is the ratio of its probabilities under distribution μ and μ'. Assume that μ' is defined on the same space than μ and is parametrised by probability matrix $B = (b_{ij})_{0 \leq i,j \leq m}$. Then, $L(\omega) = P_A(\omega)/P_B(\omega)$.

In practice, the likelihood ratio of ω is initialised to 1 and, once a transition $s_i \to s_j$ is taken, the likelihood ratio is updated on-the-fly by multiplying its current value by a_{ij}/b_{ij}. Formally, we can write any likelihood ratio as a product of power of all the ratios a_{ij}/b_{ij}:

$$L(\omega) = \frac{\prod_{i=0}^{m} \prod_{j=0}^{m} a_{ij}^{n_{ij}(\omega)}}{\prod_{i=0}^{m} \prod_{j=0}^{m} b_{ij}^{n_{ij}(\omega)}} = \prod_{i=0}^{m} \prod_{j=0}^{m} \left(\frac{a_{ij}}{b_{ij}}\right)^{n_{ij}(\omega)} \tag{6}$$

We can thus estimate γ by sampling traces under μ' and compensating each path ω_k by its likelihood ratio $L(\omega_k)$:

$$\hat{\gamma}_{N_{\mathrm{IS}}} = \frac{1}{N_{\mathrm{IS}}} \sum_{k=1}^{N_{\mathrm{IS}}} L(\omega_k) z(\omega_k) \tag{7}$$

Here $\omega_k \sim \mu'$ and N_{IS} denotes the number of simulations used by the IS estimator. An approximate confidence interval is given by $I = \left[\hat{\gamma}_{N_{\mathrm{IS}}} \pm \Phi^{-1}_{1-\delta/2} \frac{\hat{\sigma}_{N_{\mathrm{IS}}}}{\sqrt{N_{\mathrm{IS}}}}\right]$ where $\hat{\sigma}_{N_{\mathrm{IS}}}$ denotes the empirical standard deviation of the samples. The goal of IS is to reduce the variance of the rare event and so achieve a narrower confidence interval than the Monte Carlo estimator, resulting in $N_{\mathrm{IS}} \ll N$. In general, the IS distribution μ' is chosen to produce the rare events more frequently.

The IS distribution defined by $P_{\mu'} = zP_\mu/\gamma$ outputs an estimator with zero variance. Indeed, the paths that do not satisfy ϕ have a probability 0 to occur and the likelihood ratio of the successful paths is equal to γ. Sampling under this *perfect* distribution is however unrealistic since it requires to know γ which is the probability to estimate.

In practice, choosing a good importance sampling distribution in terms of variance reduction is a conundrum.

Nevertheless, in the framework of DTMCs, the cross-entropy algorithm can be used to find a good candidate for the IS distribution as shown in [14], [24].

B. Margin of error problem

In the following, we show that applying existing important sampling techniques to a DTMC learnt from a real system would result in general in significant errors. Let us consider DTMC A described in Fig. 1a. Assume that the initial state is s_0 and that our goal is to estimate the probability γ of reaching s_2. Remark that, in this simple example, $\gamma = ac/(1-ad)$. Thus, with $a = 0.0001$ and $c = 0.05$, $\gamma \approx 5.005 \times 10^{-6}$. An example of perfect IS distribution for A, called B, is given in Fig. 1c. Fig. 1d gives the ratios (a_{ij}/b_{ij}). Note that all the paths sampled with respect to B are successful and have the same likelihood ratio $L(\omega) = ac/(1-ad) = \gamma$. It follows that $V(\hat{\gamma}_N) = 0$ and, independently to confidence level $(1-\delta)$ and sample size N, the confidence interval is reduced to a single point : $I = [\hat{\gamma}_N \pm 0] = \gamma$.

Assume now that A is unknown and approximated by a transition matrix $\hat{A}$ parametrised by $\hat{a}$ and $\hat{c}$. The graph structure being identical, it is easy to find the perfect important sampling with respect to $\hat{A}$ and eventually to output $\hat{\gamma}_N(\hat{A}) = \frac{\hat{a}\hat{c}}{1-\hat{a}\hat{d}}$ and a confidence interval reduced to this point. Unfortunately, this estimation is only perfect with respect to $\hat{A}$ regardless of how close $\hat{A}$ and A are. It is extremely unlikely that $\hat{\gamma}_N(\hat{A}) = \gamma$ and consequently, the confidence interval almost surely never contains the exact probability. More importantly, a slight error of approximation of the probabilistic transitions may lead to significant different results. For example, with $\hat{a} = 0.0003$ and $\hat{c} = 0.0498$, $\hat{\gamma}_N(\hat{A}) = 1.4944 \times 10^{-5}$, which is almost three times the exact value.

Since IS implies the computation of a potentially large product of individual transition probabilities, a fine understanding of the system behaviour is necessary to be performed correctly. If the abstraction of the system is too coarse, it is unlikely the case. Even a low deviation of one particular individual transition may have large consequences on the final computation. The sensitivity of the results seriously poses the question of the validity of IS for approximated models of real-world systems, which in general are much larger and more complex than this example. This motivates us to take into account the margin of errors in our IS analysis.

C. IS for IMC

In the following, we show how to apply IS to IMC by reducing the problem to an optimization problem. For any A in an IMC $[\hat{A}]$, the exact probability $P_A(\omega)$ of a path ω falls within the following interval:

$$\prod_{i=0}^{m}\prod_{j=0}^{m}(a_{ij}^{-})^{n_{ij}(\omega)} \leq P_A(\omega) \leq \prod_{i=0}^{m}\prod_{j=0}^{m}(a_{ij}^{+})^{n_{ij}(\omega)} \qquad (8)$$

Then, given a sample of N paths and an IS distribution B, for all $A \in [\hat{A}]$,

$$\frac{1}{N}\sum_{k=1}^{N} z(\omega_k)\prod_{i=0}^{m}\prod_{j=0}^{m}\left(\frac{a_{ij}^{-}}{b_{ij}}\right)^{n_{ij}(\omega_k)} \leq \hat{\gamma}_N(A) \leq \frac{1}{N}\sum_{k=1}^{N} z(\omega_k)\prod_{i=0}^{m}\prod_{j=0}^{m}\left(\frac{a_{ij}^{+}}{b_{ij}}\right)^{n_{ij}(\omega_k)}$$

But, optimising individually each transition leads to very coarse bounds. For all i, individual probabilities $(a_{ij})_{0\leq i,j\leq m}$ must fulfil the vectorial constraint: $\sum_{j=0}^{m} a_{ij} = 1$. Moreover, a transition observed in different paths may optimise each path probability in a different way. Since we use the *once-for-all* IMC semantics, improving the bounds requires to optimise the transitions all together. For this purpose, we first rewrite the lower bound problem as a constrained minimisation problem:

$$\begin{aligned} \underset{A\in[\hat{A}]}{\text{minimize}} \quad & \sum_{k=1}^{N} z(\omega_k)\prod_{i=0}^{m}\prod_{j=0}^{m}\left(\frac{a_{ij}}{b_{ij}}\right)^{n_{ij}(\omega_k)} \\ \text{subject to} \quad & a_{ij}^{-} - a_{ij} = c^{-}(a_{ij}) \leq 0, \text{ for all } j, \\ \text{and} \quad & a_{ij} - a_{ij}^{+} = c^{+}(a_{ij}) \leq 0, \text{ for all } j, \\ \text{and} \quad & 1 - \sum_{j=0}^{m} a_{ij} = c(a_i) = 0, \text{ for all } i. \end{aligned} \qquad (9)$$

The upper bound can be handled similarly by rewriting it as a maximisation problem. We denote $A_{\min}$ and respectively $A_{\max}$ the DTMCs that minimises and maximises the optimisation problem. In what follows, for convenience, we only present our approach for the lower bound.

The minimisation problem can be simplified. First of all, it is worth noting that the probabilistic distributions from a given state are independent of each other. In other words, optimising state distribution a_i has no impact on a_j. Moreover, it is only necessary to optimise the distribution from state s if at least one transition $s \to s'$ is observed in a successful path with respect to property ϕ. Denoting α the set of indexes of successful paths, M its cardinal, $\alpha(k)$ the k-th element of α with $1 \leq k \leq M$ and T_k the set of transitions observed in $\omega_{\alpha(k)}$:

$$\begin{aligned} \underset{A\in[\hat{A}]}{\text{minimize}} \quad & \sum_{k=1}^{M}\prod_{(i\to j)\in T_k}\left(\frac{a_{ij}}{b_{ij}}\right)^{n_{ij}(\omega_{\alpha(k)})} \\ \text{subject to} \quad & a_{ij}^{-} - a_{ij} = c^{-}(a_{ij}) \leq 0, \text{ for all } j, \\ \text{and} \quad & a_{ij} - a_{ij}^{+} = c^{+}(a_{ij}) \leq 0, \text{ for all } j, \\ \text{and} \quad & 1 - \sum_{j=0}^{m} a_{ij} = c(a_i) = 0, \text{ for all } i. \end{aligned} \qquad (10)$$

In what follows, we denote $f(A)$ the objective function to minimise.

In some cases, it remains easy to evaluate a_{ij}. For example, if only one transition $s_i \to s_j$ has been taken from state s_i, then $a_{ij} = \max(a_{ij}^{-}, 1 - \sum_{j'\neq j}^{m} a_{ij'}^{+})$. This expression guarantees that a_{ij} is well defined and remains consistent

with regard to the constraints that apply to the other outgoing transitions from state s_i. Nevertheless, if several transitions from state s_i have been observed, the problem becomes much harder and requires the recourse to a minimisation algorithm.

IV. Solving the minimisation problem

Minimising $f(A)$ requires a class of algorithms for solving optimisation problems with equality and inequality constraints. Numerical methods like penalty or interior point methods (see e.g. [19]) are not suitable in our context due to the large sample size and number of observed transitions. Various statistical methods, notably in convex optimisation, could be used but their efficiency is strongly impacted by the large number of constraints. Which algorithms fit the best our problem is a complex question as the answer is likely system-dependent; providing a clear answer goes beyond this article. For more details, we discuss in the appendix alternative promising methods, notably the stochastic gradient descent [18] and the stochastic interior Point Method [5]. In this work, we propose a simple algorithm which is proper for solving our problem and converges almost surely to a global minimum. Note that the global minimum is not necessarily unique and is guaranteed to exist since the intervals for each parameter are closed and the objective function is continuous.

A. Monte Carlo Random Search algorithm

We propose to determine a global minimum by a random search into the domain of definition $[\hat{A}]$ of the function. The algorithm works as follows: starting with $A_{\min} = A^{(0)} = \hat{A}$, we sample iteratively independent candidates $A^{(l+1)}$ in $[\hat{A}]$ according to a probability distribution X covering all the DTMCs in $[\hat{A}]$. If $f(A^{(l+1)}) < f(A^{(l)})$, $A^{(l+1)}$ becomes the new minimum $A_{\min}$, otherwise the minimum remains unchanged. When a candidate is undefeated for R rounds, we stop the search and outputs $A_{\min}$ as an approximation of the minimum. We can not prove that the minimum has been reached but at least, the probability that the minimum is below $f(A_{\min})$ is less than R^{-1}.

The method is known to be convergent (see for example Theorem 2.1, page 40 in [26]) but in general the speed of convergence is low. Nevertheless, in practice, termination can be ensured by setting a maximal number of samples. Moreover, the algorithm remains easy to set up since it does not require any gradient or Hessian matrix computation.

The main difficulty is to generate DTMCs satisfying all the constraints of $[\hat{A}]$. Indeed, sampling uniformly in each interval would pose some consistency problem and would likely violate the equality constraints. Moreover, after normalisation, the interval constraints would unlikely be satisfied. Finally, we have to guarantee that the whole consistent domain of definition can be covered by the samples. Assume in what follows that we want to generate the candidate $A^{(l)}$.

B. Dirichlet distributions

Dirichlet distributions are useful when one wants to cut a string (of length 1) into $m+1$ pieces X_j of different lengths where each piece has a specific average length parametrised by α_j, but allowing some variation in the relative sizes of the pieces. We denote $\beta = \sum_{j=0}^{m} \alpha_j$. For all j, the relative average length of X_j is $\mathbb{E}_{\text{Rel}}[X_j] = \alpha_j/\beta$ and the variance of the relative length varies inversely with β: $\mathbb{V}_{\text{Rel}}(X_j) = \frac{\alpha_j(\beta-\alpha_j)}{\beta^2(\beta+1)}$.

Multiplying the random variable $X = (X_j)_{0\leq j\leq m}$ by a constant $K > 0$ does not impact the relative lengths because the length of each coordinate is multiplied by the same constant. However, the relative variances $\mathbb{V}_{\text{Rel}}(X_j)$ decreases to zero when K tends to the infinity:

$$\begin{aligned}\mathbb{V}_{\text{Rel}}(KX_j) &= \frac{K\alpha_j(K.\beta - K\alpha_j)}{K^2\beta^2(K\beta+1)} \\ &= \frac{\alpha_j(\beta-\alpha_j)}{\beta^2(K\beta+1)} \underset{K\to\infty}{\longrightarrow} 0\end{aligned}$$

Given an IMC $[\hat{A}]$, for each visited state s_i, we sample $m+1$ values denoted $(a_{ij})_{0\leq j\leq m}$ according to a Dirichlet distribution $X = (X_{ij})_{0\leq j\leq m}$ parametrised by vector $(K_i\hat{a}_{ij})_{0\leq j\leq m}$ where $K_i > 0$ is a precomputed parameter aiming to control the relative variances. If all the constraints of $[\hat{a}_{ij}]$ are satisfied, $(a_{ij})_{0\leq j\leq m}$ is the state distribution from s_i of the DTMC candidate $A^{(l)}$.

If K_i is chosen too large, the variance of each coordinate decreases and we would likely sample values that are too close to the mean $\hat{a}_{ij}$. If K_i is too low, the variance becomes larger and we would generate values that do not belong to $[\hat{a}_{ij} - \epsilon_{ij}, \hat{a}_{ij} + \epsilon_{ij}]$.

For this purpose, we set, for each transition $(i \to j)$, a value K_{ij} such that the standard deviation of X_{ij} equals ϵ_{ij}:

$$\epsilon_{ij} = \sqrt{\frac{\hat{a}_{ij}(1-\hat{a}_{ij})}{K_{ij}+1}}$$

Then,

$$K_{ij} = \frac{\hat{a}_{ij}(1-\hat{a}_{ij})}{\epsilon_{ij}^2} - 1$$

Finally, if the values K_{ij} have the same order of magnitude, we choose $K_i = \min_j K_{ij}$. We thus guarantee that the coordinate values of the candidate are well-spread around the mean while falling in their corresponding interval $[\hat{a}_{ij} \pm \epsilon_{ij}]$ with high probability since the standard deviations of X_{ij} are slightly greater or equal than the corresponding ϵ_{ij}.

C. Tuning the algorithm

If a generated vector does not fulfil the constraints, we simply discard it and generate a new one until all the constraints are satisfied. This may be however challenging if m is large or if K_{ij} have different orders of magnitude. We proposed two simple solutions to overcome these problems.

1) m is large: If the Dirichlet sampler fails to generate a DTMC candidate satisfying the $[\hat{a}_i]$ constraints in state i, a possibility is to multiply K_i by a value strictly greater than 1, for example, $\lambda = 1.1$. The goal is to smoothly reduce the variance of each coordinate while preserving their relative length, increasing the chance to sample all the coordinates in their respective intervals.

2) K_{ij} have different orders of magnitude: If K_{ij} have different orders of magnitude, choosing K_i as the minimum of K_{ij} may not be adequate since the relative variance of the corresponding coordinates may be too large. Consequently, the samples for these coordinates would likely fall out of their corresponding interval and the resulting state distribution a_i would not satisfy the constraints of $[\hat{a}_i]$. Choosing the mean or the median of K_{ij} may be more efficient though it does not fully overcome this problem.

An other solution is to handle separately the transitions with a 'large' K_{ij} and the ones with a significantly 'lower' K_{ij}. For sake of simplicity, assume that K_{i0} is large with respect to the other K_{ij} and that these K_{ij} have the same order of magnitude. We proceed in two steps: (i) We select uniformly a value a_{i0} in the interval $[\hat{a}_{i0} \pm \epsilon_{i0}] \cap [1 - \sum_{j=1}^{m} \hat{a}_{ij}^{+}; 1 - \sum_{j=1}^{m} \hat{a}_{ij}^{-}]$. The intersection guarantees the consistency of a_{i0}. Let $\beta = 1 - a_{i0}$. (ii) Once a_{i0} has been selected, the other transition values are sampled according to distribution $Y_i \sim \beta X(K_i(\hat{a}_{ij}))_{j' \neq j}$ where X is a Dirichlet distribution parametrised by $K_i(\hat{a}_{ij})$. Then, for all $j' \neq j$,

$$\mathbb{E}_{\text{Rel}}[Y_{ij'}] = \beta \frac{K_i \hat{a}_{ij'}}{\sum_{j' \neq j} K_i \hat{a}_{ij'}} = \hat{a}_{ij'} \tag{11}$$

and

$$\begin{aligned} \mathbb{V}_{\text{Rel}}(Y_{ij'}) &= \beta^2 \frac{K_i a_{ij'}(K_i\beta - K_i\hat{a}_{ij'})}{K_i^2\beta^2(K_i\beta + 1)} \\ &= \frac{\hat{a}_{ij'}(\beta - \hat{a}_{ij'})}{K_i\beta + 1} \end{aligned} \tag{12}$$

By choosing $K_i = \min_{j' \neq j} \frac{\beta - \hat{a}_{ij}}{\beta \epsilon_{ij'}^2} - \frac{1}{\beta}$, we thus ensure that $\mathbb{E}_{\text{Rel}}[Y_{ij'}] = \hat{a}_{ij'}$ and $\sqrt{\mathbb{V}_{\text{Rel}}(Y_{ij'})} \geq \epsilon_{ij'}$ for all $j > 0$. The procedure is thus repeated until all the values $a_{ij'}$ rely in their corresponding interval. Then, $(a_{ij})_{0 \leq j \leq m}$ is the state distribution in state i of candidate $A^{(l)}$.

V. Description of the Algorithm

We present in this section the pseudo-algorithm of importance sampling for IMC (Algorithm 1) and the pseudo-algorithm for the random search optimisation (Algorithm 2). For sake of simplicity, we have not included the cases 'm is large' and 'K_{ij} have different orders of magnitude' mentioned in Section IV-C.

The goal of the algorithm is, given an IMC $[\hat{A}]$, an IS distribution parametrised by B and the property ϕ, to output a $(1-\delta)$-confidence interval defined with respect to $[\hat{A}]$ instead of $\hat{A}$. The inputs of the algorithm are confidence parameter δ and sample size N used to estimate γ.

Remark 5.1: Generating matrices $A \in [\hat{A}]$ and solving the minimisation problem is independent of B. However, even if the topic of this paper is not about how B is chosen, it remains an interesting question to know if there exists a 'better' IS distribution defined with respect to the entire set of matrices in $[\hat{A}]$. In this work, we assume that the IS distribution is defined with respect to $\hat{A}$ but note that other distributions could have been chosen (for the better or the worst).

Traces are sampled from initial state s_0 with respect to probabilistic distribution B until ϕ is decided (Alg. 1, lines 3 to 5). Note that we do not need to store the entire trace. Instead, for each trace ω_k, we update on-the-fly a table containing the transitions $s_i \to s_j$ of ω_k and the number of times these transitions have been taken $n_k(s_i, s_j)$. This table is defined by the set of transitions T_k and their respective counters n_k in Algorithm 1 (lines 6 to 11). At line 13, notation $\mathbf{1}(\omega_k \models \phi)$ is the indicator function and is equal to 1 if $\omega_k \models \phi$ and 0 otherwise. We denote V_k the set of visited states in ω_k (apart the last state of the trace), V and T the respective union of V_k and T_k over all the traces. The symbolic likelihood ratio of ω_k is then entirely defined by the k-th table. If $\omega_k \not\models \phi$, the table can be deleted since $z(\omega_k)L(\omega_k) = 0$.

Once all the traces have been sampled, the tables and $[\hat{A}]$ define the minimisation problem described in (10). The function to optimise is denoted $f(A)$ at line 16 and we use Algorithm (2) for this purpose. At line 17, $g(A)$ denotes the sum of the likelihood ratio squares for the successful paths used in the evaluation of the standard deviations. Once the arguments $A_{\min}$ and $A_{\max}$ of the minimum and maximum have been determined, we evaluate $\hat{\gamma}_N(\hat{A}_{\min})$, $\hat{\sigma}_N(\hat{A}_{\min})$, $\hat{\gamma}_N(\hat{A}_{\max})$ and $\hat{\sigma}_N(\hat{A}_{\max})$ (lines 19 to 22). Finally, we output the final $(1-\delta)$-confidence interval $CI = [L, U]$ where:

$$L = \hat{\gamma}_N(\hat{A}_{\min}) - \Phi^{-1}_{1\delta/2} \frac{\hat{\sigma}_N(\hat{A}_{\min})}{\sqrt{N}}$$

$$U = \hat{\gamma}_N(\hat{A}_{\max}) + \Phi^{-1}_{1\delta/2} \frac{\hat{\sigma}_N(\hat{A}_{\max})}{\sqrt{N}}.$$

VI. Case study

In the following, we conduct multiple case studies to evaluate the efficiency of our algorithms. The challenge for our approach is to show that we are able to provide more reliable importance sampling confidence intervals with the IMC settings. Hence, we have chosen models for which we are able to obtain accurate results using numerical techniques, in order to compare them with the correct values.

The empirical coverage of the experiments is the proportion of experiments in which the exact value γ falls within the final confidence interval. To empirically verify our results we performed each simulation experiment 100 times and report the coverage of the experiments with respect to the approximated DTMC $\hat{A}$ and with the exact DTMC A. We use the same IS distribution for IS experiments and IMCIS experiments but they are performed independently. The estimators are based on $N = 10000$ traces. The optimisation is stopped when the randomly generated candidates for the minimum and the maximum are undefeated for $R = 1000$ rounds. All simulations were performed using a Java prototype.

A. Illustrative example

The first case study follows the example introduced in Section III. The model under scrutiny is a DTMC parametrised by two individual transitions $a = 10^{-4}$ and $c = 0.05$.

Algorithm 1 IMC Importance Sampling (IMCIS)

Input: $[\hat{A}]$: an IMC
B : an IS matrix
φ : a temporal property
δ : confidence parameter
N : sample size

for $k \in \{1, \ldots, N\}$ **do**
  $\omega_k = x_0$, $V_k = \emptyset$, $T_k = \emptyset$
  $l = 1$
  while $\omega_k \models \phi$ is not decided **do**
    generate s_l under IS measure B
    $\omega_k = s_0 \cdots s_l$
    $V_k = V_k \cup \{s_{l-1}\}$
    if $s_{l-1} \to s_l \notin T_k$ **then**
      $T_k = T_k \cup \{s_{l-1} \to s_l\}$ and $n_k(s_{l-1}, s_l) = 1$
    else
      $n_k(s_{l-1}, s_l) = n_k(s_{l-1}, s_l) + 1$
    end if
  end while
  $z(\omega_k) = \mathbf{1}(\omega_k \models \phi)$
end for
$V = \bigcup_{k=1}^{N} V_k$, $T = \bigcup_{k=1}^{N} T_k$
$f(A) = \sum_{k=1}^{N} z(\omega_k) \prod_{i \in V_k} \prod_{j|(i \to j) \in T_k} \left(\frac{a_{ij}}{b_{ij}}\right)^{n_k(s_i, s_j)}$
$g(A) = \sum_{k=1}^{N} z(\omega_k) \prod_{i \in V_k} \prod_{j|(i \to j) \in T_k} \left(\frac{a_{ij}}{b_{ij}}\right)^{2n_k(s_i, s_j)}$
OPTIMISATION of $f(A)$ (see Algorithm 2)
$\hat{\gamma}_N(A_{\min}) = f(A_{\min})/N$
$\hat{\gamma}_N(A_{\max}) = f(A_{\max})/N$
$\hat{\sigma}_N(A_{\min}) = g(A_{\min})/N - \hat{\gamma}_N(A_{\min})^2$
$\hat{\sigma}_N(A_{\max}) = g(A_{\max})/N - \hat{\gamma}_N(A_{\max})^2$

Output: $(1-\delta)$-confidence interval $CI = [L; U]$
where $L = \hat{\gamma}_N(A_{\min}) - \Phi^{-1}_{\delta/2} \hat{\sigma}_N(A_{\min})/\sqrt{N}$
and $U = \hat{\gamma}_N(A_{\max}) + \Phi^{-1}_{1-\delta/2} \hat{\sigma}_N(A_{\max})/\sqrt{N}$

Algorithm 2 Random Search Optimisation

Input: $[\hat{A}]$: an IMC
$f(A)$: function to optimise ($A \in [\hat{A}]$)
V : set of visited states
R : number of consecutive successes to observe
$R_{\max}$: maximal number of rounds

$R_{current} = 0$: current number of rounds
$R_{while} = 0$: current number of consecutive successes
$A_{\min} = A_{\max} = \hat{A}$
while $R_{while} < R \wedge R_{current} < R_{\max}$ **do**
  for $i \in V$ **do**
    $K_i = \min_j \frac{\hat{a}_{ij}(1-\hat{a}_{ij})}{\epsilon_{ij}^2} - 1$
    $(a_i \notin [\hat{a}_i]) = \top$
    while $a_i \notin [\hat{a}_i]$ **do**
      generate $a_i \sim$ Dirichlet$(K_i \hat{a}_i))$
    end while
  end for
  evaluate $f(A)$
  if $f(A) < f(A_{\min})$ **then**
    $A_{\min} = A$ and $R_{while} = 0$
  else
    if $f(A) > f(A_{\max})$ **then**
      $A_{\max} = A$ and $R_{while} = 0$
    else
      $R_{while} = R_{while} + 1$
    end if
  end if
  $R_{current} = R_{current} + 1$
end while

Output: $f(A_{\min})$ and $f(A_{max})$

These values are supposed to be unknown but still to fall within the respective intervals: $a \in [0.5; 5.5] \times 10^{-4}$ and $c \in [0.0493; 0.0503]$. Recall that $\gamma = ac/(1-ad)$ is the probability of reaching s_2 from s_0.

We sample under the perfect importance sampling distribution B defined with respect to the centred DTMC $\hat{A}$ parametrised by $\hat{a} = 3 \times 10^{-4}$ and $\hat{c} = 0.0498$.

This example illustrates the difference of results between our approach and the standard importance sampling approach for DTMCs. For each experiment, we calculate and then report in Table I the descriptive statistics of the number of rounds n_r necessary to find the minimum and the maximum, the corresponding matrices $A_{\min}$ and $A_{\max}$ (respectively described by the couples $(a_{\min}, c_{\min})$ and $(a_{\max}, c_{\max})$). We remark that on average, it takes between 181 and 3119 rounds to converge close to $A_{\min}$ and $A_{\max}$.

In Table II, we report the average bounds of the confidence intervals obtained by IS and by IMCIS and their mid-value. Note that the IS confidence interval is centred on $\hat{\gamma}_N(\hat{A})$ and may be then slightly different than the IMCIS mid-value. Since we sampled under the perfect distribution, the exact value of the centred DTMC, $\gamma(\hat{A}) = 1.4944 \times 10^{-5}$ is always contained in the importance sampling confidence interval. But this 100% coverage for $\gamma(\hat{A})$ drops to zero for γ since the confidence interval is reduced to $\gamma(\hat{A})$. In comparison, the IMCIS confidence interval has a 100% coverage for both $\gamma(\hat{A})$ and γ.

B. Group repair model

The following benchmark is a reliability model taken from [24], small enough (125 states) to be investigated using PRISM [17] to corroborate our results. The system is modelled as a continuous time Markov chain and comprises three types of subsystems $(1, \cdots, 3)$ containing, respectively, 4 components that may fail independently. The components fail with rates $(\alpha^2, \alpha, \alpha)$ where $\alpha = 0.1$ is supposed to be unknown, and are repaired with rate 1. In addition, components are repaired with priority according to their type (type i has highest priority than type j if $i < j$). The components of type 2 and 3 are repaired one by one as soon as one has failed whereas components of type 1 are repaired all together as soon as more than two of them have failed. The property we consider is the

Table I: Illustrative example with $a \in [5, 5.5] \times 10^{-5}$ and $c \in [0.0493, 0.0503]$.

	n_r	$\hat{a}_{\min}$	$\hat{c}_{\min}$	$\hat{a}_{\max}$	$\hat{c}_{\max}$
average	2181	5.02×10^{-5}	0.0496	5.48×10^{-4}	0.0501
min	1244	5×10^{-5}	0.0493	5.45×10^{-4}	0.0494
max	4119	5.1×10^{-5}	0.0502	5.5×10^{-4}	0.0503
st. dev.	580	2.11×10^{-7}	2.2×10^{-4}	1.25×10^{-6}	1.63×10^{-4}

Table II: Comparison between IS and IMCIS.

		95%-CI	Mid value	Coverage of $\gamma(\hat{A})$	Coverage of γ
Illustrative example	IS	$[1.494 \pm 0] \times 10^{-5}$	1.494×10^{-5}	100%	0%
	IMCIS	$[0.249; 2.7] \times 10^{-5}$	1.499×10^{-5}	100%	100%
Group repair	IS	$[1.104; 1.171] \times 10^{-7}$	1.138×10^{-7}	80%	27%
	IMCIS	$[1.029; 1.216] \times 10^{-7}$	1.123×10^{-7}	100%	75%
SWaT	IS	$[1.2; 1.7] \times 10^{-2}$	1.45×10^{-2}	-	-
	IMCIS	$[0.7; 2.2] \times 10^{-2}$	1.45×10^{-2}	-	-

probability of reaching a failure state that corresponds to the failure of all the components, before returning to the initial state of no failures. The probabilistic transitions are symbolic functions of α. For $\alpha = 0.1$, $\gamma = 1.179 \times 10^{-7}$.

In the following experiments, we used frequentist inference to compute an estimate $\hat{\alpha} = 0.0995$ and calculated a 99.9%-confidence interval CI: $\alpha \in [0.09852; 0.10048]$. We can then easily build an IMC $[A(\hat{\alpha})]$ centred on $\hat{A} = A(\hat{\alpha})$. Note that $\gamma(\hat{A}) = 1.117 \times 10^{-7}$. We then determined an importance sampling distribution by the cross-entropy algorithm for DTMC described in [24].

Table II shows that the empirical IS coverage for $\gamma(\hat{A})$ is already below 95%. This problem is well-known and documented (e.g. [25]). As for the illustrative example, the IMCIS confidence interval is larger and its coverage of $\gamma(\hat{A})$ remains perfect. The problem comes from a poor estimation of the likelihood ratio standard deviation. Detecting this phenomenon is an open problem, in practice tackled by increasing N_{IS}. On the contrary, the IMCIS confidence interval keeps a perfect coverage of $\gamma(\hat{A})$ and remains good with respect to the exact model when the IS coverage of γ drops to 27%.

Figure 2 shows a superposition of IS and IMCIS CI for the repair model. Even if the experiments have been made independently, the IS confidence intervals are almost always fully contained in the IMCIS confidence intervals, that prove empirically a better reliability. Figure 3 illustrates the evolution of the IMCIS confidence bounds of an IMCIS experiment during the optimisation step. Figure 5 shows the range of probabilities for the repair model given the interval $[\hat{\alpha} = 0.09852; 0.10048]$. Note that the average IMCIS confidence interval in Table II covers 83% of the interval of probabilities defined by $\gamma(A(\alpha))$.

C. Repair model

The following benchmark is also a failure-repair process taken from [24]. This benchmark is larger (40820 states) and is composed of 6 subsystems with respectively $5, 4, 6, 3, 7, 5$ components that fail with rates $(2.5\alpha, \alpha, 5\alpha, 3\alpha, \alpha, 5\alpha)$ where α belongs to interval $[0.8236 \times 10^{-3}, 1.1764 \times 10^{-3}]$, and are repaired with rate $(1, 1.5, 1, 2, 1, 1.5)$. As in the group repair model, components are repaired with priority according to their type (type i has highest priority than type j if $i < j$). However, the components are all repaired one by one as soon as one has failed. The property we consider is the probability of reaching a failure state that corresponds to the failure of all the components of at least one type, before returning to the initial state of no failures. We assume that $\alpha = 0.001$ in the IS experiments. For this value, $\gamma = 7.488 \times 10^{-7}$. We repeated five times our experiments. The 95% confidence intervals obtained by IS captured values in $[7.3895 \times 10^{-7}, 7.5205 \times 10^{-7}]$ while IMCIS captured values between $[5.6884 \times 10^{-7}, 9.5491 \times 10^{-7}]$. Both set of experiments are thus satisfying on this large model with respect to $\alpha = 0.001$. However, if α is not in the interval $[0.99 \times 10^{-3}, 1.1 \times 10^{-3}]$, the IS intervals do not contain the exact value γ whereas the IMCIS intervals still contain γ if α is in $[0.88 \times 10^{-3}, 1.12 \times 10^{-3}]$.

D. Secure Water Treatment model

The SecureWater Treatment testbed (SWaT) built at Singapore University of Technology and Design is a scale-down version of a real industry water treatment plant [1]. The testbed is built to facilitate research on cyber security for CPS, which has the potential to be adopted to Singapore's water treatment systems. SWAT consists of a modern six-stage process. The process begins by taking in raw water, adding necessary chemicals to it, filtering it via an Ultrafiltration (UF) system, de-chlorinating it using UV lamps, and then feeding it to a Reverse Osmosis (RO) system. A backwash process cleans the membranes in UF using the water produced by RO. We refer to [1] for more details about the system and the datasets. Automatic model learning techniques are used to construct a set of Markov chains through abstraction and refinement, based on long system execution logs. The model can be described by 70-state DTMC and IMC. Our initial state is a failure state of the system that is repaired in about 5 step units. We want to estimate probability γ that the water level

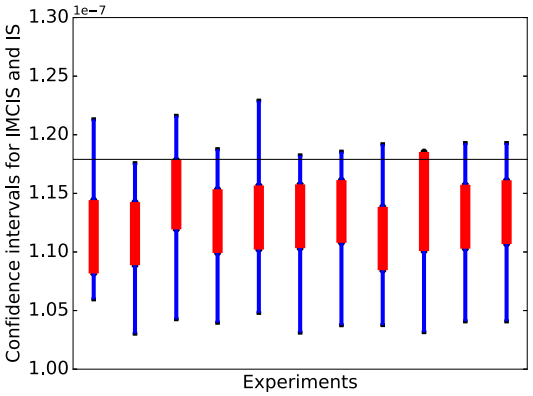


Figure 2: Repair model. Superposition of independent IS (red, thick) and IMCIS (blue, thin) 95%-confidence intervals. The black line indicates $\gamma = 1.179 \times 10^{-7}$.

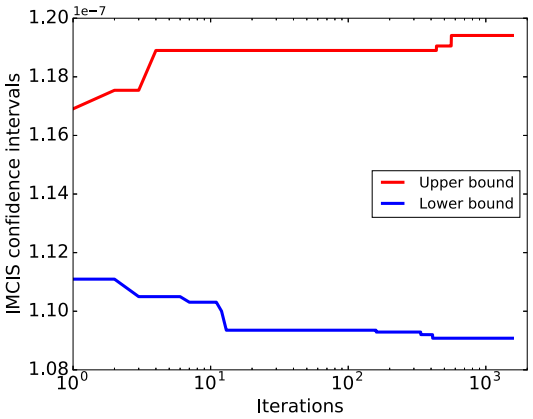


Figure 3: Repair model. Evolution of the IMCIS confidence interval bounds during the optimisation step. x-axis in log scale to show the fast changes in the first iterations.

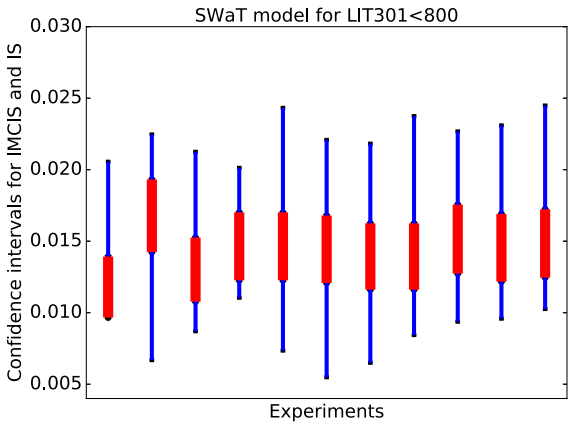


Figure 4: Water treatment model. Independent IS (red, thick) and IMCIS (blue, thin) 99%-CIs.

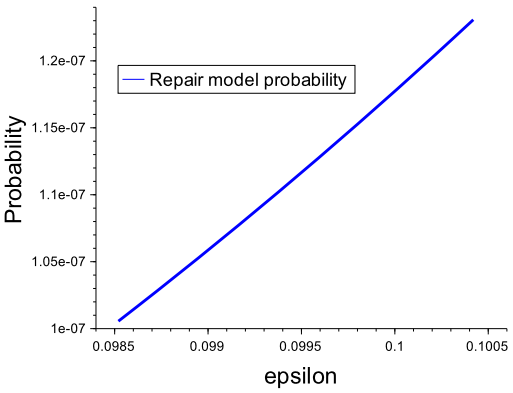


Figure 5: Ridder model probabilities for $\alpha \in [0.09852; 0.10048]$. Values calculated by PRISM.

indicator (LIT301) exceeds a threshold (> 800) within the next 30 step units. γ is unknown but supposedly small. The experiments suggest that $\gamma(\hat{A}) \in [5\times10^{-3}; 2.5\times10^{-2}]$. Resuts are reported in Table II (SWaT model). Figure 4 shows that IS is hardly reliable since the (red) IS confidence intervals do not even intersect (see the two first red CI). On the other hand, IMCIS (in blue) provides more consistent results. It is also worth noticing that the union of these IS confidence intervals is a subinterval of most of the IMCIS confidence intervals. Since the larger width of the IMCIS confidence intervals offer more chance to catch exact probability γ, we recommend the recourse to IMCIS for the estimation of CPS critical events.

VII. Conclusion

The focus of this paper was to introduce importance sampling in an IMC settings in order to take into account margin of errors inherent to approximated models. The goal of this approach is to provide more reliable confidence intervals of dependable properties in the rare event context, defined with respect to the original system. We proposed an algorithm based on random search optimisation using Dirichlet distributions to achieve this problem. The full validity of the approach is not achieved but our results are very promising and show great improvements over the similar importance sampling approach defined in the DTMC settings. As far as we know, the framework is novel and raises challenging questions. In particular, it would be interesting to compare the current algorithm with other optimisation schemes proper to our problem. An other challenge is to define a 'best' importance sampling distribution in the IMC settings. Finally, the uncertainties of the model may lead to large confidence intervals. In a future work, we plan to use it for improving the learning of a probabilistic system and to apply our approach to larger cyber physical systems.

ACKNOWLEDGMENT

This work was supported in part by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity R&D Programme (Award No. NRF2014NCR-NCR001-040) and administered by the National Cybersecurity R&D Directorate.

REFERENCES

[1] https://itrust.sutd.edu.sg/research/testbeds/secure-water-treatment-swat/.

[2] Benoît Barbot, Serge Haddad, and Claudine Picaronny. Coupling and Importance Sampling for Statistical Model Checking. In *Tools and Algorithms for the Construction and Analysis of Systems - 18th International Conference, TACAS*, pages 331–346, 2012.

[3] Anicet Bart, Benoît Delahaye, Didier Lime, Eric Monfroy, and Charlotte Truchet. Reachability in parametric interval markov chains using constraints. In *Quantitative Evaluation of Systems - 14th International Conference, QEST 2017, Berlin, Germany, September 5-7, 2017, Proceedings*, pages 173–189, 2017.

[4] Michael Benedikt, Rastislav Lenhardt, and James Worrell. LTL model checking of interval markov chains. In *Tools and Algorithms for the Construction and Analysis of Systems - 19th International Conference, TACAS 2013, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2013, Rome, Italy, March 16-24, 2013. Proceedings*, pages 32–46, 2013.

[5] Peter Carbonetto, Mark W. Schmidt, and Nando de Freitas. An interior-point stochastic approximation method and an l1-regularized delta rule. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, pages 233–240, 2008.

[6] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Statist.*, 23(4):493–507, 1952.

[7] Edmund M. Clarke and Paolo Zuliani. Statistical Model Checking for Cyber-Physical Systems. In *Automated Technology for Verification and Analysis, 9th International Symposium, ATVA 2011, Taipei, Taiwan, October 11-14, 2011. Proceedings*, pages 1–12, 2011.

[8] G. Cochran. Laplace's ratio estimator. In *Contributions to survey sampling and applied statistics*, pages 3–10. Academic Press, 1978.

[9] Costas Courcoubetis and Mihalis Yannakakis. Verifying temporal properties of finite-state probabilistic programs. In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 338–345, 1988.

[10] Benoît Delahaye, Kim G. Larsen, Axel Legay, Mikkel L. Pedersen, and Andrzej Wasowski. Consistency and refinement for interval markov chains. *J. Log. Algebr. Program.*, 81(3):209–226, 2012.

[11] William A Gale and Geoffrey Sampson. Good-Turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2(3):217–237, 1995.

[12] Thomas Hérault, Richard Lassaigne, Frédéric Magniette, and Sylvain Peyronnet. Approximate probabilistic model checking. In Bernhard Steffen and Giorgio Levi, editors, *Verification, Model Checking, and Abstract Interpretation*, volume 2937 of *LNCS*, pages 307–329. Springer, 2004.

[13] Cyrille Jegourel, Axel Legay, and Sean Sedwards. Importance splitting for statistical model checking rare properties. In Natasha Sharygina and Helmut Veith, editors, *Computer Aided Verification*, volume 8044 of *LNCS*, pages 576–591. Springer, 2013.

[14] Cyrille Jegourel, Axel Legay, and Sean Sedwards. Command-based importance sampling for statistical model checking. *Theor. Comput. Sci.*, 649:1–24, 2016.

[15] Sumit Kumar Jha, Edmund M. Clarke, Christopher James Langmead, Axel Legay, André Platzer, and Paolo Zuliani. A bayesian approach to model checking biological systems. In *Computational Methods in Systems Biology, 7th International Conference, CMSB 2009, Bologna, Italy, August 31-September 1, 2009. Proceedings*, pages 218–234, 2009.

[16] Bengt Jonsson and Kim Guldstrand Larsen. Specification and refinement of probabilistic processes. In *Proceedings of the Sixth Annual Symposium on Logic in Computer Science (LICS '91), Amsterdam, The Netherlands, July 15-18, 1991*, pages 266–277, 1991.

[17] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 2.0: A Tool for Probabilistic Model Checking. In *QEST*, pages 322–323. IEEE, 2004.

[18] Tze Leung Lai. Stochastic Approximation. *The Annals of Statistics*, 31(2):391–406, April 2003.

[19] Arkadi S. Nemirovski and Michael J. Todd. Interior-point methods for optimization. *Acta Numerica*, 17:191–234, May 2008.

[20] G. Norman, D. Parker, M. Kwiatkowska, and S. Shukla. Evaluating the reliability of NAND multiplexing with PRISM. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(10):1629–1637, 2005.

[21] Masashi Okamoto. Some Inequalities Relating to the Partial Sum of Binomial Probabilities. *Annals of the Institute of Statistical Mathematics*, 10:29–35, 1958.

[22] Martin L. Puterman. *Markov Decision Processes*. Wiley, 1994.

[23] Daniël Reijsbergen, Pieter-Tjerk de Boer, Werner R. W. Scheinhardt, and Boudewijn R. Haverkort. Rare Event Simulation for Highly Dependable Systems with Fast Repairs. In *QEST 2010, Seventh International Conference on the Quantitative Evaluation of Systems, Williamsburg, Virginia, USA, 15-18 September 2010*, pages 251–260, 2010.

[24] Ad Ridder. Importance sampling simulations of markovian reliability systems using cross-entropy. *Annals of Operations Research*, 134:119–136, 2005.

[25] Gerardo Rubino and Bruno Tuffin, editors. *Rare Event Simulation using Monte Carlo Methods*. Wiley, 2009.

[26] J. Spall. *Introduction to Stochastic Search and Optimisation*. Wiley, 2003.

[27] Moshe Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 327–338, 1985.

[28] Abraham Wald. Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, 1945.

[29] H. Younes. *Verification and Planning for Stochastic Processes with Asynchronous Events*. PhD thesis, Carnegie Mellon University, 2004.

[30] Lijun Zhang, Holger Hermanns, and David N. Jansen. Logic and model checking for hidden markov models. In *Formal Techniques for Networked and Distributed Systems - FORTE 2005, 25th IFIP WG 6.1 International Conference, Taipei, Taiwan, October 2-5, 2005, Proceedings*, pages 98–112, 2005.

APPENDIX

Various optimisation algorithms could be used in our context. Their efficiency is measured in terms of speed of convergence. Since the objective function is unlikely to be linear or at most quadratic, we can only consider non-linear algorithms. However, since the constraints are linear, statistical convex optimisation methods are relevant to our problem. General stochastic techniques, like genetic or Metropolis-Hastings algorithms may also be considered but they are not limited to convex problems and are usually less efficient in terms of speed of convergence than the convex algorithms discussed below.

1) Stochastic Gradient Descent [18]: is a stochastic approximation of the gradient descent optimization method for minimising an objective function that is written as a sum of differentiable functions, i.e. $f(A) = \sum_{k=1}^{M} L(\omega_k; A)$ where ω_k is the k-th successful path, $A \in [\hat{A}]$ and $L(\omega; A)$ denotes the likelihood of path ω given original probabilistic measure A. In the standard gradient descent, $A^{(0)} = \hat{A}$ and the parameter $A^{(j)}$ is updated at iteration $j+1$ by:

$$A^{(j+1)} = A^{(j)} - \eta \nabla f(A^{(j)}) \tag{13}$$

where η is a parameter called the learning rate in machine learning. In the stochastic gradient descent, the gradient $\nabla f(A^{(j)})$ is approximated by the gradient of only one sample:

$$A^{(j+1)} = A^{(j)} - \eta \nabla L(\omega_k; A^{(j)}) \tag{14}$$

where ω_k is chosen randomly in the set of the sampled successful paths. The convergence of the numerical and stochastic gradient descents has been analysed. The main advantage of the stochastic gradient descent is that, in our context, the gradient of a sample is easy to calculate since $L(\omega_k; A^{(j)})$ has a polynomial form. However, $A^{(j+1)}$ in Equation (14) obviously does not satisfy the equality constraints ($\sum_{j=1}^{m} a_{ij}^{(j+1)} \neq 1$). After re-normalisation, the equality constraints are satisfied but not necessarily the inequality constraints. A projection into $[\hat{A}]$ is thus necessary after solving Equation (14). Unfortunately, the projection step must be performed after each iteration by minimisation of the distance between $A^{(j+1)}$ and $[\hat{A}]$, that implies significant time over-cost.

2) Stochastic Interior Point Method [5]: is more suited for dealing inequality constraints since, at each iteration, an update is directly found into $[\hat{A}]$ by the means of the logarithmic barrier method. To apply this method, we rewrite the constrained optimisation problem as an unconstrained optimisation problem of the form:

$$\underset{A}{\text{minimize}}\ f(A) - \sum_{i=0}^{m} \lambda_i c_i - \sum_{i=0}^{m}\sum_{j=0}^{m} (\mu_{ij}^{-} \log(c^{-}(a_{ij})) + \mu_{ij}^{+} \log(c^{+}(a_{ij})))$$

where each λ_i is a Lagrange multiplier assigned to the constraint $\sum_j a_{ij} = 1$ and $\mu_{ij}^{-}, \mu_{ij}^{+} > 0$ are the barrier parameters of a_{ij}. In [5], the authors propose an approximation of the minimum using a stochastic version of the interior point method. However, in our context, the number of constraints may be huge and slow down the solving of the resulting system of equations. Indeed, we must take into account m equality constraints and a maximum of $2m^2$ inequality constraints. Solving the system of polynomial equations enriched with one Lagrangian multiplier per constraint quickly becomes intractable with respect to the number of states. Moreover, a proof of convergence is still missing according to the authors[2].

A. Prism code for the repair benchmark

We give below the code of the Prism model and the property under investigation. α must be set by the user.

```
ctmc

const int n=4;
const double alpha = 0.1;
const double alpha2 = alpha*alpha;
const double mu = 1.0;

module type1
state1 : [0..n] init 0;
[] state1 < n -> (n-state1)*alpha2 :
 (state1'=state1+1);
[] state1 >=2 -> mu : (state1'=0);
endmodule

module type2
state2 : [0..n] init 0;
[] state2 < n -> (n-state2)*alpha :
(state2'=state2+1);
[] state2 >=2 & state1 < 2 -> mu :
(state2'=0);
endmodule

module type3
state3 : [0..n] init 0;
[] state3 < n -> (n-state3)*alpha :
 (state3'=state3+1);
[] state3 > 0 & state2 < 2 & state1 < 2
 -> mu : (state3'=state3-1);
endmodule

label "failure" = state1 = n & state2 = n
 & state3 = n;
```

The property code is:

```
P=?["init" & (X !"init" U "failure")]
```

[2]The convergence was initially established in the appendix of [5] but the authors admitted on their webpage on https://pcarbo.github.io a "major flaw" in the convergence proof. So far, the convergence is still an open question