Command-based importance sampling for statistical model checking

Cyrille Jegourel, Axel Legay, Sean Sedwards

Command-based Importance Sampling for Statistical Model Checking

Cyrille Jegourel, Axel Legay and Sean Sedwards

INRIA Rennes – Bretagne Atlantique

Abstract

Statistical model checking avoids the exponential growth of states of numerical model checking, but rare properties are costly to verify. Importance sampling can reduce the cost if good importance sampling distributions can be found efficiently.

Our approach uses a tractable cross-entropy minimisation algorithm to find an optimal parametrised importance sampling distribution. In contrast to previous work, our algorithm uses a naturally defined low dimensional vector to specify the distribution, thus avoiding an explicit representation of a transition matrix. Our parametrisation leads to a unique optimum and is shown to produce many orders of magnitude improvement in efficiency on various models. In this work we link the existence of optimal importance sampling distributions to logical properties and show how our parametrisation affects this link. We also motivate and present simple algorithms to create the initial distribution necessary for cross-entropy minimisation. Finally, we discuss the open challenge of defining error bounds with importance sampling and describe how our optimal parametrised distributions may be used to infer qualitative confidence.

1. Introduction

The most common method to ensure the correctness of a system is by testing it with a number of test cases having predicted outcomes that can highlight specific problems. Such testing techniques remain the default in industrial contexts and have also been incorporated into sophisticated tools [1]. Despite this, testing is limited by the need to hypothesise scenarios that may cause failure and the fact that a reasonable set of test cases is unlikely to cover all possible eventualities. Errors and modes of failure in complex systems may remain undetected and quantifying the likelihood of failure using a series of test cases is difficult.

Static analysis has been successful in debugging very large systems [2], but its ability to analyse dynamical properties is limited by its level of abstraction. In contrast, model checking is a fine-grained exhaustive technique that verifies whether a system satisfies a dynamical temporal logic property under all possible scenarios. In recognition of the existence of nondeterministic and probabilistic

systems, and the fact that a Boolean answer is not always useful, *numerical* model checking quantifies the probability that a system satisfies a property. Numerical model checking offers precise and accurate analysis by exhaustively exploring the state space of probabilistic systems. The result of this technique is the notionally exact probability (i.e., within the limits of numerical precision and convergence stability) that a system will satisfy a property of interest, however the exponential growth of the state space limits its applicability. The typical state limit of exhaustive approaches usually represents an insignificant fraction of the state space of "real" systems. Such systems may have tens of orders of magnitude more states than the number of protons in the universe ($\approx 10^{80}$).

Symbolic model checking using efficient data structures can make certain very large models tractable [3]. It may also be possible to construct simpler but behaviourally equivalent abstractions using various symmetry reduction techniques, such as partial order reduction, bisimulation and lumping [4]. Compositional approaches may also help. In particular, components of a system may be specified in such a way that each is tractable to analysis, while their properties guarantee that certain faults are impossible. Despite these techniques, however, the size, unpredictability and heterogeneity of real systems [5] often make numerical techniques infeasible. Moreover, even if a system has been specified not to misbehave, it is nevertheless necessary to check that it meets its specification.

While the 'state explosion problem' [6] is unlikely to ever be entirely solved for all systems, simulation-based approaches are becoming increasingly tractable due to the availability of high performance parallel hardware and algorithms. In particular, *statistical* model checking (SMC) combines the simplicity of testing with the formality of numerical model checking. The core idea of SMC is to create multiple independent execution traces of the system and individually verify whether they satisfy some formally specified property. The proportion of satisfying traces is an estimate of the probability that the system satisfies the property. By thus modelling the executions of a system as a Bernoulli random variable, the absolute error of the estimate can be bounded using, for example, a confidence interval [7, Chap. 1] or a Chernoff bound [8, 9, 10]. It is also possible to use efficient techniques, such as Bayesian inference [11] and hypothesis testing [12, 13], to decide with specified statistical confidence whether the probability of a property is above or below a given threshold.

Knowing a result with less than 100% confidence is often sufficient in real applications, since the confidence bounds may be made arbitrarily tight. Moreover, a swiftly achieved approximation may prevent a lot of wasted time during model design. For many complex systems, SMC offers the only feasible means of quantifying performance. Evidence of this is that SMC has been used to find bugs in large, heterogeneous aircraft systems [5]. Dedicated SMC platforms include APMC [14], YMER [15], VESTA [16], PLASMA [17] and COSMOS [18]. Well-established numerical model checkers, such as PRISM [19] and UPPAAL [20], are now also including SMC engines. Indeed, since SMC may be applied to any discrete event trace obtained by stochastic simulation, [21] describes a modular library of SMC algorithms that may be used to construct domain-specific SMC tools.

SMC relies on multiple independent simulations, so it may be efficiently divided on parallel computer architectures, such as grids, clusters, clouds and general purpose computing on graphics processors (GPGPU). Despite this, rare properties require a challenging number of simulations. Standard error bounding strategies for SMC consider *absolute* error. As the probability of a property decreases, however, it is more useful to consider an error bound that is relative to the probability. The number of simulations required to bound the *relative* error, defined as the standard deviation of the estimate divided by its expectation, is inversely proportional to rarity. Hence, while SMC may make a verification task feasible, it may nevertheless be computationally intense. To address this problem, in this work we apply the variance reduction technique of *importance sampling* to statistical model checking.

Importance sampling works by simulating a system under a weighted (importance sampling) distribution that makes a property more likely to be seen. It then compensates the results by the weights, to estimate the probability under the original distribution. The concept arose from work on the 'Monte Carlo method' [22] in the 1940s and was originally used to quantify the performance of materials and solve otherwise intractable analytical problems with limited computer power (see, e.g., [23]).

For importance sampling to be effective it is necessary to define a "good" importance sampling distribution: (*i*) the property of interest must be seen frequently in simulations and (*ii*) the distribution of the simulation traces that satisfy the property in the importance sampling distribution must be as close as possible to the normalised distribution of the same traces in the original distribution. Failure to consider both (*i*) and (*ii*) can result in gross errors and overestimates of confidence. Moreover, the process of finding a good importance sampling distribution must itself be efficient and, in particular, should not rely on iterating over all the states or transitions of the system. The algorithms we present in this work address all these issues.

The term 'rare event' is ubiquitous in the literature. Here we specifically consider rare *properties* of paths, defined in bounded temporal logic (bounded by time or number of steps). This extends the common notion of rarity from states to paths. States are rare if the probability of reaching them from the initial state is small. Paths are rare if the probability of executing their sequence of states is unlikely–whether or not the states themselves are rare. Rare properties are therefore more general than rare states, however the distinction does not significantly alter the mathematical derivation of our algorithms. It can nevertheless affect the existence of the so-called "zero variance" optimal importance sampling distribution as a simple re-parametrisation of the states and transitions of the original system. We explore this important subject in Section 7.

1.1. Contribution

This work extends [24], describing the additional techniques necessary to apply our importance sampling framework for statistical model checking of rare events. We describe simple algorithms to initiate the cross-entropy minimisation

process by finding at least a few traces that satisfy the property. We believe this subject has been glossed over in previous work. Simple heuristics, such as unifying the probabilities of transitions from a given state, may fail if rarity is not related to low probability transitions. We also describe and illustrate some of the key phenomena relating to parametrised importance sampling and clarify some recent misconceptions about confidence.

To apply SMC to discrete space Markov models with rare properties, our approach is based on a simply-implemented cross-entropy minimisation algorithm that finds an optimal set of parameters to characterise an importance sampling distribution. We have previously shown that there is a unique optimum and that our algorithm converges [24]. Our parametrisation is at the level of *guarded commands* [25] and arises naturally from standard syntactic descriptions of models. It is thus a very tractable low dimensional vector in comparison to the state space of the model. As such, the family of distributions induced by the parametrisation is unlikely to contain the zero variance distribution, but this is not necessary for practical applications. In practice, it is sufficient to reduce the variance without over-emphasising any particular part of the trace space. We contend that a parametrisation at the level of a low level syntax, as in our case, is well placed to achieve this efficiently. Such a syntactical description of a model necessarily defines the symmetries evident in the distribution of behaviour and typically contains specific elements relevant to the property of interest.

While there will always exist pathological systems and models that are intractable to our approach, we are able to demonstrate very substantial reductions in variance on a number of models with very few parameters. To illustrate the theoretical benefits, though not necessarily practical benefits, we also show how increasing the number of parameters of a model allows the parametrised importance sampling distribution to better approximate the zero variance distribution.

1.2. Related work

Importance sampling was invented as a means to accelerate simulations of rare events [23]. Since then it has become a standard technique to allow the behaviour of a system with an "inconvenient" distribution to be simulated by a more convenient one. Cross-entropy (also known as Kullback-Leibler divergence [26]) is a standard information-theoretic measure of directed distance between distributions. The cross-entropy method [27, 28] is an algorithmic framework that facilitates the convergence of an arbitrarily parametrised distribution to an optimal distribution, without an explicit (closed form) representation of the optimal distribution. Given the general applicability of these techniques, in what follows we highlight only those recent contributions that are of specific relevance to the present work.

As a precursor to SMC, earlier work considered 'highly reliable systems' comprising components that fail probabilistically and are then repaired [29, 30, 31]. Since many critical systems need to be highly reliable, failure is often a rare event of critical importance. A challenge arises because real systems tend to

have a size and complexity that is intractable to exhaustive analysis. A focus of research in this field is therefore finding good parametrised importance sampling distributions that do not require analysis at the level of individual transition probabilities.

In [29] the author defines two structural properties of Markovian systems that allow good importance sampling distributions to be created by 'simple failure biasing'. If (i) a repair is possible in any state and (ii) the failure rates are balanced (i.e., bounded), then an importance sampling estimator that merely biases the failure rate can have a bounded relative error. If only (i) holds, the same work proposes a 'balanced failure biasing' algorithm that finds a similarly good importance sampling distribution. If (i) does not hold, such biasing schemes may fail due to the existence of high probability cycles with 'group repair' (simultaneous repair of multiple failed components). To address this problem, in [32, 33] the authors propose increasingly complex biasing schemes that make use of other structural properties.

In [31] the author defines an algorithm to construct optimal importance sampling distributions for repair models, using the cross-entropy method [27]. By parametrising a system at the level of individual transition probabilities (effectively the lowest possible level), the algorithm will converge to the perfect 'zero variance' importance sampling distribution (defined in Section 3) when it exists (see Section 7). Hence, [31] addresses the problem of systems containing group repair, but at the prohibitive cost of iterating over every transition. Since numerical algorithms have similar complexity, but avoid the cost of simulation and give results with near certainty, [31] does not provide a practical solution for SMC.

In [34] the authors construct an importance sampling distribution for 'highly dependable systems', based on dominant paths to failure. The authors frame their results in the context of (statistical) model checking, but focus on a standard reliability model and do not consider the specific problems that logical properties incur. While the results may not be generalisable, by constructing distributions based on an analysis of bounded paths, the work nevertheless hints at future directions of research applying rare event techniques to SMC.

In [35] the authors present a specific application of the cross-entropy method to a simple continuous time failure model. The system comprises independent components that fail at times that are exponentially distributed. By considering the first simultaneous failure of all components, the authors are able to use a standard closed form solution to find an importance sampling distribution that increases the occurrence of this rare event. Although the notions of temporal logic and SMC are introduced, they effectively play no part because the technique is not generalisable to other properties or systems.

In [36] the authors attempt to address the important challenge of bounding the error of estimates when using importance sampling with SMC (we discuss this open challenge in Section 8). The work contains some interesting ideas, but does not yet provide practical solutions. The basic notion is to perform numerical analysis on a reduced (abstracted) model of a system, in order to infer on the fly an importance sampling distribution that guarantees statistical

confidence. The authors assume the existence of a suitable property-specific abstraction function that maps states in the full model to states in the abstracted model, such that all abstracted traces that satisfy the property have probability greater than or equal to the traces they abstract. No algorithmic means of generating such a function is provided and we believe this is generally non-trivial. The 'coupling' mentioned in the title is included as a way to verify that an existing function is correct.

Importance *splitting* is another variance reduction technique developed in the 1940s [23]. Rather than modify the dynamics of the system, as in the case of importance sampling, the technique relies on properties that may be decomposed into a sequence of dependent 'levels'. The overall probability is thus decomposed into the product of probabilities of going from one level to the next. Since these probabilities are necessarily larger than the overall probability, their estimation is generally easier. In [37] the authors define how importance splitting may be used to verify rare properties in the context of statistical model checking, defining how properties specified in temporal logic may be decomposed into levels. Importance sampling and splitting are not mutually exclusive, hence future work may combine them in the context of SMC.

1.3. Structure of the Article

Section 2 describes notation we will use in the sequel, while Section 3 introduces the basic notions of Monte Carlo integration and importance sampling. Section 4 introduces the cross-entropy minimisation framework. Our command-based cross-entropy minimisation approach is fully described in Section 5, with the results of applying it to some case studies given in Section 6. In Section 7 we consider the conditions under which optimal importance sampling distributions exist as parametrisations of the original system. In Section 8 we discuss the challenges of specifying the confidence of importance sampling estimates. Section 9 concludes the paper.

2. Preliminaries

We consider systems described by discrete and continuous time Markov chains that may be infinite. We assume the models are specified by a set of stochastic guarded commands ('commands' for short) acting in parallel. Each command has the form (*guard*, *rate*, *action*). The *guard* enables the command and is a predicate over the state variables of the model. The *rate* is a function from the state variables to $\mathbb{R}_{>0}$, defining the rate of an exponential distribution. The *action* is an update function that modifies the state variables. In general, each command defines a set of semantically linked transitions in the resulting Markov chain. Models are thus described in a relatively compact and convenient way. The widely adopted PRISM language[1] is an example of a modelling language based on stochastic guarded commands.

[1] http://www.prismmodelchecker.org/manual/ThePRISMLanguage/

The semantics of a stochastic guarded command is a Markov jump process. The semantics of a parallel composition of commands is a system of concurrent Markov jump processes. Sample execution traces can be generated by discrete-event simulation (e.g., [38]). In any state, zero or more commands may be enabled. If no commands are enabled the system is in a halting state. In all other cases the enabled commands "compete" to execute their actions: sample times are drawn from the exponential distributions defined by their rates and the shortest time "wins".

Execution traces (equivalently paths) are thus sequences of the form $\omega = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} s_2 \xrightarrow{t_2} ...$, where each $s_i \in S$ is a state of the model and $t_i \in \mathbb{R}_{>0}$ is the time spent in the state s_i (the delay time) before moving to the state s_{i+1}. In the case of discrete time, $t_i \equiv 1, \forall i$. When we are not interested by the times of jump epochs, we denote a trace $\omega = s_0 s_1 ....$ The length of trace ω is the number of transitions it contains and is denoted $|\omega|$. We denote by $\omega_{\geq k}$ the suffix of ω starting at s_k, i.e., $s_k s_{k+1} \cdots$.

2.1. Statistical Model Checking

The process of statistical model checking estimates the probability that a system satisfies a property by the proportion of simulation traces in a random sample that individually satisfy it. To achieve this, the statistical model checker constructs an automaton to accept only traces satisfying a property specified using time bounded temporal logic. Given a randomly generated trace ω, the automaton outputs a 1 if ω is accepted and 0 otherwise. Statistical model checking can thus be seen as the estimation of the success parameter of a Bernoulli random variable with support $\{0, 1\}$. By using this abstraction, SMC is also able to test hypotheses about the parameter and inherits theory that allows the statistical confidence of results to be calculated.

The following abstract syntax is typical of a bounded linear temporal logic used in SMC:

$$\varphi = \alpha \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \mathbf{X}\varphi \mid \mathbf{F}^t\varphi \mid \mathbf{G}^t\varphi \mid \varphi\mathbf{U}^t\varphi \tag{1}$$

Symbol α denotes an *atomic proposition* that may be *true* or *false* in any state $s \in S$. Operators $\vee$, $\wedge$ and $\neg$ are the standard Boolean connectives. $\mathbf{F}^t$, $\mathbf{G}^t$ and $\mathbf{U}^t$ are temporal operators that apply to time interval $[0, t]$, where $t \in \mathbb{R}_{>0}$ may denote steps or real time and the interval is relative to the interval of any enclosing operator. We refer to this informally as a "relative interval". To simplify the following notation, it is assumed that if a property requires the next state to be satisfied and no next state exists, the property is not satisfied. Thus, given a property φ with syntax (1), the semantics of the satisfaction relation $\omega \models \varphi \equiv \omega_{\geq 0} \models \varphi$ is inductively defined as follows:

- $\omega_{\geq k} \models \alpha \iff \alpha$ evaluates to *true* in state s_k
- $\omega_{\geq k} \models \varphi_1 \vee \varphi_2 \iff \omega_{\geq k} \models \varphi_1 \vee \omega_{\geq k} \models \varphi_2$
- $\omega_{\geq k} \models \varphi_1 \wedge \varphi_2 \iff \omega_{\geq k} \models \varphi_1 \wedge \omega_{\geq k} \models \varphi_2$

- $\omega_{\geq k} \models \neg\varphi \iff \omega_{\geq k} \not\models \varphi$
- $\omega_{\geq k} \models \mathbf{X}\varphi \iff \omega_{\geq k+1} \models \varphi$
- $\omega_{\geq k} \models \mathbf{F}^t\varphi \iff \exists i \geq k \in \mathbb{N} : \sum_{l\in\{k,\ldots,i\}} t_l \leq t \wedge \omega_{\geq i} \models \varphi$
- $\omega_{\geq k} \models \mathbf{G}^t\varphi \iff \exists i \geq k \in \mathbb{N} : \sum_{l\in\{k,\ldots,i\}} t_l \leq t \wedge \sum_{l\in\{k,\ldots,i+1\}} t_l > t \wedge \forall l \in \{k,\ldots,i\} : \omega_{\geq l} \models \varphi$
- $\omega_{\geq k} \models \varphi_1\mathbf{U}^t\varphi_2 \iff \exists i \geq k \in \mathbb{N} : \sum_{l\in\{k,\ldots,i\}} t_l \leq t \wedge \omega_{\geq i} \models \varphi_2 \wedge (i = k \vee \forall l \in \{k,\ldots,i-1\} : \omega_{\geq l} \models \varphi_1)$

$\mathbf{F}^t$, $\mathbf{G}^t$ and $\mathbf{U}^t$ are related in the following way: $\mathbf{G}^t = \neg(\mathbf{F}^t\neg\varphi)$, $\mathbf{F}^t\varphi = \mathit{true}\mathbf{U}^t\varphi$, hence $\mathbf{G}^t\varphi = \neg(\mathit{true}\mathbf{U}^t\neg\varphi)$. Informally: $\mathbf{X}\varphi$ means that φ will be true in the next state; $\mathbf{F}^t\varphi$ means that φ will be true at least once in the relative interval $[0,t]$; $\mathbf{G}^t\varphi$ means that φ will always be true in the relative interval $[0,t]$; $\psi\mathbf{U}^t\varphi$ means that in the relative interval $[0,t]$, φ will eventually be true and ψ will be true until it is.

3. Monte Carlo Integration and Importance Sampling

Statistical model checking is based on the concept of Monte Carlo integration [39, Ch. 3]. Given a random variable X, with sample space $\omega \in \Omega$ and distribution f, the expectation of an integrable function $z(X)$ can be expressed as

$$\mathrm{E}_f[z(X)] = \int_\Omega z(\omega)\,\mathrm{d}f(\omega). \tag{2}$$

To estimate $\mathrm{E}_f[z(X)]$, Monte Carlo integration works by drawing N samples $\omega_i, i \in \{1,\ldots,N\}$, at random according to distribution f and calculating

$$\mathrm{E}_f[z(X)] \approx \frac{1}{N}\sum_{i=1}^{N} z(\omega_i). \tag{3}$$

With increasing N, the right hand side of (3) is guaranteed to converge to the left hand side by the law of large numbers. In the context of SMC, Ω is a probability space of paths, f is the probability measure over Ω and z represents the output of the model checking automaton described in Section 2.1. Later, in the specific context of SMC, we denote this particular function z by $\mathbb{1}(\omega \models \varphi)$, to emphasise its characteristics.

Figure 1a illustrates how (3) works. The outer square denotes the space of all traces Ω, the leaf shape denotes the set of traces that satisfy φ. The red dots are uniformly sampled at random from Ω, such that the fraction of samples falling within the leaf is an approximation of the probability that the system will satisfy φ. Figure 1b illustrates the problem when a property is rare. Fewer samples fall within the leaf and, moreover, the coverage of the leaf is apparently less uniform than in Fig. 1a. Unbiased convergence is still guaranteed with increasing N, but the variance of the estimate is higher.

Equation (3) can thus be used to estimate the probability that a system will satisfy a property φ, where γ is defined by

$$\gamma = \int_{\Omega} z(\omega)\,\mathrm{d}f(\omega) \tag{4}$$

and the standard "crude" Monte Carlo estimator of γ can be written

$$\tilde{\gamma} = \frac{1}{N_{\mathrm{MC}}} \sum_{i=1}^{N_{\mathrm{MC}}} z(\omega_i). \tag{5}$$

N_{MC} denotes the number of simulations used by the standard Monte Carlo estimator and ω_i is sampled according to f, denoted $\omega_i \sim f$. Note that $z(\omega_i)$ is effectively the realisation of a Bernoulli random variable with parameter γ. Hence $\mathrm{Var}(\tilde{\gamma}) = \gamma(1-\gamma)/N_{\mathrm{MC}}$ and for $\gamma \to 0$, $\mathrm{Var}(\tilde{\gamma}) \approx \gamma/N_{\mathrm{MC}}$.

Let f be absolutely continuous with respect to another probability measure f' over Ω, then (4) can be written

$$\gamma = \int_{\Omega} z(\omega)\frac{\mathrm{d}f(\omega)}{\mathrm{d}f'(\omega)}\,\mathrm{d}f'(\omega). \tag{6}$$

$L = \mathrm{d}f/\mathrm{d}f'$ is the *likelihood ratio* function, so

$$\gamma = \int_{\Omega} L(\omega)z(\omega)\,\mathrm{d}f'(\omega). \tag{7}$$

We can thus estimate γ by simulating under f' and compensating by L:

$$\tilde{\gamma} = \frac{1}{N_{\mathrm{IS}}} \sum_{i=1}^{N_{\mathrm{IS}}} L(\omega_i)z(\omega_i) \tag{8}$$

Here $\omega_i \sim f'$ and N_{IS} denotes the number of simulations used by the importance sampling estimator. The goal of importance sampling is to reduce the variance of the rare event and so achieve a narrower confidence interval than the Monte Carlo estimator, resulting in $N_{\mathrm{IS}} \ll N_{\mathrm{MC}}$. In general, the importance sampling distribution f' is chosen to produce the rare property more frequently. Figure 1c illustrates the basic notion of importance sampling. The sampling distribution is weighted in such a way that most of the samples fall within the leaf. The fraction of samples falling within the leaf is no longer an approximation of the probability we seek, but knowing the values of the weights it is possible to compensate and calculate an unbiased estimate.

The optimal importance sampling distribution, denoted f^* and defined as f conditioned on the rare event, produces only traces satisfying the rare property. Formally, we define f^* by

$$\mathrm{d}f^* = \frac{z\,\mathrm{d}f}{\gamma}. \tag{9}$$

Figure 1d illustrates the notion of the optimal importance sampling distribution. All the samples fall within the leaf and the coverage is uniform. This leads to

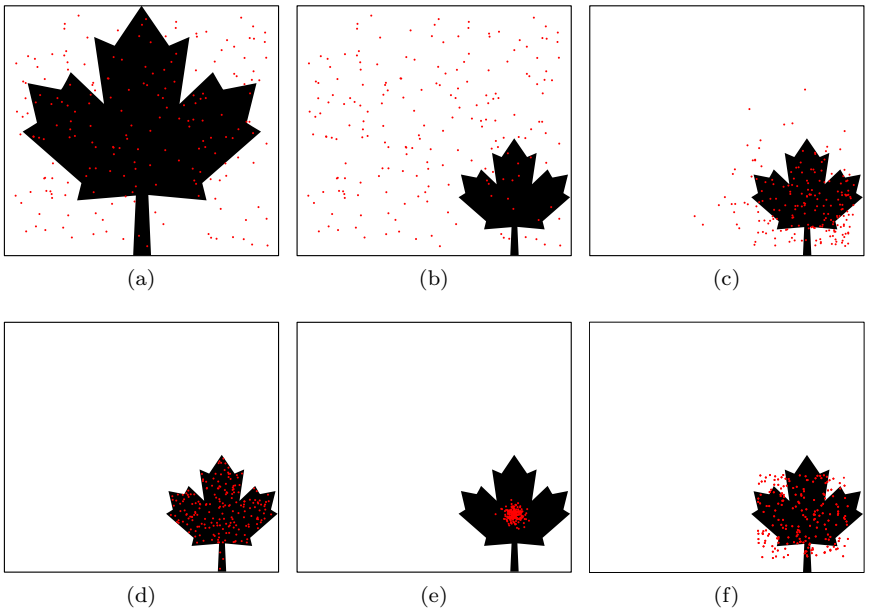

Figure 1: Monte Carlo integration.

the term 'zero variance estimator', since under f^*, traces for which $z = 0$ have zero probability of being seen and $L = \gamma$ whenever $z = 1$. Note, however, that in general the optimal importance sampling distribution f^* does not itself have zero variance.

In practice, it is usually only possible to observe the percentage of successful simulations and not possible to judge how uniformly the distribution covers the target area. Moreover, the percentage of success does not necessarily indicate the quality of the importance sampling distribution. For example, the distribution illustrated in Fig. 1e produces 100% success but is focused on only a small percentage of the target area. This distribution will have low sample variance, giving a false impression of high confidence, but will produce a severe underestimate of the true probability.

Figure 1f illustrates the notion of a minimum cross-entropy parametrised distribution. The distribution is more focused than Fig. 1c, but not pathologically so, like Fig. 1e. It does not perfectly cover the leaf, like the theoretically optimal distribution of Fig. 1d, because the optimal distribution is not a member of the family generated by its particular parametrisation. Intuitively, the minimum cross-entropy distribution is one which optimally balances focus and coverage, given the parametrisation.

3.1. Importance Sampling for Parametrised Systems

Importance sampling schemes have been described as falling into two broad categories: *state dependent tilting* and *state independent tilting* [40]. State de-

pendent tilting refers to importance sampling distributions that individually weight ('tilt') every transition probability in the system. State independent tilting refers to importance sampling distributions that change classes of transition probabilities, independent of state. State dependent tilting offers greatest precision, but is infeasible in a large model unless it can be done on the fly by a function that exploits the model's symmetries. Such symmetries do not always exist in real systems. State independent tilting is more tractable but may not produce good importance sampling distributions. Our approach may be seen as *parametrised* tilting, that potentially affects all transitions differently, but does so according to a set of parameters.

In the context of SMC, the distribution f introduced in Section 3 usually arises from the specifications of a model described in some relatively high level language. Such models do not, in general, explicitly specify the probabilities of individual transitions, but do so implicitly by parametrised functions over the states. We therefore consider a class of models that can be described by guarded commands [25] extended with stochastic rates. Our parametrisation is a vector of strictly positive values $\lambda \in \mathbb{R}^n_{>0}$ that *tilt* (multiply) the stochastic rates and thus maintain the absolutely continuous property between distributions. Note that this class includes both discrete and continuous time Markov chains and that in the latter case our mathematical treatment works with the embedded discrete time process.

In what follows we are therefore interested in parametrised distributions and write $f(\cdot, \lambda)$, where $\lambda = \{\lambda_1, \ldots, \lambda_n\}$ is a vector of parameters, and distinguish different probability measures by their parameters. In particular, we denote by μ the original vector of parameters of the model and $f(\cdot, \mu)$ is therefore the original measure. We can thus rewrite (7) as

$$\gamma = \int_\Omega L(\omega) z(\omega) \, \mathrm{d}f(\omega, \lambda), \tag{10}$$

where $L(\omega) = \mathrm{d}f(\omega, \mu)/\mathrm{d}f(\omega, \lambda)$. We can also rewrite (9) as

$$\mathrm{d}f^* = \frac{z \, \mathrm{d}f(\cdot, \mu)}{\gamma} \tag{11}$$

and write $f(\cdot, \lambda^*)$ for the optimal parametrised measure. We define the optimal parametrised measure as that which minimises the *cross-entropy* [26] between $f(\cdot, \lambda)$ and f^* for a given parametrisation and note that, in general, $f^* \neq f(\cdot, \lambda^*)$.

4. The Cross-Entropy Method

Cross-entropy [26] (alternatively *relative entropy* or Kullback-Leibler divergence) has been shown to be a uniquely correct directed measure of distance between distributions [41]. With regard to the present context, it has also been shown to be useful in finding optimum distributions for importance sampling [28, 40, 31].

Given two probability measures f and f' over the same probability space Ω, the cross-entropy from f to f' is given by

$$\begin{aligned} \mathrm{CE}(f, f') &= \int_\Omega \log \frac{\mathrm{d}f(\omega)}{\mathrm{d}f'(\omega)} \, \mathrm{d}f(\omega) && (12) \\ &= \int_\Omega \log \mathrm{d}f(\omega) \, \mathrm{d}f(\omega) - \int_\Omega \log \mathrm{d}f'(\omega) \, \mathrm{d}f(\omega) \\ &= \mathrm{H}(f) - \int_\Omega \log \mathrm{d}f'(\omega) \, \mathrm{d}f(\omega), && (13) \end{aligned}$$

where $\mathrm{H}(f)$ is the entropy of f. To find λ^* we minimise $\mathrm{CE}(\frac{zf(\cdot,\mu)}{\gamma}, f(\cdot,\lambda))$, noting that $\mathrm{H}(f(\cdot,\mu))$ is independent of λ:

$$\lambda^* = \arg\max_\lambda \int_\Omega z(\omega) \log \mathrm{d}f(\omega, \lambda) \, \mathrm{d}f(\omega, \mu) \tag{14}$$

Estimating λ^* directly using (14) is difficult, so we re-write it using importance sampling measure $f(\cdot, \lambda')$ and likelihood ratio $L(\omega) = \mathrm{d}f(\omega, \mu)/\mathrm{d}f(\omega, \lambda')$:

$$\lambda^* = \arg\max_\lambda \int_\Omega z(\omega) L(\omega) \log \mathrm{d}f(\omega, \lambda) \, \mathrm{d}f(\omega, \lambda') \tag{15}$$

Using (15) we can construct an unbiased importance sampling estimator of λ^* and use it as the basis of an iterative process to obtain successively better estimates:

$$\tilde{\lambda^*} = \lambda^{(j+1)} = \arg\max_\lambda \sum_{i=1}^{N} z(\omega_i^{(j)}) L^{(j)}(\omega_i^{(j)}) \log \mathrm{d}f(\omega_i^{(j)}, \lambda) \tag{16}$$

N is the number of simulation runs generated on each of the j iterations, $\lambda^{(j)}$ is the j^{th} set of estimated parameters, $L^{(j)}(\omega) = \mathrm{d}f(\omega, \mu)/\mathrm{d}f(\omega, \lambda^{(j)})$ is the j^{th} likelihood ratio, $\omega_i^{(j)}$ is the i^{th} path generated using $f(\cdot, \lambda^{(j)})$ and $\mathrm{d}f(\omega_i^{(j)}, \lambda)$ is the probability of path $\omega_i^{(j)}$ under the distribution $f(\cdot, \lambda^{(j)})$.

5. Commanded-based Cross-Entropy Algorithm

We consider a system of n stochastic guarded commands with vector of rate functions $\eta = (\eta_1, \ldots, \eta_n)$ and corresponding vector of parameters $\lambda = (\lambda_1, \ldots, \lambda_n)$. We thus define n classes of transitions. In any given state x_s, reached after s transitions, the probability that command $k \in \{1 \ldots n\}$ is chosen is given by

$$\frac{\lambda_k \eta_k(x_s)}{\langle \eta(x_s), \lambda \rangle},$$

where η is explicitly parametrised by x_s to emphasise its state dependence and the notation $\langle \cdot, \cdot \rangle$ denotes a scalar product. For the purposes of simulation we consider a space of finite paths $\omega \in \Omega$. Let $\bigsqcup_{k=1}^{n} J_k(\omega) = \{0, \cdots, |\omega| - 1\}$ be

the disjoint union of sets such that each $J_k(\omega)$ contains the indices of states in which a type k transition occurred in path ω. Let $U_k(\omega)$ be the number of transitions of type k occurring in ω. Let $\bigsqcup_{k=1}^{n} J_k(\omega) = \{0, \cdots, |\omega| - 1\}$ be the disjoint union of sets such that each $J_k(\omega)$ contains the indices of states in which a type k transition occurred in path ω. We therefore have

$$\mathrm{d}f(\omega, \lambda) = \prod_{k}^{n} \left((\lambda_k)^{U_k(\omega)} \prod_{s \in J_k(\omega)} \frac{\eta_k(x_s)}{\langle \eta(x_s), \lambda \rangle} \right).$$

The likelihood ratios are thus of the form

$$L^{(j)}(\omega) = \prod_{k}^{n} \left(\left(\frac{\mu_k}{\lambda_k^{(j)}} \right)^{U_k(\omega)} \prod_{s \in J_k(\omega)} \frac{\langle \eta(x_s), \lambda^{(j)} \rangle}{\langle \eta(x_s), \mu \rangle} \right).$$

We define $\eta_k^{(i)}(x_s)$ and $\eta^{(i)}(x_s)$ as the respective values of η_k and η functions in state x_s of the i^{th} trace. We substitute the previous expressions in the cross-entropy estimator (16) and for compactness substitute $z_i = z(\omega_i)$, $J_k^{(i)} = J_k(\omega_i)$, $u_i(k) = U_k(\omega_i)$ and $l_i = L^{(j)}(\omega_i)$ to get

$$\begin{aligned} & \arg\max_{\lambda} \sum_{i=1}^{N} l_i z_i \log \prod_{k}^{n} \left(\lambda_k^{u_i(k)} \prod_{s \in J_k^{(i)}} \frac{\eta_k^{(i)}(x_s)}{\langle \eta^{(i)}(x_s), \lambda \rangle} \right) \\ = \quad & \arg\max_{\lambda} \sum_{i=1}^{N} \sum_{k}^{n} l_i z_i \\ & \left(u_i(k) \log(\lambda_k) + \sum_{s \in J_k^{(i)}} \log(\eta_k^{(i)}(x_s)) - \sum_{s \in J_k^{(i)}} \log(\langle \eta^{(i)}(x_s), \lambda \rangle) \right) \end{aligned} \tag{17}$$

We denote the argument of the arg max in (17) as $F(\lambda)$ and derive the following partial differential equation:

$$\frac{\partial F}{\partial \lambda_k}(\lambda) = 0 \Leftrightarrow \sum_{i=1}^{N} l_i z_i \left(\frac{u_i(k)}{\lambda_k} - \sum_{s=1}^{|\omega_i|} \frac{\eta_k^{(i)}(x_s)}{\langle \eta^{(i)}(x_s), \lambda \rangle} \right) = 0 \tag{18}$$

The quantity $|\omega_i|$ is the length of path ω_i.

Theorem 1. *A solution of* (18) *is almost surely a unique maximum, up to a normalising scalar.*

Proof. Consider

$$F_i(\lambda) = \sum_{k=1}^{n} \left(u_i(k)\ \log(\lambda_k) + \sum_{s \in J_k^{(i)}} \log(\eta_k^{(i)}(x_s)) - \sum_{s \in J_k^{(i)}} \log(\langle \eta^{(i)}(x_s), \lambda \rangle) \right)$$

and each element of this sum, $F_{i,k}(\lambda)$. Thus, note that $F_i(\lambda) = \sum_{k=1}^{n} F_{i,k}(\lambda)$ and $F(\lambda) = \sum_{i=1}^{N} l_i z_i F_i(\lambda)$. Note that for the sake of simplicity, we occasionally omit index i in the notations.

Using a standard result, it is sufficient to show that the Hessian matrix H_i of F_i in λ is negative semi-definite.

Hessian matrix H_i of F_i in λ is of the following form, with $v_k^{(s)} = \frac{\eta_k(x_s)}{\langle \eta(x_s), \lambda \rangle}$ and $v_k = (v_k^{(s)})_{1 \leq s \leq |\omega|}$:

$$H_i = G - D$$

$G = (g_{kk'})_{1 \leq k,k' \leq n}$ is the following Gram matrix:

$$g_{kk'} = \langle v_k, v_{k'} \rangle$$

D is a diagonal matrix, such that

$$d_{kk} = \frac{U_k(\omega)}{\lambda_k^2}.$$

Note that $U_k(\omega)$ is the number of times a transition of type k has been chosen over $|\omega| - 1$ transitions. On average, $U_k(\omega)$ is equal to the sum of probabilities of choosing transition k in each state x_s. Thus, asymptotically, $d_{kk} = \frac{1}{\lambda_k} \sum_{s=1}^{|\omega|} v_k^{(s)}$. We write $\mathbf{1}_{|\omega|} = (1, \ldots, 1)$ for the vector of $|\omega|$ elements 1, hence

$$d_{kk} = \frac{1}{\lambda_k} \langle v_k, \mathbf{1}_{|\omega|} \rangle.$$

Furthermore, $\forall s, \sum_{k=1}^{n} \lambda_k v_k^{(s)} = 1$. So, $\sum_{k'=1}^{n} \lambda_{k'} v_{k'} = \mathbf{1}_{|\omega|}$. Finally,

$$d_{kk} = \sum_{k'=1}^{n} \frac{\lambda_{k'}}{\lambda_k} \langle v_k, v_{k'} \rangle.$$

Let x be a non-zero vector in $\mathbb{R}^n$. To prove the theorem we need to show that $-x^t H_i x \geq 0$.

$$\begin{aligned}
-x^t H_i x &= x^t D x - x^t G x \\
&= \sum_{k,k'} \frac{\lambda_{k'}}{\lambda_k} \langle v_k, v_{k'} \rangle x_k^2 - \sum_{k,k'} \langle v_k, v_{k'} \rangle x_k x_{k'} \\
&= \sum_{k<k'} \left(\left[\frac{\lambda_{k'}}{\lambda_k} x_k^2 + \frac{\lambda_k}{\lambda_{k'}} x_{k'}^2 - 2 x_k x_{k'} \right] \langle v_k, v_{k'} \rangle \right) \\
&= \sum_{k<k'} \left(\sqrt{\frac{\lambda_{k'}}{\lambda_k}} x_k - \sqrt{\frac{\lambda_k}{\lambda_{k'}}} x_{k'} \right)^2 \langle v_k, v_{k'} \rangle \\
&\geq 0
\end{aligned}$$

The Hessian matrix H of F is of the general form

$$H = \sum_{i=1}^{N} l_i z_i H_i,$$

which is a positively weighted sum of non-positive matrices.

Moreover, for all $\lambda \in \mathbb{R}^n_{\geq 0}$,

$$\left(x^t H x = 0\right) \Leftrightarrow \left(\forall k \forall k' > k,\ x_k \neq 0 \wedge \frac{\lambda_{k'}}{\lambda_k} = \frac{x_{k'}}{x_k}\right) \Leftrightarrow \left(\exists r \in \mathbb{R}_{\neq 0},\ x = r\lambda\right). \tag{19}$$

This is because for all $\lambda \in \mathbb{R}^n$, $F(\lambda) = F(r\lambda)$ for all $r \in \mathbb{R}_{\neq 0}$. Geometrically, it means that the function is flat along a line generated by a vector λ. If λ was a solution of (18) then $r\lambda, r \in \mathbb{R}_{\geq 0}$ would also be a solution.

Assume now that there exists two non-collinear vectors, λ and μ, which are solutions of (18). By concavity of H, these two vectors are global maxima of F, implying that F is a constant over the cone generated by vectors λ and μ. In particular, function F would be constant along the line segment $\alpha\lambda + (1-\alpha)\mu$, with $\alpha \in [0,1]$. Let $y \in \mathbb{R}^n$ be the direction vector of the line containing this segment and ν an element in the interior of this segment. Denoting by $H(\nu)$ the Hessian of F at point ν, $y^t H(\nu) y = 0$. But y is not collinear to vector ν, which contradicts hypothesis (19).

A solution λ^* of (18) is thus a unique maximum up to a linear constraint over its norm. □

The fact that there is a unique optimum makes it conceivable to find λ^* using standard optimising techniques such as Newton and quasi-Newton methods. To do so would require introducing a suitable normalising constraint in order to force the Hessian to be negative definite. In the case of the cross-entropy algorithm of [31], this constraint is inherent because it works at the level of individual transition probabilities that sum to 1 in each state. We note here that in the case that our parameters apply to individual transitions, such that one parameter corresponds to exactly one transition, (22) may be transformed to Equation (9) of [31] by constraining in every visited state x, $\langle \eta(x), \lambda \rangle = 1$. Equation (9) of [31] has been shown in [42] to converge to f^* (for simple unbounded reachability), implying that under these circumstances $f(\cdot, \lambda^*) = f^*$ and that it may be possible to improve our parametrised importance sampling distribution by increasing the number of parameters. We illustrate this phenomenon in Fig. 3.

Equation (18) leads to the following expression for λ_k:

$$\lambda_k = \frac{\sum_{i=1}^{N} l_i z_i u_i(k)}{\sum_{i=1}^{N} l_i z_i \sum_{s=1}^{|\omega_i|} \frac{\eta_k^{(i)}(x_s)}{\langle \eta^{(i)}(x_s), \lambda \rangle}} \tag{20}$$

In this form the expression is not useful because the right hand side is dependent on λ_k in the scalar product. Hence, in contrast to update formulae based

on unbiased estimators, as given by (16) and in [31, 40], we construct an iterative process based on a biased estimator, but having a fixed point that is the optimum:

$$\lambda_k^{(j+1)} = \frac{\sum_{i=1}^{N} l_i z_i u_i(k)}{\sum_{i=1}^{N} l_i z_i \sum_{s=1}^{|\omega_i|} \frac{\eta_k^{(i)}(x_s)}{\langle \eta^{(i)}(x_s), \lambda^{(j)} \rangle}}. \tag{21}$$

Equation (21) is the basis of Algorithm 1 and can be seen as an implementation of (20) that uses the previous estimate of λ in the scalar product. As a result, in contrast to previous applications of the cross-entropy method, (21) converges by reducing the distance between successive distributions, rather than by explicitly reducing the distance from the optimum.

5.1. Smoothing

It is conceivable that certain guarded commands play no part in traces that satisfy the property, in which case (21) would make the corresponding parameter zero with no adverse effects. It is also conceivable that an important command is not seen on a particular iteration, but making its parameter zero would prevent it being seen on any subsequent iteration. To avoid this it is necessary to adopt a 'smoothing' strategy [31] that reduces the significance of an unseen command without setting it to zero. Smoothing therefore acts to preserve important but as yet unseen parameters. It is of increasing importance as the parametrisation gets closer to the level of individual transition probabilities, since only a tiny proportion of possible transitions are usually seen on any simulation run. Typical strategies include adding a small fraction of the original parameters, or a fraction of the parameters from the previous iteration, to the new parameter estimate. With smoothing parameter $\alpha \in]0, 1[$, these two strategies can be summarised as follows:

- Weighting with the original parameters:

$$\lambda_k^{(j+1)} = \alpha \mu_k + (1 - \alpha) \frac{\sum_{i=1}^{N} l_i z_i u_i(k)}{\sum_{i=1}^{N} l_i z_i \sum_{s=1}^{|\omega_i|} \frac{\eta_k^{i}(x_s)}{\langle \eta^{i}(x_s), \lambda^{(j)} \rangle}} \tag{22}$$

- Weighting with the previous parameters:

$$\lambda_k^{(j+1)} = \alpha \lambda_k^{(j)} + (1 - \alpha) \frac{\sum_{i=1}^{N} l_i z_i u_i(k)}{\sum_{i=1}^{N} l_i z_i \sum_{s=1}^{|\omega_i|} \frac{\eta_k^{i}(x_s)}{\langle \eta^{i}(x_s), \lambda^{(j)} \rangle}} \tag{23}$$

We have found that our parametrisation is often insensitive to smoothing strategy because each parameter typically governs many transitions and most parameters are affected by each run. The smoothing strategy adopted in the case studies described below is to multiply the parameter of unseen commands by 0.95. The effects of this can be seen clearly in Fig. 12. Whatever the strategy, since the parameters are unconstrained it is advisable to normalise them after each iteration (i.e., $\sum_k \lambda_k = \text{const.}$), in order to judge convergence.

Algorithm 1: Cross-Entropy Algorithm for Parametrised Commands

Data:
μ: the original parameters
$\lambda^{(0)}$: the initial parameters
N: the number of paths per iteration

$j = 0$
while $\lambda^{(j)}$ *have not converged and* $j < j_{\max}$ (see § 5.4) **do**
  $A = \vec{0}$
  $B = 0$
  $S = 0$
  for $i \in \{1, \ldots, N\}$ **do**
    $\omega_i = x_0$
    $l_i = 1$
    $\vec{u}_i = \vec{0}$
    $S = 0$
    $s = 1$
    while $\omega_i \models \phi$ *is not decided* **do**
      generate x_s under measure $f(., \lambda^{(j)})$
      $\omega_i = x_0 \cdots x_s$
      $l_i \leftarrow l_i \times \frac{\mu(x_{s-1}, x_s)\langle \eta^i(x_{s-1}), \lambda^{(j)} \rangle}{\lambda(x_{s-1}, x_s)\langle \eta^i(x_{s-1}), \lambda^{(j)} \rangle}$
      update $\vec{u_i}$
      $S \leftarrow S + \frac{\eta^i_k(x_s)}{\langle \eta^i(x_s), \lambda^{(j)} \rangle}$
      $s \leftarrow s + 1$
    $z_i = \mathbb{1}(\omega_i \models \phi)$
    $A \leftarrow A + l_i z_i \vec{u_i}$
    $B \leftarrow B + l_i z_i S$
    $\lambda^{(j+1)}_k = \frac{A_k}{B}$
    $\lambda^{(j+1)} \leftarrow \frac{\lambda^{(j+1)}}{\|\lambda^{(j+1)}\|}$
    smoothing of $\lambda^{(j+1)}$
  $j \leftarrow j + 1$
$\lambda^* \leftarrow \lambda^{(j-1)}$

5.2. Convergence

Theorem 1 proves that there is a unique optimum (λ^*) of (18), which is therefore the unique solution of (20). Equation (21) differs from (20) only in the iteration index of the parameters, hence any fixed point of (21) is also a solution of (20). Since (20) has a unique solution, (21) has a unique fixed point that is the optimum and we conclude that if Algorithm 1 converges, it must converge to the unique optimum. We do not provide a formal proof of convergence here, but note that we have never observed divergent or chaotic behaviour in practice. The algorithm is guaranteed to terminate with probability 1 by simply bounding the maximum number of iterations ($j_{\max}$). The number of samples per iteration is necessarily finite, so convergence is probabilistic and not necessarily monotonic. We typically observe rapid initial convergence that slows to stochastic fluctuations as the parameters approach their optimum values.

The inclusion of smoothing in the algorithm is a practical measure to prevent parameters being rejected prematurely when using finite numbers of simulations. Smoothing may have the undesirable side effect of slowing convergence and, when using (22), may prevent the algorithm from reaching the theoretical optimum. E.g., if the optimal value of λ_k is ≈ 0, (22) will nevertheless set $\lambda_l = \alpha\mu_k$. In practice, however, the smoothing strategy is chosen to avoid problems and have insignificant effect on the final distribution.

Given an adequate initial distribution and sufficient successful traces from the first iteration, (22) and (23) should provide a better set of parameters. In practice we have found that a single successful trace is often sufficient to initiate convergence. This is in part due to the existence of a unique optimum and partly to the fact that each parameter controls a command that usually governs a large number of semantically-linked transitions. The expected behaviour is that on successive iterations the number of traces that satisfy the property increases, however it is important to note that the algorithm minimises the cross-entropy and that the number of traces that satisfy the property is merely emergent of that. As has been noted, in general $f(\cdot, \lambda^*) \neq f^*$, hence it is likely that fewer than 100% of traces will satisfy the property when simulating under $f(\cdot, \lambda^*)$. One consequence of this is that an initial set of parameters may produce more traces that satisfy the property than the final set (see, e.g., Figs. 2 and 10).

Once the parameters have converged it is then possible to perform a final set of simulations to estimate the probability of the rare property. Algorithm 4 describes this process. The usual assumption is that $N \ll N_{\text{IS}} \ll N_{\text{MC}}$, however it is often the case that parameters converge fast, so it is expedient to use some of the simulation runs generated during the course of the optimisation (i.e., Algorithm 1) as part of the final estimation.

5.3. Initial Distribution

Algorithm 1 requires an initial simulation distribution ($f(\cdot, \lambda^{(0)})$) that produces at least a few traces that satisfy the property using N simulation runs. Finding $f(\cdot, \lambda^{(0)})$ for an arbitrary model may seem to be an equivalently difficult problem to estimating γ, but this is not in general the case. When a

property (e.g., failure of the system) is semantically linked to an explicit feature of the model (e.g, a command for component failure), good initial parameters may be found relatively easily by heuristic methods such as failure biasing [29]. Alternatively, if the model and property are similar to a previous combination for which parameters were found, those parameters are likely to provide a good initial distribution.

Increasing the parameters associated to commands with obviously small rates may help, along the lines of failure biasing. It is also possible during simulation to make every transition from any given state have equal probability. In this case parameters are state dependent and calculated on the fly, while the occurrence of commands may still be counted to infer static importance sampling parameters. Note, however, that the rareness of a property expressed in temporal logic may not be related to low transition probabilities. That is, the rareness of a trace (a specific sequence of states) in trace space does not necessarily imply that its transition probabilities are low.

A further important observation is that the rareness of the property in trace space does not imply that good parameters are rare in parameter space. Consequently, a random search of parameter space often requires many orders of magnitude fewer attempts to find an example of the rare property than the expected number under the original distribution (i.e., $1/\gamma$). This phenomenon is the basis of the algorithmic approaches to finding initial distributions given below.

Figure 2 illustrates the parameter space of the chemical model described in Section 6.3. Although the majority of parameters, including those which generate the original distribution (red dot), fall into a region where the probability of satisfying the property is near zero (upper triangle), a significant region of the parameter space ($\approx 37\%$) gives near 100% success (lower triangle). A narrow strip between these two regions (indicated by a grey line in Fig. 2) contains parameters with intermediate levels of success, among which is the unique vector of parameters for minimum cross-entropy. The figure also shows how two different initial parameter vectors converge to the optimum. Although time cannot be discerned from the figure, in both cases the algorithm converges rapidly to the intermediate region, which tends to contain distributions with relatively low cross-entropy. The algorithm then converges more slowly to the point of minimum cross-entropy. In the figure the point is marked by a single blue dot, but since convergence is statistical, the two end points are close but not exactly the same.

5.3.1. Algorithms for Initial Distributions

An effective strategy to find initial distributions is to simulate with random parameters until a trace satisfying the rare property is observed – the parameters used to generate the trace become $\lambda^{(0)}$. Note, however, that the observation of such a trace does not imply that the parameters will necessarily generate satisfying traces with high probability. Choosing parameters in this way is effectively drawing from the joint distribution of parameters and simulation traces, hence an individual success may also indicate a high density of

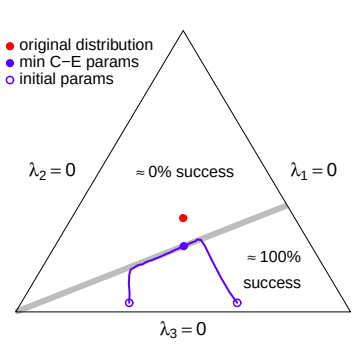

Figure 2: Parameter simplex of three parameter chemical model.

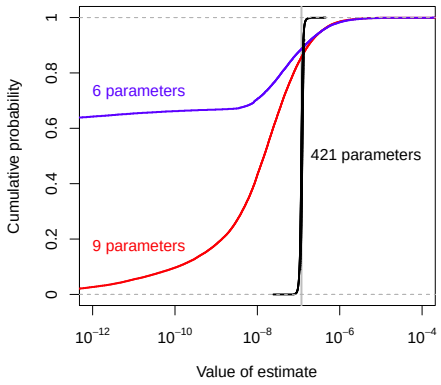

Figure 3: Importance sampling with increasing numbers of parameters.

parameters with relatively low probability of satisfying the property. To account for these eventualities, we give two algorithms.

Algorithms 2 and 3 select parameters uniformly at random by sampling from a Dirichlet distribution with vector of parameters $(1, \ldots, 1)$, denoted $\mathrm{Dir}(\mathbf{1})$. Algorithm 2 (Optimistic) assumes that the density of parameters is more or less uniform, such that any increase in probability of observing a successful trace is only due to parameters being good. Algorithm 3 (Pessimistic) generalises Algorithm 2 and allows parameters to be judged on their actual performance, rather than relying on the assumption of uniformity. The algorithm performs $1 < N \ll 1/\gamma$ simulation runs per randomly chosen parameter set. The values of N and θ are chosen according to the degree of "pessimism". N is typically no more than the per-iteration N in Algorithm 1. The value of θ is chosen such that $\lceil N\theta \rceil \geq 1$ is a realistic expectation of the number of successes.

It is easy to construct pathological examples for which no parameters exist that improve on the original distribution, although we have found this to be unusual with real case studies. If such a case does arise, however, it will be indicated by the number of iterations of Algorithms 2 and 3 performed without success. Under such circumstances, it is useful to note that the probability of success of both algorithms is exactly the probability of the rare event. It is therefore possible to use the reciprocal of the number of simulations as a crude upper bound of γ.

Algorithm 2: Optimistic

$z(\cdot) \leftarrow$model checking function
repeat
  sample $\lambda \sim \mathrm{Dir}(\mathbf{1})$
  generate trace $\omega \sim f(\cdot, \lambda)$
until $z(\omega) = 1$;
$\lambda^{(0)} \leftarrow \lambda$

Algorithm 3: Pessimistic

$N \leftarrow$runs per iteration
$\theta \in (0, 1] \leftarrow$acceptance threshold
$z(\cdot) \leftarrow$model checking function
repeat
  sample $\lambda \sim \mathrm{Dir}(\mathbf{1})$
  generate N traces $\omega_i \sim f(\cdot, \lambda)$
until $\sum_i^N z(\omega_i) > N\theta$;
$\lambda^{(0)} \leftarrow \lambda$

5.3.2. Division of Commands

Our approach exploits the symmetries created by the syntactical definition of the model, under the heuristic assumption that commands link transitions that often have an unambiguous semantic relevance to the property (e.g., individual commands that govern component failure and repair when considering the property of all components failing). Our guarded commands modelling language nevertheless allows the *guard*, *rate* and *action* to be complex conditional functions of state variables. When commands have "confused" semantics with respect to the property, parametrisation at the level of the original commands may be too crude. To increase the performance of our approach, it may thus be useful to divide each guarded command into a set of commands with less confused semantics. Without loss of expressiveness, we believe this can be partially achieved a priori by restricting the syntax of expressions (e.g., by not allowing conditional actions or rates). We also believe it may be possible to decompose guards automatically. For example, using standard techniques to factorise Boolean expressions, command (*guard*, *rate*, *action*) can be divided into (*guard′*, *rate*, *action*) and (*guard″*, *rate*, *action*), with *guard′*$\vee$ *guard″* $\equiv$ *guard*, *guard′*$\wedge$*guard″* $\equiv$ *false* and *rate*, *action* unchanged. Additionally, guards comprising inequalities of the form $x < u$, with x an integer state variable with lower bound l and u an integer upper bound, are simply divided as follows: $x < u \equiv \bigvee_{k=l}^{u-1} x = k$.

While the automated division of commands remains the subject of future work, we have used this idea to tweak the performance of some of our case studies and to demonstrate the existence of the optimal distribution in a repair model (see Fig. 3).

5.4. The Rare Event Simulation Process

In summary, we run Algorithm 1 to find optimal importance sampling parameters with respect to property φ, then run Algorithm 4 to estimate γ, the probability of φ.

Algorithm 1 is supplied with an initial set of parameters $\lambda^{(0)}$ that is generated using one of the procedures described in Section 5.3. The outer 'while' loop (line 2) corresponds to the cross-entropy minimisation. This loop terminates when parameters $\lambda^{(j)}$ converge or when the upper bound of iterations ($j_{\max}$) is reached. A convergence criterion can be satisfied, for example, whenever

Algorithm 4: Importance sampling by $f(.,\lambda^*)$

Data:
μ: the original parameters
λ^*: the optimal parameters computed by Algorithm 1
N_{IS}: the number of paths
for $i \in \{1,\ldots,N_{IS}\}$ **do**
  $\omega_i = x_0$
  $l_i = 1$
  $s = 1$
  while $\omega_i \models \phi$ *is not decided* **do**
    generate x_s under measure $f(.,\lambda^*)$
    $\omega_i = x_0 \cdots x_s$
    $l_i \leftarrow l_i \times \frac{\mu(x_{s-1},x_s)\langle\eta^i(x_{s-1}),\lambda^{(j)}\rangle}{\lambda(x_{s-1},x_s)\langle\eta^i(x_{s-1}),\lambda^{(j)}\rangle}$
    $s = s + 1$
  $z_i = \mathbb{1}(\omega_i \models \phi)$
$\tilde{\gamma} = \frac{1}{N_{IS}} \sum_{i=1}^{N_{IS}} z_i l_i$

$\max_{0\le k\neq l\le 2} \|\lambda^{(j-k)} - \lambda^{(j-l)}\| \le \epsilon$. Note, however, that to facilitate comparisons we simply bound the number of iterations to generate the experimental results reported in Section 6. On line 12, the inner 'while' loop is the path generator, in which likelihood ratio l_i is updated on the fly. On line 16, each time a transition of type k is taken, the corresponding coordinate of u_i is incremented by 1. Line 23 corresponds to the normalisation of $\lambda^{(j)}$. On line 24, parameter $\lambda^{(j)}$ is smoothed by a strategy described in Section 5.1. The resulting parameters are used to generate the new samples.

Algorithm 4 essentially implements the path generating loop of Algorithm 1, but without counting the occurrence of commands.

6. Case Studies

The following case studies are included to illustrate the performance of our algorithms and parametrisation. The principal motivation of statistical techniques is to address intractable state space, however the trade-offs between numerical and standard Monte Carlo in this regard are well understood. The particular challenge for our approach is to show that its parametrisation is able to generate good importance sampling distributions. Hence, with the exception of the chemical system, we have chosen models for which we are able to obtain accurate results using numerical techniques, in order to compare the estimates produced by our algorithms with the nominally correct values.

The first case study is a repair model, the second a standard queueing network and the third is an example of a chemically reactive system. In the two first cases, initial distributions are created on the fly by assigning equal probability to all enabled transitions from a state. In the third case, an initial distribution

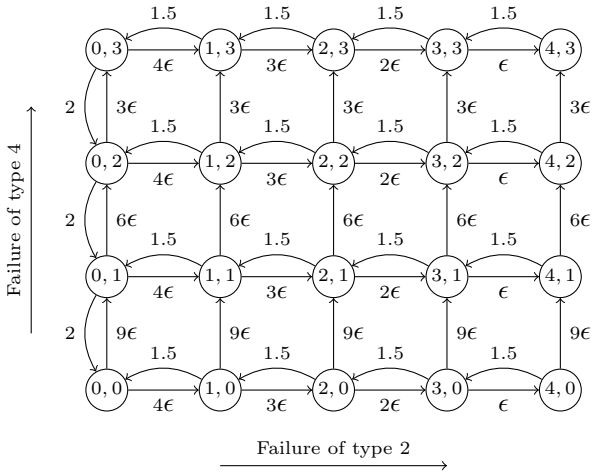

Figure 4: Relationship between failed components of type 2 and type 4 in the repair model. Node labelling gives $\#type2,\#type4$. Edgle labelling gives rates of transitions.

is found using Algorithm 2, with fewer than 500 iterations. This value is less than N and considerably less than $1/\gamma$. All simulations were performed using our statistical model checking platform, Plasma [17].

6.1. Repair Model

The need to certify system reliability often motivates the use of formal methods and thus reliability models are studied extensively in the literature. The following example is taken from [31] and features a moderately large state space of $40,320$ states, which can be investigated using numerical methods to corroborate our results.

The system is modelled as a continuous time Markov chain and comprises six types of subsystems $(1,\ldots,6)$ containing, respectively, $(5,4,6,3,7,5)$ components that may fail independently. We denote by $n_u(k)$ the number of components of type k. The system's evolution begins with no failures and with various stochastic rates the components fail and are repaired. The subsystem failure rates are $(2.5\epsilon,\epsilon,5\epsilon,3\epsilon,\epsilon,5\epsilon)$, $\epsilon = 0.001$, and the repair rates are $(1.0,1.5,1.0,2.0,1.0,1.5)$, respectively. In addition, components are repaired with priority according to their type. Each subsystem type is modelled by two guarded commands, one for failure and one for repair, using a single variable to count the number of failed components. Figure 4 illustrates the relationship between failed components of type 2 and type 4. For simplicity, the other subsystems are not shown.

The property we consider is the probability of a complete failure of a subsystem (i.e., the failure of all components of one type), given an initial condition of no failures. This can be expressed in temporal logic as $\mathrm{P}[\mathbf{X}(\neg \mathit{init}\,\mathbf{U}^{1000}\,\mathit{failure})]$. The true probability, calculated numerically, is $\gamma = 7.488 \times 10^{-7}$.

If only one component in the whole system has failed, its immediate repair would violate the property. With this knowledge we are able to make an a priori judicious division of each subsystem's command for repair. We thus discriminate three cases: (*i*) one component of the subsystem has failed and none of the others; (*ii*) one component of the subsystem and at least one of another type have failed; and (*iii*) at least two components of the same subsystem have failed. Note that the repair command of the components of lower repair priority is just divided into cases (*i*) and (*iii*), as the second case cannot occur. From an initial 12-command model (six subsystems comprising one command for failure and one command for repair), we constructed a semantically equivalent model of 23 commands (five subsystems comprising one command for failure and three commands for repair and one subsystem comprising one command for failure and two commands for repair). Finally, for each type k, the failure command guard enables the corresponding transitions only if the maximal number of subsystem components $n_u(k)$ have not already failed. We split the guard into $n_u(k)$ guards, each of them enabling the transition for a particular number of failed type-k components between 0 and $n_u(k) - 1$. This model contains 47 commands. In what follows, the three models are denoted $\mathrm{r}(i, j)$, with i the number of modules and j the number of commands.

We applied Algorithm 1 to our models, in each case starting from an initial distribution that assigns equal probability to the enabled transitions of any state. We set $N = 10000$ simulations for each of the 50 cross-entropy iterations, noting that convergence was usually observed in fewer than 20 iterations. The final iteration was used as the importance sampling estimator.

To empirically verify our results we performed each simulation experiment 100 times. Note that the importance sampling estimators are based on $N_{\mathrm{IS}} = 10000$ traces but required a total of 500000 samples. We therefore compare the results of importance sampling with the theoretical values of crude Monte Carlo experiments based on $N_{\mathrm{MC}} = 500000$ samples.

We make use of the concept of skewness, which measures the asymmetry of a distribution X. An unbiased estimator is given by

$$\mathrm{skew}(X) = \frac{n}{(n-1)(n-2)} \frac{\sum_{k=1}^{n} (X_k - \bar{x})^3}{(\hat{\sigma^2})^{3/2}}, \tag{24}$$

with $\bar{x}$ and $\hat{\sigma^2}$ the usual unbiased mean and variance estimates.

A skewness close to zero indicates that the estimators are distributed evenly around the mean. A negative value indicates that the distribution is left-tailed (the mass of the estimators is concentrated on the right), while a positive value indicates that the distribution is right-tailed.

For each estimator, we produced a standard approximate 95%-confidence interval $CI(\hat{\gamma}_n) = [0.99\,(\hat{\gamma}_n - 1.96\hat{\sigma}_n/\sqrt{n})\,; 1.01\,(\hat{\gamma}_n + 1.96\hat{\sigma}_n/\sqrt{n})]$, with $\hat{\sigma}_n$ the usual unbiased sample standard deviation estimate. In Table 1 we report the mean value of $\hat{\sigma}_n$ over the 100 experiments in the line labelled σ_n. This value is to be compared with the standard deviation of Bernoulli distribution z, as described in Section 3. As the length of the confidence interval is proportional

experiment	r(6,12)	r(6,23)	r(6,47)	MC
mean	7.45×10^{-7}	7.45×10^{-7}	7.46×10^{-7}	7.49×10^{-7}
std. dev.	6.39×10^{-9}	5.89×10^{-9}	3.66×10^{-9}	1.22×10^{-6}
skewness	0.03	-0.02	0.26	1.63
σ_n	7.09×10^{-7}	5.91×10^{-7}	3.28×10^{-7}	8.65×10^{-4}
coverage	100%	100%	100%	31%

Table 1: Results of repair model

to σ_n, low $\hat{\sigma}_n$ values imply a narrower confidence interval centred around mean value $\hat{\gamma}_n$. However, $\hat{\gamma}_n$ remains an estimation of γ_n. Using values 0.99 and 1.01 is a classic technique to slightly enlarge the approximate confidence interval and increase the chance to strictly fulfil $P(\gamma \in CI(\hat{\gamma}_n)) \geq 0.95$. The coverage is the percentage of approximate confidence intervals that contained the exact value γ. The expected coverage value is thus expected to be greater than 95%.

In Table 1 we also indicate for the three models the empirical mean, standard deviation, skewness and coverage of the estimates in columns r(6,12), r(6,23) and r(6,47). For comparison, the notional values for Monte Carlo estimates based on $N_{\mathrm{MC}} = 500000$ sample size are given in column MC.

We can see that for all models the cross-entropy algorithm gives a very accurate estimate, with variance decreasing with increasing numbers of commands. The skewness shows that the estimates are evenly distributed about the mean. Last but not least, the confidence intervals always contain the exact value and the length of the confidence intervals is maintained narrow due to low σ_n values. In contrast, since most of the Monte Carlo estimates are equal to 0, their underlying confidence interval is reduced to 0 at 69%. With probability 31%, they are equal to $1/(5 \times 10^{-6})$ and contain γ but at the price of a very large width.

Figure 12 shows the convergence of parameters for a particular experiment with r(6,12) and highlights the effects of the adopted smoothing strategy. While most parameters converge to stable values, the parameters denoted by green and magenta lines (corresponding to repair of components of types 5 and 6, respectively) are continually attenuated by the smoothing factor (0.95 in this case). Their commands are not seen in successful traces, suggesting that they are less important than the other parameters with respect to the property. Most of the other parameters lie in the approximate range 0.01 to 0.25, but the parameters denoted by red and blue lines (corresponding to the failure and repair, respectively, of components of type 4) are significantly outside. It is clear that increasing the failure rate of components of type 4 is critical to the property. The fact that repair transitions are generally made less likely by the algorithm agrees with the intuition that we are interested in direct paths to failure. The fact that they are not necessarily made zero reinforces the point

that the algorithm seeks to consider *all* paths to failure, including those that have intermediate repairs.

Figure 5 plots the number of paths satisfying $\mathbf{X}(\neg init \mathbf{U}^{1000} failure)$ and suggests that for this model the parametrised distribution is close to the optimum. Figure 6 plots the estimated probability and sample standard deviation during the course of the algorithm, superimposed on the probability calculated by numerical model checking (horizontal line). The long term average agrees well with the true value (an error of -0.5%, based on an average excluding the first estimate), justifying our use of the sample standard deviation as an indication of the efficacy of the algorithm: our importance sampling parameters provide a variance reduction of more than 10^6 with respect to the variance of the Bernoulli random variable z ($\approx \gamma$).

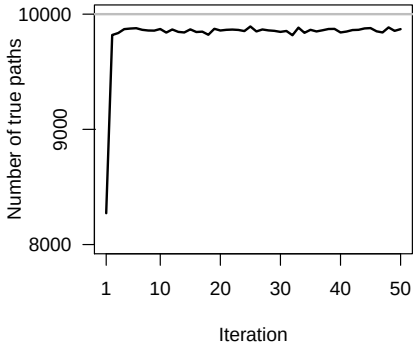

Figure 5: Convergence of number of paths satisfying $\mathbf{X}(\neg init \mathbf{U}^{1000} failure)$ in the repair model r(3,12) using $N = 10000$.

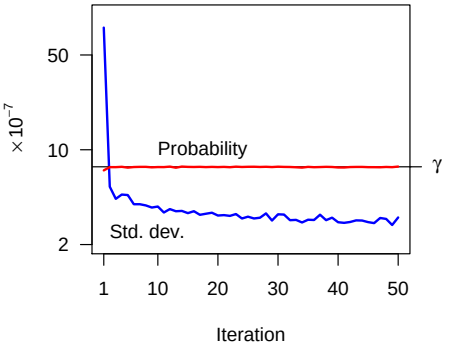

Figure 6: Convergence of empirical mean and standard deviation of likelihood ratio for repair model r(3,47) using $N = 10000$. γ indicates true probability.

6.2. Tandem Queueing Network

The following example is adapted from [43] and represents a queueing network of two queues of customers (M/Cox2/1-queue and M/M/1) composed sequentially. The system is modelled as a continuous time Markov chain and originally comprises two modules. We add a third passive module whose purpose is to count the number of steps.

Both queueing servers have capacity $c = 20$. The first desk receives customers with rate $\lambda = 4c$. With probability 0.1, the server of the first desk handles a customer's request in one phase with rate $\mu_1 = 2$; with probability 0.9, the server needs an extra phase to treat the request with rate $\mu_2 = 2$. In this case, an internal variable of the first module, ph is set to 2 instead of 1. Once served, customers leave the first desk and join the queue of the second desk where service occurs with rate $\kappa = 4$. If the second queue is full, the first desk is said to be blocked. In this case, the first desk can still pass a request

experiment	t(2,5)	t(2,65)	MC
mean	1.93×10^{-8}	1.9×10^{-8}	1.93×10^{-8}
std. dev.	2.1×10^{-9}	1.13×10^{-9}	1.8×10^{-7}
skewness	2.89	-1.18	9.28
σ_n	2.05×10^{-7}	6.8×10^{-8}	1.34×10^{-4}
coverage	97%	89%	$< 1.2\%$

Table 2: Results for tandem queue model.

from the first phase to the second phase but is then completely blocked while the second queue is full. This situation is called a *saturation*. Denoting by c_1 and c_2 the respective current number of customers at each desk, *saturation* is equivalent to $(c_1 = c) \wedge (c_2 = c) \wedge (ph = 2)$. The property of interest is the probability of saturation of the network within 20 steps, $\mathrm{P}[\mathbf{F}^{20} saturation)]$. The exact probability is $\gamma = 1.934 \times 10^{-8}$.

We performed 100 estimation experiments on each of two models of the above system; one containing 5 commands (denoted t(2,5)) and the other containing 65 commands (denoted t(2,65)). In each case we performed 30 iterations of Algorithm 1 with $N = 20000$ simulations per iteration, recording the statistics of the last iteration. Thus $N_{\mathrm{IS}} = 20000$. The results are given in Table 2, whose last column contains the theoretical values of crude Monte Carlo experiments based on 600000 samples, i.e., the total number of simulations used by the cross-entropy approach.

The results in Table 2 shows that our approach is able to make useful reductions in variance with respect to crude Monte Carlo, although the difference in performance between the two models is marginal. We hypothesise that this is due to keeping the number of cross-entropy iterations fixed in the two sets of experiments.

6.3. Chemical Network

There is an increasing expectation that formal methods can be applied to biological systems. The network of chemical reactions given below is abstract but typical of biochemical systems and demonstrates the potential of SMC to handle the enormous state spaces of biological models. In particular, we demonstrate the efficacy of our algorithm by applying it to quantify two rare dynamical properties of the system.

We consider a well stirred chemically reacting system comprising five reactants (molecules of type A, B, C, D and E), a dimerisation reaction and two decay reactions. We denote the instantaneous number of molecules of A, B, C, D and E by state variables A, B, C, D and E, respectively. The reactions are modelled by three guarded commands, having importance sampling parameters

λ_1, λ_2 and λ_3, respectively:

$$(A > 0 \wedge B > 0, \lambda_1 \times A \times B, A \leftarrow A - 1; B \leftarrow B - 1; C \leftarrow C + 1) \quad (25)$$

$$(C > 0, \lambda_2 \times C, C \leftarrow C - 1; D \leftarrow D + 1) \quad (26)$$

$$(D > 0, \lambda_3 \times D, D \leftarrow D - 1; E \leftarrow E + 1) \quad (27)$$

Under the assumption that the molecules move randomly and that elastic collisions significantly outnumber unreactive, inelastic collisions, the system may be simulated using mass action kinetics as a continuous time Markov chain [38]. The semantics of (25) is that if a molecule of type A encounters a molecule of type B they will combine to form a molecule of type C after a delay drawn from an exponential distribution with mean $\lambda_1 \times A \times B$. The decay reactions (26) and (27) have the semantics that a molecule of type C (D) spontaneously decays to a molecule of type D (E) after a delay drawn from an exponential distribution with mean $\lambda_2 \times C$ ($\lambda_3 \times D$). A typical simulation run is illustrated in Fig. 7, where the units of the x-axis are steps rather than time to aid clarity. A and B combine rapidly to form C, which peaks before decaying slowly to D. The production of D also peaks, while E rises monotonically.

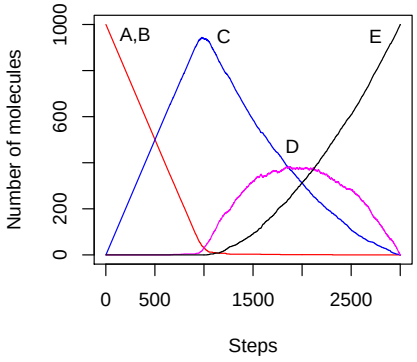

Figure 7: A typical stochastic simulation trace of reactions (25–27).

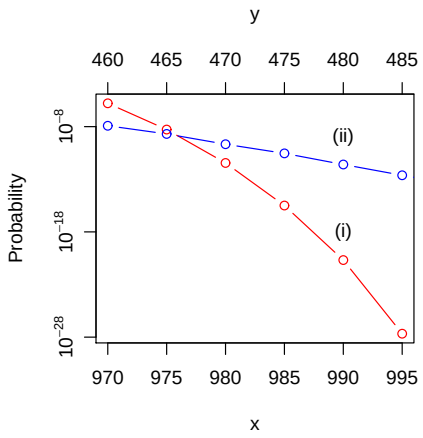

Figure 8: (i) $\mathrm{P}[\mathbf{F}^{3000}\, C \geq x]$ (ii) $\mathrm{P}[\mathbf{F}^{3000}\, D \geq y]$

With an initial vector of molecules $(1000, 1000, 0, 0, 0)$, corresponding to variables (A, B, C, D, E), the state space comprises approximately 1.6×10^8 states and 4.8×10^8 transitions. Although extremely simple in the context of typical biological systems, the model is intractable to numerical analysis. By inspection, we can infer that it is possible for the numbers of molecules of C and D to reach the initial number of A and B molecules (i.e., 1000) and that this is unlikely. To find out exactly how unlikely we consider the probabilities of the following rare properties defined in linear temporal logic: (i) $\mathbf{F}^{3000}\, C \geq x, x \in \{970, 975, 980, 985, 990, 995\}$ and (ii) $\mathbf{F}^{3000}\, D \geq y, y \in \{460, 465, 470, 475, 480, 485\}$. The results are plotted in Fig. 8.

mean	std. dev.	skewness	σ_n	coverage
1.502×10^{-10}	7.013×10^{-12}	0.9	1.895×10^{-9}	94%

Table 3: Results of chemical model experiment $c(100000)$.

Having found an initial set of parameters by the means described in Section 5.3, Algorithm 1 was iterated 100 times using $N \in \{500, 1000, 10000, 30000\}$. Despite the large state space, these values of N were found to be sufficient to produce reliable results (demonstrated in Table 3). The convergence of parameters for the property $\mathbf{F}^{3000}\, D \geq 470$ can be seen in Fig. 9. To two decimal places, the resulting optimal parameters are $\lambda^* \approx (1.14, 1.16, 0.70)$. For this particular estimation problem it seems that there is no advantage in setting N above 1000. In general, since we eventually expect the probability of seeing traces that satisfy the property to be close to 1, the value of N is typically chosen to ensure that we see a few successful traces in the early iterations.

Figure 10 illustrates that the number of paths satisfying a property can actually decrease as the quality of the distribution improves. Figure 11 illustrates the convergence of the estimate and the empirical variance of the likelihood ratio, using the importance sampling parameters generated during the course of running the algorithm. The initial set of parameters appear to give a very low variance, however this is clearly erroneous with respect to subsequent values. This is the situation illustrated in Fig. 1e.

To judge the quality of our parametrised importance sampling distribution, we performed 100 estimation experiments, each using 100000 traces. The results are summarised in Table 3. Although the exact value is unknown, the standard deviation and the skewness of the estimator distribution strongly suggest that the value is consistent with the mean $\gamma \approx 1.5 \times 10^{-10}$. The mean of the standard deviation of the true path likelihood ratios, denoted σ_n, is also very low in comparison to that of the Bernoulli random variable used in standard Monte Carlo. As the variance of a Bernoulli random variable is approximately equal to the probability in the context of rare events and assuming that the sample variance is close to the true variance, Fig. 11 suggests that we have made a variance reduction with respect to a Bernoulli distribution with parameter 1.502×10^{-10} of approximately 2×10^8. The half length of our importance confidence intervals was on average of 0.13×10^{-10}. If we consider that 1000 paths and 50 iterations were sufficient to get a good approximation of λ^* and that 100000 paths were necessary for our importance sampling estimator, we required a total of 150000 sampled paths. In comparison, given error $\epsilon = 0.13 \times 10^{-10}$ and confidence $1 - \alpha = 0.95$, achieving the same precision with the standard Chernoff bound [9] would require more than 10^{22} paths.

7. Existence of Distributions

The definition of the optimal distribution (9) is said in the literature to be not useful because it contains the quantity that is being sought, i.e., γ.

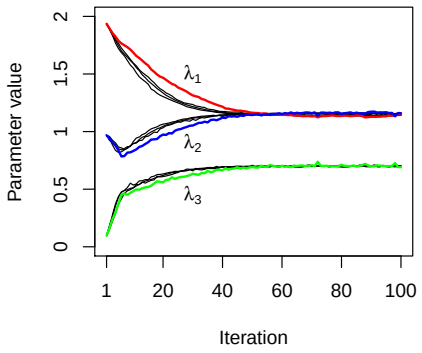

Figure 9: Convergence of parameters for $\mathbf{F}^{3000}$ $D \geq 470$ in the chemical model using $N \in \{500, 1000, 10000, 30000\}$. Heavy lines indicate $N = 500$.

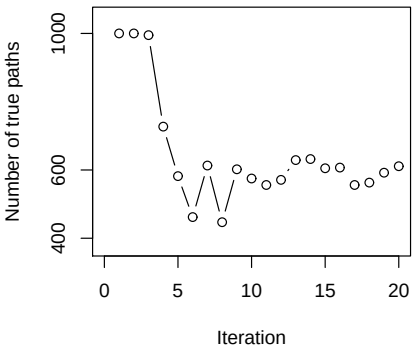

Figure 10: Convergence of number of paths satisfying $\mathbf{F}^{3000}$ $D \geq 470$ in the chemical model using $N = 1000$.

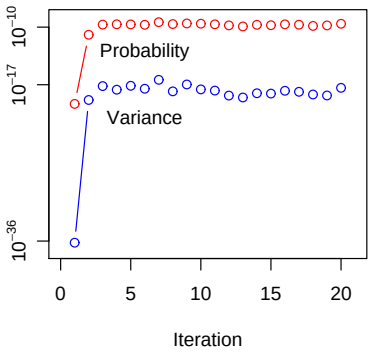

Figure 11: Convergence of empirical mean and variance of likelihood ratio for chemical model, using $\mathbf{F}^{3000}$ $D \geq 470$ and $N = 1000$.

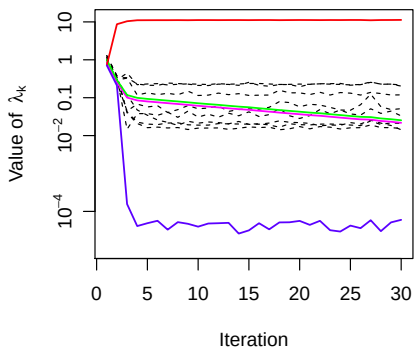

Figure 12: Convergence of parameters and effect of smoothing (green and magenta lines) in repair model using $N = 10000$.

If that were the only limitation, it would be possible to devise a sequence of approximations that converged to f^*. In fact, (9) is often not useful because f is generated implicitly by a transition system – f is merely notional, with no compact functional representation. A further problem in the context of general SMC, in contrast to simple reachability properties of reliability models, is that it may be impossible to generate f^* using only the states and transitions of f. In general, a command-based minimum cross-entropy importance sampling distribution will not achieve f^*, but in the following subsection we prove a number of theorems that define the conditions under which it may be possible to better approximate f^* by increasing the number of parameters. We note, however, that the non-existence of the optimal distribution in the family of distributions induced by the parametrisation does not imply poor performance.

Figure 3 illustrates the beneficial effects of increasing the number of parameters of a parametrised importance sampling distribution, considering a repair

model with group repair taken from [31] and property $\mathbf{X}(\neg \mathit{init}\mathbf{U}^{1000}\mathit{failure})$. In this case the optimal distribution can be generated by a re-parametrisation of the original transitions. The system comprises three types of component, hence the simplest model contains just six parameters, corresponding to the commands for failure and repair of each component. Repair transitions occur in most successful traces, but a significant number of traces fail to satisfy the property by prematurely returning to the initial state. Hence, the eleven parameter model isolates the repair transitions that lead directly to *init*, allowing them to be set to near zero without affecting other repairs. By allocating a parameter to every transition probability (the 421 parameter model), Algorithm 1 is able to converge to the optimal distribution.

7.1. Existence Theorems

Let $(s_i)_{i\geq 0}$ be a discrete-time Markov chain with state space S and initial state s_0. A path ω of the Markov chain is a sequence of transitions from one state to the next. We assume the chain evolves from state s_0 until property φ is decided and that the probability of reaching a decision using a finite number of transitions is equal to 1. The number of transitions until φ is decided is denoted $\tau = |\omega|$ and, by definition, $0 < \tau < \infty$. The set of all paths ω starting in s_0 is denoted Ω. We denote by T the set of transitions between states and by P the transition probability matrix. We write $P(s_{i-1}, s_i)$ to denote the transition probability from state s_{i-1} to s_i. A transition between state s_i and s_j is denoted $s_i \to s_j$.

Let $\gamma : S \to [0, 1]$ be a function giving the probability that an arbitrary (finite) path starting from state s satisfies temporal property φ. The probability of satisfying φ from the initial state s_0 is simply denoted γ. Note that we check path transitions and not states. Hence, if path ω is decided after reaching state s, we set $\gamma(s) = 0$ because no transitions are available.

Let $z : \Omega \to \{0, 1\}$ be the model checking function $\mathbb{1}(\omega \models \varphi)$. As traces are stochastic realisations of the system, the behaviour of function z is modelled as a Bernoulli random variable Z. By definition, $\gamma = \mathrm{E}[Z]$.

Let P be absolutely continuous with respect to a transition probability matrix Q. The likelihood ratio of path ω is given by

$$L(\omega) = \prod_{i=1}^{\tau} \frac{P(s_{i-1}, s_i)}{Q(s_{i-1}, s_i)}.$$

Under the measure induced by Q,

$$\gamma = \mathrm{E}_Q[ZL].$$

Theorem 2. *Given a temporal property φ, if there exists a function $c : T \to \{0, 1\}$ such that $z(\omega) = c(s_{\tau-1} \to s_\tau)$, there exists an importance sampling estimator of γ with zero-variance in this setting.*

Proof. Let $c : T \to \{0, 1\}$ such that $c(s_i \to s_j) = 1$ if $(s_i \to s_j) \models \varphi$ and 0 otherwise.

Model checking proceeds by generating a transition and checking whether the property is decided. If the property is decided, the process is halted and the result is returned. If not, a new transition is generated. Hence, the property is decided on the last transition of a trace, i.e., $z(\omega) = c(s_{\tau-1} \to s_\tau)$.

So, by construction,

$$Z = \sum_{i=1}^{\tau} c(s_{i-1} \to s_i) = c(s_{\tau-1} \to s_\tau)$$

Note that Z only depends on the value of the last transition $c(s_{\tau-1} \to s_\tau)$, since no further transitions will influence the result. Furthermore, c is equal to 0 all along the path and is equal to 0 or 1 at the last step.

We rewrite random variable ZL as follows:

$$ZL = c(s_{\tau-1} \to s_\tau) \prod_{i=1}^{\tau} \frac{P(s_{i-1}, s_i)}{Q(s_{i-1}, s_i)}$$

Consider $Q(s_{i-1}, s_i)$ proportional to $P(s_{i-1}, s_i)(c(s_{i-1} \to s_i) + \gamma(s_i))$. In this case,

$$\begin{aligned} Q(s_{i-1}, s_i) &= \frac{P(s_{i-1}, s_i)(c(s_{i-1} \to s_i) + \gamma(s_i))}{\sum_{s' \in S} P(s_{i-1}, s')(c(s_{i-1} \to s') + \gamma(s'))} \\ &= \frac{P(s_{i-1}, s_i)(c(s_{i-1} \to s_i) + \gamma(s_i))}{\gamma(s_{i-1})} \end{aligned}$$

Then,

$$\begin{aligned} ZL &= c(s_{\tau-1} \to s_\tau) \prod_{i=1}^{\tau} \frac{P(s_{i-1}, s_i)}{Q(s_{i-1}, s_i)} \\ &= c(s_{\tau-1} \to s_\tau) \prod_{i=1}^{\tau} \frac{P(s_{i-1}, s_i)\gamma(s_{i-1})}{P(s_{i-1}, s_i)(c(s_{i-1} \to s_i) + \gamma(s_i))} \\ &= c(s_{\tau-1} \to s_\tau) \frac{\gamma(s_0)}{c(s_{\tau-1} \to s_\tau) + \gamma(s_\tau)} \\ &= \gamma \end{aligned}$$

The last equality comes from the fact that s_τ is a terminal state and so $\gamma(s_\tau) = 0$. It follows that ZL is a constant random variable and so has zero variance. □

Consequently, some common properties have a zero variance importance sampling estimator. We list a few of them next.

Theorem 3. *Consider the stopping criterion "reach Δ", where Δ is a set of states strictly included in S, such that the probability of reaching Δ in a finite time is 1. Let $A \subset \Delta$, an initial state $s_0 \notin \Delta$ and $\varphi = \mathbf{F}\, s \in A$.*

There exists an importance sampling estimator of γ with zero variance in this setting.

Proof. Let $c : T \to \{0, 1\}$ such that $c(s_i \to s_j) = 1$ if $s_j \in A$ and 0 otherwise.

Trace ω is checked at each transition. So, by construction,

$$z(\omega) = \sum_{i=1}^{\tau} c(s_{i-1} \to s_i) = c(s_{\tau-1} \to s_\tau)$$

Then, the theorem is a consequence of Theorem 2. □

Theorem 4. *Consider the stopping criterion "reach Δ", where Δ is a set of states strictly included in S, such that the probability of reaching Δ in a finite time is* 1*. Let $A \subset \Delta$, an initial state $s_0 \notin \Delta$ and $\varphi = \mathbf{G}\, s \in S \setminus A$.*

There exists an importance sampling estimator of γ with zero-variance in this setting.

Proof. Let $d : T \to \{0, 1\}$ such that $d(s_i \to s_j) = 1$ if $s_j \in A$ and 0 otherwise.

By construction,

$$z(\omega) = 1 - \sum_{i=1}^{\tau} d(s_{i-1} \to s_i) = 1 - d(s_{\tau-1} \to s_\tau)$$

The theorem follows by applying Theorem 2 with the functional equality $c = 1 - d$. □

Theorem 5. *Consider the following stopping criteria "reach A" a set of states strictly included in S such that that the probability of reaching A in a finite time is* 1*.*

Let B a non empty set such that $A \cap B = \emptyset$, an initial state $s_0 \notin A \cup B$ and $\varphi = \neg(s \in B)\mathbf{U}\, s \in A$.

There exists an importance sampling estimator of γ with zero-variance in this setting.

Proof. This theorem is a corollary of Theorem 3. Indeed, the property is not simple reachability because it is not enough to just reach A: B must also be avoided. However, as any trace reaching B before A is unsuccessful, the problem is similar to Theorem 3 by defining $\Delta = A \sqcup B$. □

Whenever the property is time-bounded, the theorem does not hold in general. Indeed, the same transition could provoke with probability 1 a violation in a case and a satisfaction in another case. The following example demonstrates this.

Consider the simple transition system depicted in Fig. 13, together with the property $\mathbf{X}(\mathbf{G}^4 \neg(s = s_0 \vee s = s_5))$. No paths containing transitions b, f or g satisfy the property, while paths containing transition e always satisfy the property and transitions a, c, and d exist in paths that both satisfy and do not satisfy the property. If we change the time bound of $\mathbf{G}$ to 3, the nature of transitions a, b, c, d, e and g is unchanged. However, the nature of transition f depends on the time at which the transition is taken. For example, path $acdf$ satisfies the property but afg does not.

Nevertheless, if transitions that are forbidden (i.e., cause the property not to be satisfied) when the property is unbounded remain forbidden when the property is bounded, Theorems 3, 4 and 5 remain valid.

The situation for continuous time properties is poor, as expressed in the following theorem.

Theorem 6. *The zero variance distribution for a time-bounded property of a continuous time Markov chain (CTMC) cannot be represented by a re-parametrisation of the transition rates of the CTMC.*

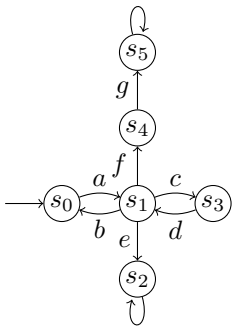

Figure 13: Simple transition system.

Proof. Let $X = \{X(t), t \geq 0\}$ be a CTMC evolving in state space $\mathcal{X}$. We denote by π_0 the initial distribution and by $\lambda_{x,y}$ the jump rate from x to y, $y \neq x$. The departure rate from x is $\lambda_x = \sum_{y \neq x} \lambda_{x,y}$. A finite path of this CTMC is denoted $\omega = (\omega_0, t_0, \omega_1, t_1, \cdots, \omega_n, t_n)$, with ω_k the k-th visited state of the chain and t_k the sojourn time in ω_k.

Let ϕ be a property of interest that is time-bounded by T and assume that there exists a re-parametrisation of the values $\lambda_{x,y}$ defining a zero variance distribution. Recall that the zero variance distribution is such that the property occurs with probability 1 within T time units. By the definition of a CTMC, whatever the parametrisation, the density of a finite path is given by the product of individual transition probabilities from ω_k to ω_{k+1} times the product of the densities for leaving ω_k after sojourn time t_k, which simplifies to

$$\mathrm{d}P(\omega) = \left(\prod_{k=0}^{n-1} \lambda_{\omega_k, \omega_{k+1}} \right) \exp\left(\sum_{k=0}^{n} \lambda_{\omega_k} t_k \right).$$

For the zero variance distribution to exist, the parametrisation should thus assign a zero density to the set of paths of length $\sum_{k=0}^{n} t_k > T$, which is not possible given that for all λ_{ω_k} and t_k, $\exp\left(\sum_{k=0}^{n} \lambda_{\omega_k} t_k\right) > 0$. □

8. Confidence

Bounding the error of estimates produced by importance sampling in the context of SMC remains an open challenge. This is because the importance sampling distribution is only specified implicitly by the transition kernel and the property. It therefore has unknown form and variance. Moreover, importance sampling breaks the link between the probability of seeing an event and its significance, hence what is observed from a finite number of simulations may not adequately approximate the true distribution. In the case of standard Monte

Carlo estimation, the form of the distribution is known to be Bernoulli, with only the parameter unknown. By assuming a worst case variance it is possible to predict convergence and calculate a lower bound of the number of simulations necessary to guarantee a level of statistical confidence.

It is common to bound the error of SMC estimates with the standard confidence interval [7, Section 1.1] or a Hoeffding bound [10]. The confidence interval relies on the fact that the distribution of estimates of a Bernoulli random variable converges rapidly to a normal. With the assumption that the number of samples will always be sufficient for convergence, it is possible to estimate the probability that an N-sample estimate $\hat{\gamma}$ is within ϵ of the true value γ using

$$\mathrm{P}(\parallel \gamma - \hat{\gamma} \parallel \leq \epsilon) \approx 2\Phi(\epsilon\sqrt{(N-1)/\hat{\gamma}(1-\hat{\gamma})}) - 1. \tag{28}$$

Function Φ is the cumulative density of a standard normal. Equation (28) is applicable in general, but the accuracy of the approximation relies on the assumption that $N\hat{\gamma}(1-\hat{\gamma})/(N-1)$ is a good estimate of the true variance of the random variable. This is valid in the case of a Bernoulli random variable, but may be grossly in error in the case of importance sampling.

With traces containing a finite number of steps, the range of values that the likelihood ratio distribution may take is finitely bounded, implying that its variance is also finitely bounded and that the distribution of estimates will converge to normality in the limit of samples (by the central limit theorem). Some authors have thus inferred that a confidence interval may be applied, but there are fundamental problems. With only the guarantee of finite variance, it cannot be assumed that the number of samples will be sufficient for adequate convergence to normality. Moreover, as demonstrated by the first point in Fig. 11, a poor importance sampling distribution may underestimate γ by tens of orders of magnitude, giving an estimated variance $N\hat{\gamma}(1-\hat{\gamma})/(N-1) \approx 0$ and leading to grossly overestimated confidence (28).

The Hoeffding bound does not rely on convergence to normality and requires only the minimum and maximum possible values (denoted a, b, respectively) of the estimator distribution to relate the number of samples to the probability that the estimate will lie within ϵ of the true value:

$$\mathrm{P}(\parallel \gamma - \hat{\gamma}_N \parallel \geq \epsilon) \leq 2e^{-2N\epsilon^2/(b-a)^2}. \tag{29}$$

In the case of (3), $a = 0$ and $b = 1$ and (29) reduces to the usual Okamoto bound [9].

The Hoeffding bound may be correctly applied to importance sampling estimates using the minimum and maximum possible values of the likelihood ratio. In practice, however, these values are not known and must be conservatively estimated to ensure correctness (e.g., by assuming worst case likelihood ratio on every transition). In all but exceptional cases, such estimates do not provide bounds that require significantly fewer simulations than standard Monte Carlo.

The problems outlined above assume no prior knowledge about the importance sampling distribution. In the case of the distributions that result from

the convergence of Algorithm 1, it may be assumed that they avoid the pathological case illustrated in Fig. 1e and are like the distribution of Fig. 1f. While minimum cross-entropy does not in general imply minimum variance (except in the unusual case that $f^* = f(\cdot, \lambda^*)$ [44]), the fraction of successful paths under $f(\cdot, \lambda^*)$ is an indication of how close it is to f^* and therefore an indication of how reliable the estimate is.

As a statistical process, the results of importance sampling may be both positive and negative with respect to the true value. Although the importance sampling estimator is unbiased, with too few samples there may be a predominance of underestimates before eventually a much larger overestimate is seen. The converse cannot happen because a low value of likelihood ratio function $L(\omega) = \mathrm{d}f(\omega)/\mathrm{d}f'(\omega)$ implies that path ω has relatively high probability under the importance sampling distribution f'–low estimates cannot be rare.

This phenomenon is the same as using too few samples with standard Monte Carlo: most estimates will be zero, but eventually there will be a non-zero result, such that the average of all results converges to the correct value. The difference is that consistent non-zero results with importance sampling give an impression of correctness that cannot be contradicted without further simulations, whereas a zero result with standard Monte Carlo immediately indicates a problem.

Overall, we may conclude that importance sampling will typically produce an underestimate. If the fraction of successful traces under the importance sampling distribution is greater than 0.5, we may conclude that the underestimate is relatively small. To guard against pathological cases (e.g., Fig. 1e), such as may result from incomplete convergence of Algorithm 1, we should not expect more variance reduction than the number of samples we use (typically $10^3 - 10^5$ in our examples). Hence, if the sample variance is more than the number of samples times less than the estimate (as in the first point of Fig. 11) we should be suspicious that Algorithm 1 did not use sufficient samples per iteration or did not use sufficient iterations.

9. Conclusions

We have devised a simple and tractable cross-entropy minimisation algorithm to find optimal parametrised importance sampling distributions for statistical model checking. Our parametrisation is automatically generated from the syntax of models described by guarded commands and leads to a unique optimum. Linking the parametrisation to the description of the model in this way gives our approach an advantage when the rare property is related to semantic features expressed in the syntax, such as explicit commands for failure or repair in reliability models. This explains why we are able to demonstrate significant improvements in efficiency with very few parameters.

While the minimum cross-entropy distribution is not in general the same as the minimum variance distribution [44], it nevertheless guarantees certain qualitative properties and allows us to heuristically avoid erroneous results. Formally bounding the error of results obtained by importance sampling is a long-standing

open problem, however we feel further progress may be made by identifying specific classes of models for which the relationship between minimum cross entropy and minimum variance can be defined.

References

[1] P. Godefroid, M. Y. Levin, D. A. Molnar, Automated whitebox fuzz testing, in: NDSS, Vol. 8, 2008, pp. 151–166.

[2] D. Delmas, J. Souyris, Static Analysis: 14th International Symposium, SAS 2007, Kongens Lyngby, Denmark, August 22-24, 2007. Proceedings, Springer Berlin Heidelberg, 2007, Ch. Astrée: From Research to Industry, pp. 437–451.

[3] J. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, L.-J. Hwang, Symbolic model checking: 10^{20} states and beyond, Information and computation 98 (2) (1992) 142–170.

[4] J. G. Kemeny, J. L. Snell, Finite Markov Chains, Springer, 1976.

[5] A. Basu, S. Bensalem, M. Bozga, B. Caillaud, B. Delahaye, A. Legay, Statistical abstraction and model-checking of large heterogeneous systems, in: J. Hatcliff, E. Zucca (Eds.), Formal Techniques for Distributed Systems, Vol. 6117 of LNCS, Springer, 2010, pp. 32–46.

[6] E. M. Clarke, E. A. Emerson, J. Sifakis, Model checking: algorithmic verification and debugging, Commun. ACM 52 (11) (2009) 74–84.

[7] G. Rubino, B. Tuffin (Eds.), Rare Event Simulation using Monte Carlo Methods, Wiley, 2009.

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

[9] M. Okamoto, Some inequalities relating to the partial sum of binomial probabilities, Annals of the Institute of Statistical Mathematics 10 (1) (1958) 29–35.

[10] W. Hoeffding, Probability inequalities for sums of bounded random variables, Journal of the American Statistical Association 58 (301) (1963) 13–30.

[11] S. K. Jha, E. M. Clarke, C. J. Langmead, A. Legay, A. Platzer, P. Zuliani, A Bayesian approach to model checking biological systems, in: Computational Methods in Systems Biology, Springer, 2009, pp. 218–234.

[12] A. Wald, Sequential tests of statistical hypotheses, The Annals of Mathematical Statistics 16 (2) (1945) 117–186.

[13] H. Younes, R. Simmons, Probabilistic verification of discrete event systems using acceptance sampling, in: Computer Aided Verfication, Vol. 2404, Springer, 2002, pp. 23–39.

[14] T. Hérault, R. Lassaigne, F. Magniette, S. Peyronnet, Approximate probabilistic model checking, in: B. Steffen, G. Levi (Eds.), Verification, Model Checking, and Abstract Interpretation, Vol. 2937 of LNCS, Springer, 2004, pp. 307–329.

[15] H. Younes, YMER: A statistical model checker, in: K. Etessami, S. Rajamani (Eds.), Computer Aided Verification, Vol. 3576 of LNCS, Springer, 2005, pp. 171–179.

[16] K. Sen, M. Viswanathan, G. A. Agha, VESTA: A statistical model-checker and analyzer for probabilistic systems, in: Quantitative Evaluation of Systems, IEEE, 2005, pp. 251–252.

[17] C. Jegourel, A. Legay, S. Sedwards, A platform for high performance statistical model checking – PLASMA, in: C. Flanagan, B. König (Eds.), Tools and Algorithms for the Construction and Analysis of Systems, LNCS, Springer, Tallinn, Estonia, 2012, pp. 498–503.

[18] P. Ballarini, H. Djafri, D. M., H. S., N. Pekergin, Cosmos: A statistical model checker for the hybrid automata stochastic logic, in: Eighth International Conference on Quantitative Evaluation of Systems (QEST), 2011, pp. 143–144.

[19] M. Kwiatkowska, G. Norman, D. Parker, PRISM: Probabilistic symbolic model checker, in: T. Field, P. Harrison, J. Bradley, U. Harder (Eds.), Computer Performance Evaluation: Modelling Techniques and Tools, Vol. 2324 of LNCS, Springer, 2002, pp. 113–140.

[20] J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson, W. Yi, UPPAAL – a tool suite for automatic verification of real-time systems, in: R. Alur, T. Henzinger, E. Sontag (Eds.), Hybrid Systems III, Vol. 1066 of LNCS, Springer, 1996, pp. 232–243.

[21] B. Boyer, K. Corre, A. Legay, S. Sedwards, PLASMA-lab: A flexible, distributable statistical model checking library, in: K. Joshi, M. Siegle, M. Stoelinga, P. R. D'Argenio (Eds.), Quantitative Evaluation of Systems, Vol. 8054 of LNCS, Springer, 2013, pp. 160–164.

[22] N. Metropolis, S. Ulam, The Monte Carlo method, Journal of the American Statistical Association 44 (247) (1949) 335–341.

[23] H. Kahn, Stochastic (Monte Carlo) attenuation analysis, Tech. Rep. P-88, Rand Corporation (July 1949).

[24] C. Jegourel, A. Legay, S. Sedwards, Cross-entropy optimisation of importance sampling parameters for statistical model checking, in: P. Madhusudan, S. A. Seshia (Eds.), Computer Aided Verification, Vol. 7358 of LNCS, Springer, 2012, pp. 327–342.

[25] E. W. Dijkstra, Guarded commands, nondeterminacy and formal derivation of programs, Commun. ACM 18 (1975) 453–457.

[26] S. Kullback, Information Theory and Statistics, Dover, 1968.

[27] R. Y. Rubinstein, Optimization of computer simulation models with rare events, European Journal of Operations Research 99 (1997) 89–112.

[28] R. Rubinstein, The cross-entropy method for combinatorial and continuous optimization, in: Methodology and Computing in Applied Probability, Vol. 1, Kluwer Academic, 1999, pp. 127–190.

[29] P. Shahabuddin, Importance sampling for the simulation of highly reliable markovian systems, Management Science 40 (3) (1994) 333–352.

[30] P. Heidelberger, Fast simulation of rare events in queueing and reliability models, ACM Trans. Model. Comput. Simul. 5 (1995) 43–85.

[31] A. Ridder, Importance sampling simulations of markovian reliability systems using cross-entropy, Annals of Operations Research 134 (2005) 119–136.

[32] S. Juneja, P. Shahabuddin, Fast simulation of Markov chains with small transition probabilities, Management Science 47 (4) (2001) 547–562.

[33] S. Juneja, P. Shahabuddin, Splitting-based importance-sampling algorithm for fast simulation of Markov reliability models with general repair-policies, IEEE Transactions on Reliability 50 (3) (2001) 235–245.

[34] D. Reijsbergen, P.-T. de Boer, W. R. W. Scheinhardt, B. R. Haverkort, Rare event simulation for highly dependable systems with fast repairs, in: Quantitative Evaluation of Systems, 2010, pp. 251–260.

[35] E. Clarke, P. Zuliani, Statistical model checking for cyber-physical systems, in: T. Bultan, P.-A. Hsiung (Eds.), Automated Technology for Verification and Analysis, Vol. 6996 of LNCS, Springer, 2011, pp. 1–12.

[36] B. Barbot, S. Haddad, C. Picaronny, Coupling and importance sampling for statistical model checking, in: C. Flanagan, B. König (Eds.), Tools and Algorithms for the Construction and Analysis of Systems, LNCS, Springer, Tallinn, Estonia, 2012, pp. 331–346.

[37] C. Jegourel, A. Legay, S. Sedwards, Importance splitting for statistical model checking rare properties, in: N. Sharygina, H. Veith (Eds.), Computer Aided Verification, Vol. 8044 of LNCS, Springer, 2013, pp. 576–591.

[38] D. T. Gillespie, Exact stochastic simulation of coupled chemical reactions, Journal of Physical Chemistry 81 (1977) 2340–2361.

[39] C. P. Robert, G. Casella, Monte Carlo statistical methods, 2nd Edition, Springer, 2004.

[40] P.-T. de Boer, V. F. Nicola, R. Y. Rubinstein, Adaptive importance sampling simulation of queueing networks, in: Winter Simulation Conference, Vol. 1, 2000, pp. 646–655.

[41] J. Shore, R. Johnson, Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy, IEEE Transactions on Information Theory 26 (1) (1980) 26–37.

[42] A. Ridder, Asymptotic optimality of the cross-entropy method for markov chain problems, Procedia Computer Science 1 (1) (2010) 1571–1578.

[43] H. Hermanns, J. Meyer-Kayser, M. Siegle, Multi terminal binary decision diagrams to represent and analyse continuous time Markov chains, in: B. Plateau, W. Stewart, M. Silva (Eds.), Proc. 3rd International Workshop on Numerical Solution of Markov Chains (NSMC'99), Prensas Universitarias de Zaragoza, Zaragoza, 1999, pp. 188–207.

[44] T. Homem-de Mello, A study on the cross-entropy method for rare-event probability estimation, INFORMS Journal on Computing 19 (3) (2007) 381–394.