Cross-entropy optimisation of importance sampling parameters for statistical model checking

Cyrille Jegourel, Axel Legay, and Sean Sedwards

INRIA Rennes - Bretagne Atlantique,
{cyrille.jegourel,axel.legay,sean.sedwards}@inria.fr

Abstract. Statistical model checking avoids the exponential growth of states associated with probabilistic model checking by estimating probabilities from multiple executions of a system and by giving results within confidence bounds. Rare properties are often important but pose a particular challenge for simulation-based approaches, hence a key objective for statistical model checking (SMC) is to reduce the number and length of simulations necessary to produce a result with a given level of confidence. *Importance sampling* can achieves this, however to maintain the advantages of SMC it is necessary to find good importance sampling distributions without considering the entire state space.
Here we present a simple algorithm that uses the notion of cross-entropy to find an optimal importance sampling distribution. In contrast to previous work, our algorithm uses a naturally defined low dimensional vector of parameters to specify this distribution and thus avoids the intractable explicit representation of a transition matrix. We show that our parametrisation leads to a unique optimum and can produce many orders of magnitude improvement in simulation efficiency. We demonstrate the efficacy of our methodology by applying it to models from reliability engineering and biochemistry.

1 Introduction

The need to provide accurate predictions about the behaviour of complex systems is increasingly urgent. With computational power becoming ever-more affordable and compact, computational systems are inevitably becoming increasingly concurrent, distributed and adaptive, creating a correspondingly increased burden to check that they function correctly. At the same time, users expect high performance and reliability, prompting the need for equally high performance analysis tools and techniques.

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. Testing techniques have been effective discovering bugs in many industrial applications and have been incorporated into sophisticated tools [9]. 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 may remain undetected and quantifying the likelihood of failure using a series of test cases is difficult.

Model checking is a formal technique that verifies whether a system satisfies a property specified in temporal logic under all possible scenarios. In recognition of non-deterministic systems and the fact that a Boolean answer is not always useful, *probabilistic* model checking quantifies the probability that a system satisfies a property. In particular, 'numerical' (alternatively 'exact') probabilistic model checking offers precise and accurate analysis by exhaustively exploring the state space of non-deterministic systems and has been successfully applied to a wide variety of protocols, algorithms and systems. The result of this technique is the exact (within limits of numerical precision) probability that a system will satisfy a property of interest, however the exponential growth of the state space limits its applicability. The typical 10^8 state limit of exhaustive approaches usually represents an insignificant fraction of the state space of real systems that may have tens of orders of magnitude more states than the number of protons in the universe ($\sim 10^{80}$).

Under certain circumstances it is possible to guarantee the performance of a system by specifying it in such a way that (particular) faults are impossible. Compositional reasoning and various symmetry reduction techniques can also be used to combat state-space explosion, but in general the size, unpredictability and heterogeneity of real systems [2] make these techniques infeasible. Static analysis has also been highly successful in analysing and debugging software and other systems, although it cannot match the precision of quantitative analysis of dynamic properties achieved using probabilistic and stochastic temporal logic.

While the state space explosion problem is unlikely to ever be adequately solved for all systems, simulation-based approaches are becoming increasingly tractable due to the availability of high performance hardware and algorithms. In particular, statistical model checking (SMC) combines the simplicity of testing with the formality and precision of numerical model checking; the core idea being to create multiple independent execution traces of the system and individually verify whether they satisfy some given property. By modelling the executions as a Bernoulli random variable and using advanced statistical techniques, such as Bayesian inference [14] and hypothesis testing [27], the results are combined in an efficient manner to decide whether the system satisfies the property with some level of confidence, or to estimate the probability that it does. Knowing a result with less than 100% confidence is often sufficient in real applications, since the confidence bounds may be made arbitrarily tight. Moreover, SMC may offer the only feasible means of quantifying the performance of many complex systems. Evidence of this is that SMC has been used to find bugs in large, heterogeneous aircraft systems [2]. Notable SMC platforms include APMC [11], YMER [28] and VESTA [23]. Moreover, well-established numerical model checkers, such as PRISM [17] and UPPAAL [3], are now also including SMC engines.

A key challenge facing SMC is to reduce the length (steps and cpu time) and number of simulation traces necessary to achieve a result with given confidence. The current proliferation of parallel computer architectures (multiple cpu cores,

grids, clusters, clouds and general purpose computing on graphics processors, etc.) favours SMC by making the production of multiple independent simulation runs relatively easy. Despite this, certain models still require a large number of simulation steps to verify a property and it is thus necessary to make simulation as efficient as possible. Rare properties pose a particular problem for simulation-based approaches, since they are not only difficult to observe (by definition) but their probability is difficult to bound [10].

The term 'rare event' is ubiquitous in the literature, but here we specifically consider rare *properties* defined in temporal logic. This distinguishes rare states from rare paths that may or may not contain rare states. In what follows we consider discrete space Markov models and present a simple algorithm to find an optimal set of importance sampling parameters, using the concept of minimum cross-entropy [16, 25]. Our parametrisation arises naturally from the syntactic description of the model and thus constitutes a low dimensional vector in comparison to the state space of the model. We show that this parametrisation has a unique optimum and demonstrate its effectiveness on reliability and (bio)chemical models. We describe the advantages and potential pitfalls of our approach and highlight areas for future research.

2 Importance sampling

Our goal is to estimate the probability of a property by simulation and bound the error of our estimation. When the property is not rare there are standard bounding formulae (e.g., the Chernoff and Hoeffding bounds [4, 12]) that relate absolute error, confidence and the required number of simulations to achieve them, *independent* of the probability of the property. As the property becomes rarer, however, absolute error ceases to be useful and it is necessary to consider relative error, defined as the standard deviation of the estimate divided by its expectation. With Monte Carlo simulation relative error is unbounded with increasing rarity [21], but it is possible to bound the error by means of importance sampling [24, 10].

Importance sampling is a technique that can improve the efficiency of simulating rare events and has been receiving considerable interest of late in the field of SMC (e.g., [5, 1]). It works by simulating under an (importance sampling) distribution that makes a property more likely to be seen and then uses the results to calculate the probability under the original distribution by compensating for the differences. The concept arose from work on the 'Monte Carlo method' [18] in the Manhattan project during 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., [15]). 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 paths that satisfy the property in the importance sampling distribution must be as close as possible to the distribution of the same paths in the original distribution (up to a normalising factor). The literature in this

field sometimes uses the term 'zero variance' to describe an optimal importance sampling distribution, referring to the fact that with an optimum importance sampling distribution all simulated paths satisfy the property and the estimator has zero variance. It is important to note, however, that a sub-optimal distribution may meet requirement (i) without necessarily meeting requirement (ii). Failure to consider (ii) can result in gross errors and overestimates of confidence (e.g. a distribution that simulates just one path that satisfies the given property). The algorithm we present in Section 3 addresses both (i) and (ii).

Importance sampling schemes fall into two broad categories: *state dependent tilting* and *state independent tilting* [6]. State dependent tilting refers to importance sampling distributions that individually bias ('tilt') every transition probability in the system. State independent tilting refers to importance sampling distributions that change classes of transition probabilities, independent of state. The former offers greatest precision but is infeasible for large models. The latter is more tractable but may not produce good importance sampling distributions. Our approach is a kind of *parametrised tilting* that potentially affects all transitions differently, but does so according to a set of parameters.

2.1 Estimators

Let Ω be a probability space of paths, with f a probability density function over Ω and $z(\omega) \in \{0, 1\}$ a function indicating whether a path ω satisfies some property ϕ. In the present context, z is defined by a formula of an arbitrary temporal logic over execution traces. The probability γ that ϕ occurs in a path is then given by

$$\gamma = \int_{\Omega} z(\omega) f(\omega) \, \mathrm{d}\omega \tag{1}$$

and the standard Monte Carlo estimator of γ is given by

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

N_{MC} denotes the number of simulations used by the Monte Carlo estimator and ω_i is sampled according to f. Note that $z(\omega_i)$ is effectively the realisation of a Bernoulli random variable with parameter γ. Hence $\mathrm{Var}(\tilde{\gamma}) = \gamma(1 - \gamma)$ and for $\gamma \to 0$, $\mathrm{Var}(\tilde{\gamma}) \approx \gamma$. Let f' be another probability density function over Ω, absolutely continuous with zf, then Equation (1) can be written

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

$L = f/f'$ is the *likelihood ratio* function, so

$$\gamma = \int_{\Omega} L(\omega) z(\omega) f'(\omega) \, \mathrm{d}\omega \tag{2}$$

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

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

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 $N_{\text{IS}} \ll N_{\text{MC}}$. In general, the importance sampling distribution f' is chosen to produce the rare property more frequently, but this is not the only criterion. The optimal importance sampling distribution, denoted f^* and defined as f conditioned on the rare event, produces only traces satisfying the rare property:

$$f^* = \frac{zf}{\gamma} \tag{3}$$

This leads to the term 'zero variance estimator' with respect to Lz, noting that, in general, $\text{Var}(f^*) \geq 0$.

In the context of SMC f 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 [7] extended with stochastic rates. Our parametrisation is a vector of strictly positive values $\lambda \in (\mathbb{R}^+)^n$ that 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 density functions by their parameters. In particular, μ is the original vector of the model and $f(\cdot, \mu)$ is therefore the original density. We can thus rewrite Equation (2) as

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

where $L(\omega) = f(\omega, \mu)/f(\omega, \lambda)$. We can also rewrite Equation (3)

$$f^* = \frac{zf(\cdot, \mu)}{\gamma}$$

and write for the optimal parametrised density $f(\cdot, \lambda^*)$. We define the optimum parametrised density function as the density that minimises the *cross-entropy* [16] between $f(\cdot, \lambda)$ and f^* for a given parametrisation and note that, in general, $f^* \neq f(\cdot, \lambda^*)$.

2.2 The cross-entropy method

Cross-entropy [16] (alternatively *relative entropy* or Kullback-Leibler divergence) has been shown to be a uniquely correct directed measure of distance between distributions [25]. With regard to the present context, it has also been shown to be useful in finding optimum distributions for importance sampling [22, 6, 19].

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

$$\begin{aligned}\mathrm{CE}(f, f') = \int_\Omega f(\omega) \log \frac{f(\omega)}{f'(\omega)} \,\mathrm{d}\omega &= \int_\Omega f(\omega) \log f(\omega) - f(\omega) \log f'(\omega) \,\mathrm{d}\omega \\ &= \mathrm{H}(f) - \int_\Omega f(\omega) \log f'(\omega) \,\mathrm{d}\omega \end{aligned} \tag{4}$$

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

$$\lambda^* = \arg\max_\lambda \int_\Omega z(\omega) f(\omega, \mu) \log f(\omega, \lambda) \,\mathrm{d}\omega \tag{5}$$

Estimating λ^* directly using Equation (5) is hard, so we re-write it using importance sampling density $f(\cdot, \lambda')$ and likelihood ratio function $L(\omega) = f(\omega, \mu)/f(\omega, \lambda')$:

$$\lambda^* = \arg\max_\lambda \int_\Omega z(\omega) L(\omega) f(\omega, \lambda') \log f(\omega, \lambda) \,\mathrm{d}\omega \tag{6}$$

Using Equation (6) 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:

$$\tilde{\lambda^*} = \lambda^{(j+1)} = \arg\max_\lambda \sum_{i=1}^{N_j} z(\omega_i^{(j)}) L^{(j)}(\omega_i^{(j)}) \log f(\omega_i^{(j)}, \lambda) \tag{7}$$

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

3 A parametrised cross-entropy algorithm

We consider a system of n 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, the probability that command $k \in \{1 \ldots n\}$ is chosen is given by

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

where η is parametrised by x 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 $U_k(\omega)$ be the number of transitions of type k occurring in ω. We therefore have

$$f(\omega, \lambda) = \prod_k^n \left((\lambda_k)^{U_k(\omega)} \prod_{s=1}^{U_k(\omega)} \frac{\eta_k(x_s)}{\langle \eta(x_s), \lambda \rangle} \right)$$

The likelihood ratios are thus of the form

$$L^{(j)}(\omega) = \prod_k^n \left(\left(\frac{\mu_k}{\lambda_k^{(j)}} \right)^{U_k(\omega)} \prod_{s=1}^{U_k(\omega)} \frac{\langle \eta(x_s), \lambda^{(j)} \rangle}{\langle \eta(x_s), \mu \rangle} \right)$$

We substitute these expressions in the cross-entropy estimator Equation (7) and for compactness substitute $z_i = z(\omega_i)$, $u_i(k) = U_k(\omega_i)$ and $l_i = L^{(j)}(\omega_i)$ to get

$$\begin{aligned} &\arg\max_\lambda \sum_{i=1}^N l_i z_i \log \prod_k^n \left(\lambda_k^{u_i(k)} \prod_{s=1}^{u_i(k)} \frac{\eta_k^i(x_s)}{\langle \eta^i(x_s), \lambda \rangle} \right) \qquad (8) \\ &= \arg\max_\lambda \sum_{i=1}^N \sum_k^n l_i z_i u_i(k) \left(\log(\lambda_k) + \sum_{s=1}^{u_i(k)} \log(\eta_k^i(x_s)) - \sum_{s=1}^{u_i(k)} \log(\langle \eta^i(x_s), \lambda \rangle) \right) \end{aligned}$$

We partially differentiate with respect to λ_k and get the non-linear system

$$\frac{\partial F}{\partial \lambda_k}(\lambda) = 0 \Leftrightarrow \sum_{i=1}^N l_i z_i \left(\frac{u_i(k)}{\lambda_k} - \sum_{s=1}^{|\omega_i|} \frac{\eta_k^i(x_s)}{\langle \eta^i(x_s), \lambda \rangle} \right) = 0 \qquad (9)$$

where $|\omega_i|$ is the length of the path ω_i.

Theorem 1. *A solution of Equation (9) is almost surely a maximum, up to a normalising scalar.*

Proof. Using a standard result, it is sufficient to show that the Hessian matrix in λ is negative semi-definite. Consider f_i:

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

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

$$H_i = G - D$$

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

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

and D is a diagonal matrix such that

$$d_{kk} = \frac{u_k}{\lambda_k^2}.$$

Note that asymptotically $d_{kk} = \frac{1}{\lambda_k} \sum_{s=1}^{N} v_k^{(s)}$. We write $\mathbb{1}_N = (1, \ldots, 1)$ for the vector of N elements 1, hence

$$d_{kk} = \frac{1}{\lambda_k} \langle v_k, \mathbb{1}_N \rangle.$$

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

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

Let $x \in \mathbb{R}^n$. To prove the theorem we need to show that $-x^t H x \geq 0$.

$$\begin{aligned} -x^t H x &= x^t D x - x^t G x \qquad (10) \\ &= \sum_{k,k'} \frac{\lambda_{k'}}{\lambda_k} \langle v_k, v_{k'} \rangle x_k^2 - \sum_{k,k'} \langle v_k, v_{k'} \rangle x_k x_{k'} \\ &= \sum_{k<k'} \left(\left[\frac{\lambda_{k'}}{\lambda_k} x_k^2 + \frac{\lambda_k}{\lambda_{k'}} x_{k'}^2 - 2 x_k x_{k'} \right] \langle v_k, v_{k'} \rangle \right) \\ &= \sum_{k<k'} \left(\sqrt{\frac{\lambda_{k'}}{\lambda_k}} x_k - \sqrt{\frac{\lambda_k}{\lambda_{k'}}} x_{k'} \right)^2 \langle v_k, v_{k'} \rangle \\ &\geq 0 \end{aligned}$$

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. □

The Hessian is negative *semi*-definite because if λ is a solution then $x\lambda, x \in \mathbb{R}^+$, is also a solution. The fact that there is a unique optimum, however, 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 [19], 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, Equation (12) may be transformed to Equation (9) of [19] by constraining $\langle K, \lambda \rangle = 1$. Equation (9) of [19] has been shown in [20] to converge to f^*, 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.

3.1 The algorithm

Equation (9) leads to the following expression for λ_k:

$$\lambda_k = \frac{\sum_{i=1}^{N} l_i z_i u_i(k)}{\sum_{i=1}^{N} l_i z_i \sum_{s=1}^{|\omega_i|} \frac{K_k^s}{\langle K^s, \lambda \rangle}} \tag{11}$$

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 Equation (7) and in [19, 6], we construct an iterative process based on a biased estimator but having a fixed point that is the optimum:

$$\lambda_k^{(j+1)} = \frac{\sum_{i=1}^{N_j} l_i z_i u_i(k)}{\sum_{i=1}^{N_j} l_i z_i \sum_{s=1}^{|\omega_i|} \frac{K_k^s}{\langle K^s, \lambda^{(j)} \rangle}} \tag{12}$$

Equation (12) can be seen as an implementation of Equation (11) which uses the previous estimate of λ in the scalar product, however it works by reducing the distance between successive distributions, rather than by explicitly reducing the distance from the optimum. To show that the algorithm works, we first recall that Theorem 1 proves that there is a unique optimum (λ^*) of Equation (9) which is therefore the unique solution of Equation (11). By inspection and comparison with Equation (11), we see that any fixed point of Equation (12) is also a solution of Equation (11). Since Equation (11) has a unique solution, Equation (12) has a unique fixed point that is the optimum.

Initial distribution The algorithm requires an initial simulation distribution ($f(\cdot, \lambda^{(0)})$) that produces at least a few traces that satisfy the property ('successful' traces) within N_0 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: the rareness of the property in trace space does not imply that good parameters are rare in parameter space. In particular, 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 can be found relatively easily by heuristic methods such as failure biasing [24]. The choice of $\lambda^{(0)}$ and N_0 is dependent on the model and the rarity of the property, but when the number of parameters is small and the property is very rare, an effective strategy is to simulate with random parameters until a suitable trace is observed. 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 estimate. Increasing the parameters associated to obviously small rates may help (along the lines of failure biasing), however the rareness of a property expressed in temporal logic may not always be related to low transition probabilities. The reliability of finding good initial distributions for arbitrary systems and temporal properties is the subject of ongoing work.

Smoothing It is conceivable that certain guarded commands play no part in traces that satisfy the property, in which case Equation (12) 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 [19] 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 particular importance when the parametrisation is close to the level of individual transition probabilities, since only a tiny fraction of possible transitions are usually seen on an individual simulation run. Typical strategies include adding a small fraction of the initial or previous parameters to every new parameter estimate. We have found that our parametrisation is often insensitive to smoothing strategy since each parameter typically governs many transitions and a large fraction of parameters are touched by each run. The smoothing strategy adopted for the examples shown below was to divide the parameter of unseen commands by two (a compromise between speed of convergence and safety). The effects of this can be seen clearly in Figure 6. 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 progress.

Convergence Given a sufficient number of successful traces from the first iteration, Equation (12) should provide a better set of parameters. In practice we have found that a single successful trace is sufficient to initiate convergence. This is in part due to the existence of a unique optimum and partly to the fact that each parameter generally 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 optimises the *quality* of the distribution 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., Figure 4).

Once the parameters have converged it is then possible to perform a final set of simulations to estimate the probability of the rare property. The usual assumption is that $N_j \ll N_{\text{IS}} \ll N_{\text{MC}}$, 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 as part of the final estimation.

4 Examples

The following examples are included to illustrate the performance of our algorithm and parametrisation. The first is an example of a chemical system, often used to motivate stochastic simulation, while the second is a standard repair

model. In both cases, initial distributions were found by the heuristic of performing single simulations using parameters drawn from a Dirichlet distribution (i.e., drawn uniformly from parameter space) and using the first set of parameters that produce a path satisfying the property. For the chosen examples fewer than 500 attempts were necessary; a value less than N_j and considerably less than $1/\gamma$, the expected number of simulations necessary to see a single successful trace. All simulations were performed using our statistical model checking platform, PLASMA [13].

4.1 Chemical network

Following the success of the human genome project, with vast repositories of biological pathway data available online, 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, E), a dimerisation reaction (13) and two decay reactions (14,15):

$$A + B \xrightarrow{k_1} C \tag{13}$$

$$C \xrightarrow{k_2} D \tag{14}$$

$$D \xrightarrow{k_3} E \tag{15}$$

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 [8]. The semantics of Equation (13) 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 k_1. The decay reactions 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 k_2 (k_3). The reactions (13,14,15) are modelled by three guarded commands having importance sampling parameters λ_1, λ_2 and λ_3, respectively. A typical simulation run is illustrated in Figure 1, where the x-axis is 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.

With an initial vector of molecules $(1000, 1000, 0, 0, 0)$, corresponding to types (A, B, C, D, E), the state space contains $\sim 10^{15}$ states. We know from a static analysis of the reactions 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) $\Diamond C \geq x, x \in$

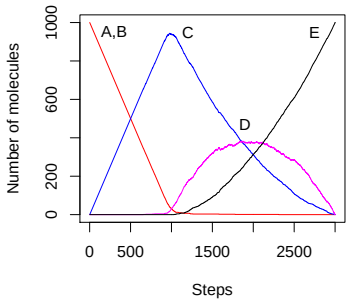

Fig. 1. A typical stochastic simulation trace of reactions (13-15).

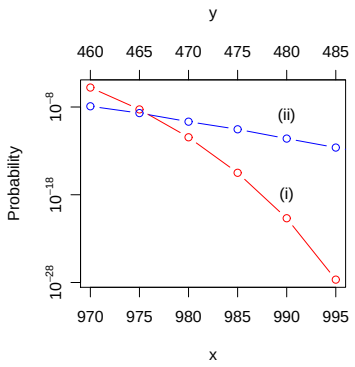

Fig. 2. (i) $\Pr[\Diamond\, C \geq x]$ (ii) $\Pr[\Diamond\, D \geq y]$

$\{970, 975, 980, 985, 990, 995\}$ and (ii) $\Diamond\, D \geq y, y \in \{460, 465, 470, 475, 480, 485\}$. The results are plotted in Figure 2.

Having found an initial set of parameters by the heuristic means described above, the algorithm (Equation (12)) was iterated 20 times using $N_j = 1000$. Despite the large state space, this value of N_j was found to be sufficient to produce reliable results. The convergence of parameters for the property $\Diamond\, D \geq 470$ can be seen in Figure 3. Figure 4 illustrates that the number of paths satisfying a property can actually decrease as the quality of the distribution improves. Figure 5 illustrates the convergence of the estimate and sample variance 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. Noting that the variance of standard Monte Carlo simulation of rare events gives a variance approximately equal to the probability and assuming that the sample variance is close to the true variance, Figure 5 suggests that we have made a variance reduction of approximately 10^7.

4.2 Repair model

To a large extent the need to certify system reliability motivates the use of formal methods and thus reliability models are studied extensively in the literature. The following example is taken from [19] and features a moderately large state space of 40,320 states that 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. The system's evolution begins with no failures and with various probabilistic rates the components fail and are repaired. The 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. Each subsystem type is modelled by

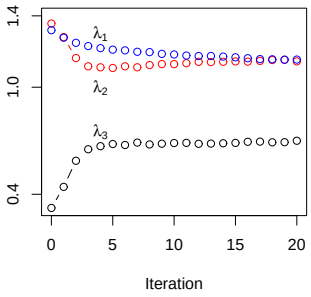

Fig. 3. Convergence of parameters for $\Diamond D \geq 470$ in the chemical model using $N_j = 1000$.

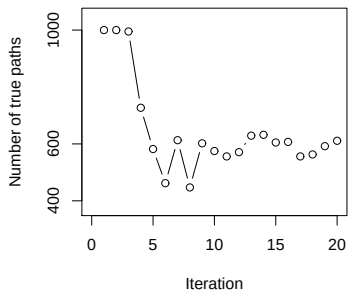

Fig. 4. Convergence of number of paths satisfying $\Diamond D \geq 470$ in the chemical model using $N_j = 1000$.

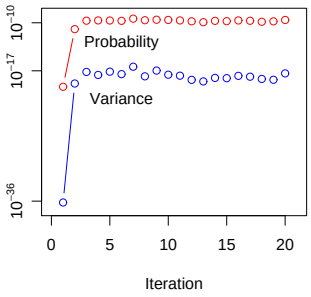

Fig. 5. Convergence of probability and sample variance for $\Diamond D \geq 470$ in the chemical model using $N_j = 1000$.

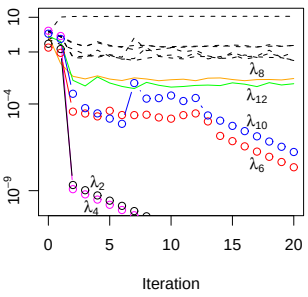

Fig. 6. Convergence of parameters and effect of smoothing (circles) in repair model using $N_j = 10000$.

two guarded commands: one for failure and one for repair. The property under investigation 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 $\Pr[\mathrm{X}(\neg \mathit{init}\ \mathrm{U}\ \mathit{failure})]$.

Figure 6 shows the convergence of parameters (dashed/solid lines) and highlights the effects of the adopted smoothing strategy (circles). Parameters λ_2 and λ_4 (the parameters for repair commands of types 1 and 2, respectively) are attenuated from the outset by the convergence of the other parameters (because of the normalisation). Once their values are small relative to the normalisation constant (12 in this case), their corresponding commands no longer occur and their values experience exponential decay as a result of smoothing (division by two at every subsequent step). Parameters λ_6 and λ_{10} (the parameters for repair commands of types 3 and 5, respectively) converge for 12 steps but then also decay. The parameters for the repair commands of types 4 and 6 (solid lines) are the smallest of the parameters that converge. The fact that the repair transitions are 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 7 plots the number of paths satisfying X($\neg$*init* U *failure*) and suggests that for this model the parametrised distribution is close to the optimum. Figure 8 plots the estimated probability and sample variance during the course of the algorithm and superimposes the true probability calculated by PRISM [26] . The long term average agrees well with the true value (an error of -1.7%, based on an average excluding the first two estimates), justifying our use of the sample variance as an indication of the efficacy of the algorithm: our importance sampling parameters provide a variance reduction of more than 10^5.

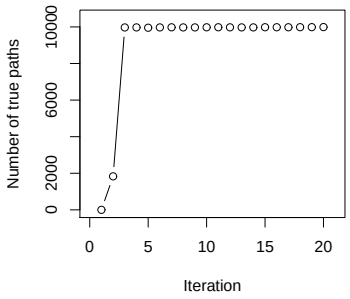

Fig. 7. Convergence of number of paths satisfying X($\neg$*init* U *failure*) in the repair model using $N_j = 10000$.

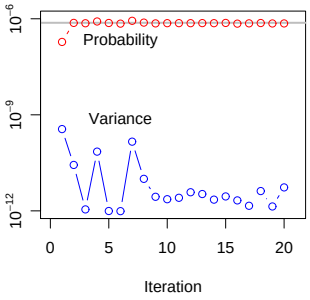

Fig. 8. Convergence of estimated probability and sample variance for repair model using $N_j = 10000$. True probability shown as horizontal line.

5 Conclusions and future work

Statistical model checking addresses the state space explosion associated with exact probabilistic model checking by estimating the parameters of an empirical distribution of executions of a system. By constructing an executable model, rather than an explicit representation of the state space, SMC is able to quantify and verify the performance of systems that are intractable to an exhaustive approach. SMC trades certainty for tractability and often offers the only feasible means to certify real-world systems. Rare properties pose a particular problem to Monte Carlo simulation methods because the properties are difficult to observe and the error in their estimated probabilities is difficult to bound. Importance sampling is a well-established means to reduce the variance of rare events but requires the construction of a suitable importance sampling distribution without resorting to the exploration of the entire state space.

We have devised a natural parametrisation for importance sampling and have provided a simple algorithm, based on cross-entropy minimisation, to optimise the parameters for use in statistical model checking. We have shown that our

parametrisation leads to a unique optimum and have demonstrated that with very few parameters our algorithm can make significant improvements in the efficiency of statistical model checking. We have shown that our approach is applicable to standard reliability models and to the kind of huge state space models found in systems biology. We therefore anticipate that our methodology has the potential to be applied to many complex natural and man-made systems.

An ongoing challenge is to find ways to accurately bound the error of results obtained by importance sampling. Specifically, the sample variance of the results may be a very poor indicator of the true variance (i.e. with respect to the unknown true probability). Recent work has addressed this problem using Markov chain coupling applied to a restricted class of models and logic [1], but a simple universal solution remains elusive. A related challenge is to find precise means to judge the quality of the importance sampling distributions we create. Our algorithm finds an optimum based on an automatic parametrisation of a model described in terms of guarded commands. Linking the importance sampling 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. Potentially confounding this advantage is the fact that the syntactic description is likely optimised for compactness or convenience, rather than consideration of importance sampling. As a result, there may be alternative ways of describing the same model that produce better importance sampling distributions. Applying existing work on the robustness of estimators, we hope to adapt our algorithm to provide hints about improved parametrisation.

References

1. Benoît Barbot, Serge Haddad, and Claudine Picaronny. Coupling and Importance Sampling for Statistical Model Checking. In Cormac Flanagan and Barbara König, editors, *TACAS'12*, LNCS, Tallinn, Estonia, March 2012. Springer. To appear.
2. Ananda Basu, Saddek Bensalem, Marius Bozga, Benoît Caillaud, Benoît Delahaye, and Axel Legay. Statistical abstraction and model-checking of large heterogeneous systems. In John Hatcliff and Elena Zucca, editors, *Formal Techniques for Distributed Systems*, volume 6117 of *LNCS*, pages 32–46. Springer Berlin / Heidelberg, 2010.
3. Johan Bengtsson, Kim Larsen, Fredrik Larsson, Paul Pettersson, and Wang Yi. Uppaal a tool suite for automatic verification of real-time systems. In Rajeev Alur, Thomas Henzinger, and Eduardo Sontag, editors, *Hybrid Systems III*, volume 1066 of *LNCS*, pages 232–243. Springer Berlin / Heidelberg, 1996.
4. Herman Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the sum of Observations. *Ann. Math. Statist.*, 23(4):493–507, 1952.
5. Edmund Clarke and Paolo Zuliani. Statistical model checking for cyber-physical systems. In Tevfik Bultan and Pao-Ann Hsiung, editors, *ATVA*, volume 6996 of *LNCS*, pages 1–12. Springer Berlin / Heidelberg, 2011.
6. P.-T. De Boer, V.F. Nicola, and R.Y. Rubinstein. Adaptive importance sampling simulation of queueing networks. In *Winter Simulation Conference*, volume 1, pages 646–655, 2000.

7. Edsger W. Dijkstra. Guarded commands, nondeterminacy and formal derivation of programs. *Commun. ACM*, 18:453–457, August 1975.
8. D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81:2340–2361, 1977.
9. P. Godefroid, M. Levin, and Molnar. D. Automated whitebox fuzz testing. In *NDSS*, 2008.
10. Philip Heidelberger. Fast simulation of rare events in queueing and reliability models. *ACM Trans. Model. Comput. Simul.*, 5:43–85, January 1995.
11. Thomas Hérault, Richard Lassaigne, Frédéric Magniette, and Sylvain Peyronnet. Approximate probabilistic model checking. In Bernhard Steffen and Giorgio Levi, editors, *Verification, Model Checking, and Abstract Interpretation*, volume 2937 of *LNCS*, pages 307–329. Springer Berlin / Heidelberg, 2004.
12. Wassily Hoeffding. Probability Inequalities for Sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.
13. Cyrille Jegourel, Axel Legay, and Sean Sedwards. A Platform for High Performance Statistical Model Checking – PLASMA. In Cormac Flanagan and Barbara König, editors, *TACAS*, LNCS, Tallinn, Estonia, March 2012. Springer. To appear.
14. Sumit K. Jha and et al. A Bayesian Approach to Model Checking Biological Systems. In *CMSB*, pages 218–234. Springer-Verlag, 2009.
15. H. Kahn. Stochastic (Monte Carlo) Attenuation Analysis. Technical Report P-88, Rand Corporation, July 1949.
16. S. Kullback. *Information Theory and Statistics*. Dover, 1968.
17. Marta Kwiatkowska, Gethin Norman, and David Parker. PRISM: Probabilistic Symbolic Model Checker. In Tony Field, Peter Harrison, Jeremy Bradley, and Uli Harder, editors, *Computer Performance Evaluation: Modelling Techniques and Tools*, volume 2324 of *LNCS*, pages 113–140. Springer Berlin / Heidelberg, 2002.
18. N. Metropolis and S. Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341, September 1949.
19. Ad Ridder. Importance sampling simulations of markovian reliability systems using cross-entropy. *Annals of Operations Research*, 134:119–136, 2005.
20. Ad Ridder. Asymptotic optimality of the cross-entropy method for markov chain problems. *Procedia Computer Science*, 1(1):1571 – 1578, 2010.
21. Gerardo Rubino and Bruno Tuffin, editors. *Rare Event Simulation using Monte Carlo Methods*. Wiley, 2009.
22. R. Rubinstein. The Cross-Entropy Method for Combinatorial and Continuous Optimization. 1:127–190, 1999.
23. K. Sen, M. Viswanathan, and G. A. Agha. VESTA: A statistical model-checker and analyzer for probabilistic systems. In *QEST*, pages 251–252. IEEE, September 2005.
24. Perwez Shahabuddin. Importance Sampling for the Simulation of Highly Reliable Markovian Systems. *Management Science*, 40(3):333–352, March 1994.
25. J. Shore and R. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on Information Theory*, 26(1):26–37, January 1980.
26. The PRISM website. www.prismmodelchecker.org.
27. H. Younes and R. Simmons. Probabilistic verification of discrete event systems using acceptance sampling. In *CAV*, volume 2404, pages 23–39. Springer, 2002.
28. Håkan Younes. YMER: A Statistical Model Checker. In Kousha Etessami and Sriram Rajamani, editors, *CAV*, volume 3576 of *LNCS*, pages 171–179. Springer Berlin / Heidelberg, 2005.