Importance Splitting for Statistical Model Checking Rare Properties

Cyrille Jegourel, Axel Legay, Sean Sedwards

To cite this version:

Cyrille Jegourel, Axel Legay, Sean Sedwards. Importance Splitting for Statistical Model Checking Rare Properties. Computer Aided Verification, Jul 2013, Saint-Pétersbourg, Russia. pp.576 - 591, 10.1007/978-3-642-39799-8_38 . hal-01087826

HAL Id: hal-01087826
https://hal.inria.fr/hal-01087826

Submitted on 26 Nov 2014

Importance Splitting for Statistical Model Checking Rare Properties

Cyrille Jegourel, Axel Legay and Sean Sedwards

{cyrille.jegourel,axel.legay,sean.sedwards}@inria.fr

Abstract Statistical model checking avoids the intractable growth of states associated with probabilistic model checking by estimating the probability of a property from simulations. Rare properties are often important, but pose a challenge for simulation-based approaches: the relative error of the estimate is unbounded. A key objective for statistical model checking rare events is thus to reduce the variance of the estimator. *Importance splitting* achieves this by estimating a sequence of conditional probabilities, whose product is the required result. To apply this idea to model checking it is necessary to define a *score function* based on logical properties, and a set of *levels* that delimit the conditional probabilities. In this paper we motivate the use of importance splitting for statistical model checking and describe the necessary and desirable properties of score functions and levels. We illustrate how a score function may be derived from a property and give two importance splitting algorithms: one that uses fixed levels and one that discovers optimal levels adaptively.

1 Introduction

Model checking offers the possibility to verify the correctness of complex systems in an automatic way [6]. This concept has now been extended to probabilistic systems, where some or all non-determinism is resolved to probabilities or stochastic rates [1]. This extension is of fundamental importance, since in many practical applications it is necessary to quantify the probability of a property (e.g., system failure) or the expectation of an amount (e.g., the yield of a process).

To give results with certainty, model checking algorithms effectively perform an exhaustive traversal of the state space of the system. In most real applications, however, the state space is intractable, scaling exponentially with the number of interacting components. Abstraction and symmetry reduction may make certain classes of systems tractable, but are not generally applicable. This limitation has prompted the development of *statistical* model checking, that employs an executable model of the system to estimate the probability of a property from a number of independent simulations.

Statistical model checking is a Monte Carlo method [17] that takes advantage of robust statistical techniques to bound the error of the estimated result (e.g., [5,23]). To quantify a property it is necessary to observe the property and increasing the number of observations generally increases the confidence of the estimate. Rare properties thus pose a problem to statistical model checking, since they are difficult to observe and often highly relevant to system performance (e.g., system failure is usually required to be rare). Fortunately, many Monte Carlo methods for rare events were devised in the early days of computing. In particular, *importance sampling* [13,15] and *importance splitting* [14,15,20] may be successfully applied to statistical model checking.

Importance sampling and importance splitting have been widely applied to specific simulation problems in science and engineering. Importance sampling works by estimating a result using biased simulations and compensating for the bias. Importance splitting works by reformulating the rare probability as a product of less rare probabilities conditioned on levels that must be achieved.

Earlier work [22,10] extended the original applications of importance splitting to more general problems of computational systems. Recent work has explicitly considered the use of importance sampling in the context of statistical model checking [18,11,2,12]. In what follows, we describe some of the limitations of importance sampling and motivate the use of importance splitting applied to statistical model checking, linking the concept of levels and score functions to temporal logic.

The remainder of the paper is organised as follows. Section 2 defines the basic notions and notation required in the sequel. Section 3 discusses statistical model checking rare events and introduces importance sampling and splitting. Section 4 defines the important properties of score functions (required to define levels) and describes how such functions may be derived from logical properties. Section 5 gives two importance splitting algorithms, while Section 6 illustrates their use on several examples, using different score functions.

2 Preliminaries

We consider stochastic discrete-event systems. This class includes any stochastic process that can be thought of as occupying a single state for a duration of time before an instantaneous transition to a new state. In particular, we consider systems described by discrete and continuous time Markov chains. Sample execution paths can be generated through discrete-event simulation (e.g., [9]). Execution paths are sequences of the form $\omega = s_0 \overset{t_0}{\rightarrow} s_1 \overset{t_1}{\rightarrow} s_2 \overset{t_2}{\rightarrow} ...$, 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 path $\omega = s_0 s_1 ...$. The length of path ω includes the initial state and is denoted $|\omega|$. A prefix of ω is a sequence $\omega_{\leq k} = s_0 s_1 ... s_k$ with $k < |\omega| \in \mathbb{N}$. We denote by $\omega_{\geq k}$ the suffix of ω starting at s_k.

2.1 Temporal logic

A simulation trace results from an execution of the system and is a finite sequence of visited states labelled with either discrete time step numbers or real times that are accumulated random delays. The discrete or continuous time

Markov chains that describe the system may be infinite. The process of statistical model checking estimates the probability that a system satisfies a property from the number of simulation traces within a sample, which individually satisfy the property. In this paper we consider properties specified with time bounded temporal logic, having the following abstract syntax:

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

α is an *atomic proposition* that may be true (denoted $\top$) or false in any state $s \in S$. $\vee$, $\wedge$ 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 $t \in \mathbb{R}$ may denote steps or real time and the interval is relative to the interval of any enclosing operator. 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 an arbitrary suffix $\omega_{\geq k}$ and a property φ with syntax (1), the semantics of $\omega_{\geq k} \models \varphi$ is defined:

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

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

3 Statistical model checking rare events

We consider a stochastic system $\mathcal{S}$ and a temporal logic property φ that may be true or false with respect to an execution trace. Our objective is to calculate the probability γ that an arbitrary execution trace ω satisfies φ, denoted $\gamma = \mathrm{P}(\omega \models \varphi)$. To decide the truth of a particular trace ω', we define a model checking function $z(\omega) \in \{0, 1\}$ that takes the value 1 if $\omega' \models \varphi$ and 0 if $\omega' \not\models \varphi$.

Let Ω be the set of paths induced by $\mathcal{S}$, with $\omega \in \Omega$ and f a probability measure over Ω. Then

$$\gamma = \int_{\Omega} z(\omega)\,\mathrm{d}f \qquad (2) \qquad \text{and} \qquad \gamma \approx \frac{1}{N}\sum_{i=1}^{N} z(\omega_i)$$

N denotes the number of simulations 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 $\mathrm{Var}(\gamma) = \gamma(1-\gamma)$ and for $\gamma \to 0$, $\mathrm{Var}(\gamma) \approx \gamma$.

When a property is not rare there are useful bounding formulae (e.g., the Chernoff bound [5]) that relate *absolute* error, confidence and the required number of simulations to achieve them. 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. For a Bernoulli random variable the relative error is given by $\sqrt{\gamma(1-\gamma)}/\gamma$, that is unbounded as $\gamma \to 0$. In standard Monte Carlo simulation, γ is the expected fraction of executions in which the rare event will occur. If the number of simulation runs is significantly less than $1/\gamma$, as is necessary when γ is very small, no occurrences of the rare property will likely be observed. A number of simulations closer to $100/\gamma$ is desirable to obtain a reasonable estimate. Hence, the following techniques have been developed to reduce the number of simulations required or, equivalently, to reduce the variance of the rare event and so achieve greater confidence for a given number of simulations.

3.1 Importance Sampling

Importance sampling works by biasing the system dynamics in favour of a property of interest, simulating under the new dynamics, then unbiasing the result to give a true estimate. Referring to (2), let f' be another probability measure over Ω, absolutely continuous with respect to zf, then (2) can be rewritten

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

where $L = \mathrm{d}f/\mathrm{d}f'$ is the *likelihood ratio*. We can thus estimate γ by simulating under f' and compensating by L: $\gamma \approx \frac{1}{N}\sum_{i=1}^{N} L(\omega_i) z(\omega_i)$. $L(\omega_i)$ may be calculated with little overhead during individual simulation runs.

In general, the importance sampling measure f' is chosen to produce the rare property more frequently, but this is not the only criterion. The optimal importance sampling measure, denoted f^* and defined as f conditioned on the rare event, is exactly the distribution of the rare event: $f^* = zf/\gamma$. The challenge of importance sampling is to find a good *change of measure*, i.e., a measure f' that is close to f^*. An apparently good change of measure may produce the rare property more frequently (thus reducing the variance with respect to the *estimated* value) but increase the variance with respect to the *true* value. In [12] we describe an efficient algorithm to find a change of measure that avoids this phenomenon.

It remains an open problem with importance sampling to quantify the performance of apparently 'good' distributions. A further challenge arises from properties and systems that require long simulations. In general, as the length of a path increases, its probability diminishes exponentially, leading to very subtle differences between f and f' and consequent problems of numerical precision.

3.2 Importance splitting

The earliest application of importance splitting is perhaps that of [14], where it was used to calculate the probability that neutrons would pass through certain shielding materials. This physical example provides a convenient analogy for the more general case. The system comprises a source of neutrons aimed at one side of a shield of thickness T. It is assumed that neutrons are absorbed by random interactions with the atoms of the shield, but with some small probability γ it is possible for a neutron to pass through the shield. The distance travelled in the shield can then be used to define a set of increasing levels $l_0 = 0 < l_1 < l_2 < \cdots < l_n = T$ that may be reached by the paths of neutrons, with the property that reaching a given level implies having reached all the lower levels. Though the overall probability of passing through the shield is small, the probability of passing from one level to another can be made arbitrarily close to 1 by reducing the distance between the levels.

These concepts can be generalised to simulation models of arbitrary systems, where a path is a simulation trace. By denoting the abstract level of a path as l, the probability of reaching level l_i can be expressed as $\mathrm{P}(l > l_i) = \mathrm{P}(l > l_i \mid l > l_{i-1})\mathrm{P}(l > l_{i-1})$. Defining $\gamma = \mathrm{P}(l > l_n)$ and observing $\mathrm{P}(l > l_0) = 1$, it is possible to write

$$\gamma = \prod_{i=1}^{n} \mathrm{P}(l > l_i \mid l > l_{i-1}) \tag{3}$$

Each term of the product (3) is necessarily greater than or equal to γ. The technique of importance splitting thus uses (3) to decompose the simulation of a rare event into a series of simulations of conditional events that are less rare. There have been many different implementations of this idea, but a generalised procedure is as follows.

Assuming a set of increasing levels is defined as above, a number of simulations are generated, starting from a distribution of initial states that correspond to reaching the current level. The procedure starts by estimating $\mathrm{P}(l \geq l_1 | l \geq l_0)$, where the distribution of initial states for l_0 is usually given (often a single state). Simulations are stopped as soon as they reach the next level; the final states becoming the empirical distribution of initial states for the next level. Simulations that do not reach the next level (or reach some other stopping criterion) are discarded. In general, $\mathrm{P}(l \geq l_i | l \geq l_{i-1})$ is estimated by the number of simulation traces that reach l_i, divided by the total number of traces started from l_{i-1}. Simulations that reached the next level are continued from where they stopped. To avoid a progressive reduction of the number of simulations, the generated distribution of initial states is sampled to provide additional initial states for new simulations, thus replacing those that were discarded.

In physical and chemical systems, distances and quantities may provide a natural notion of level that can be finely divided. In the context of model-checking arbitrary systems, variables may be Boolean and temporal properties may not contain an obvious notion of level. To apply importance splitting to statistical model checking it is necessary to define a set of levels based on a sequence of temporal properties, φ_i, that have the logical characteristic

$$\varphi = \varphi_n \Rightarrow \varphi_{n-1} \Rightarrow \cdots \Rightarrow \varphi_0$$

Each φ_i is a strict restriction of the property φ_{i-1}, formed by the conjunction of φ_i with property ψ_i, such that $\varphi_i = \varphi_{i-1} \wedge \psi_i$, with $\varphi_0 \equiv \top$. Hence, φ_i can be written $\varphi_i = \bigwedge_{j=1}^{i} \psi_j$. This induces a strictly nested sequence of sets of paths $\Omega_i \subseteq \Omega$:

$$\Omega_n \subset \Omega_{n-1} \subset \cdots \subset \Omega_0$$

where $\Omega_i = \{\omega \in \Omega : \omega \models \varphi_i\}$, $\Omega_0 \equiv \Omega$ and $\forall \omega \in \Omega, \omega \models \varphi_0$. Thus, for arbitrary $\omega \in \Omega$,

$$\gamma = \prod_{i=1}^{n} \mathrm{P}(\omega \models \varphi_i \mid \omega \models \varphi_{i-1}),$$

that is analogous to (3).

A statistical model checker implementing bounded temporal logic will generally assign variables to track the status of the time bounds of temporal operators. Importance splitting requires these variables to be included as part of the state that is stored when a trace reaches a given level.

The choice of levels is crucial to the effectiveness of importance splitting. To minimise the relative variance of the final estimate it is desirable to choose levels that make $\mathrm{P}(\omega \models \varphi_i \mid \omega \models \varphi_{i-1})$ the same for all i (see, e.g., [7]). A simple decomposition of a property may give levels with widely divergent conditional probabilities, hence Section 4 introduces the concept of a *score function* and techniques that may be used to increase the possible resolution of levels. Given sufficient resolution, a further challenge is to define the levels. In practice, these are often guessed or found by trial and error, but Section 5.2 gives an algorithm that finds optimal levels adaptively.

4 Score functions

Score functions generalise the concept of levels described in Section 3.2.

Definition 1. *Let $J_0 \supset J_1 \supset \ldots \supset J_n$ be a set of nested intervals of $\mathbb{R}$ and let $\varphi_0 \Leftarrow \varphi_1 \Leftarrow \cdots \Leftarrow \varphi_n = \varphi$ be a set of nested properties. The mapping $\Phi : \Omega \to \mathbb{R}$ is a* level-based *score function of property φ if and only if $\forall k : \omega \models \varphi_k \iff \Phi(\omega) \in J_k$ and $\forall i, j \in \{0, \ldots, |\omega|\} : i < j \implies \Phi(\omega_{\leq i}) \leq \Phi(\omega_{\leq j})$*

In general, the aim of a score function is to discriminate good paths from bad with respect to a property. In the case of a level-based score function, paths that have a higher score are clearly better because they satisfy more of the overall property. Given a nested sequence of properties $\varphi_0 = \top \Leftarrow \varphi_1 \Leftarrow \cdots \Leftarrow \varphi_n = \varphi$, a simple score function may be defined

$$\Phi(\omega) = \sum_{k=1}^{n} \mathbb{1}(\omega \models \varphi_k) \tag{4}$$

$\mathbb{1}(\cdot)$ is an indicator function taking the value 1 when its argument is true and 0 otherwise. Various ways to decompose a logical property are given in Section 4.1.

While a level-based score function directly correlates logic to score, in many applications the property of interest may not have a suitable notion of levels to exploit; the logical levels may be too coarse or may distribute the probability unevenly. For these cases it is necessary to define a more general score function.

Definition 2. *Let $J_0 \supset J_1 \supset ... \supset J_n$ a set of nested intervals of $\mathbb{R}$ and $\Omega = \Omega_0 \supset \Omega_1 \supset \cdots \supset \Omega_n$ a set of nested subsets of Ω. The mapping $\Phi : \Omega \to \mathbb{R}$ is a* general *score function of property φ if and only if $\forall k : \omega \in \Omega_k \iff \Phi(\omega) \in J_k$ and $\omega \models \varphi \iff \omega \in \Omega_n$ and $\forall i, j \in \{0, \dots, |\omega|\} : i < j \implies \Phi(\omega_{\leq i}) \leq \Phi(\omega_{\leq j})$*

Informally, Definition 2 states that a general score function requires that the highest scores be assigned to paths that satisfy the overall property and that the score of a path's prefix is non-decreasing with increasing prefix length.

When no formal levels are available, an effective score function may still be defined using heuristics, that only loosely correlate increasing score with increasing probability of satisfying the property. For example, a time bounded property, not explicitly correlated to time, may become increasingly less likely to be satisfied as time runs out (i.e., with increasing path length). The heuristic in this case would assign higher scores to shorter paths. A score function based on coarse logical levels may be improved by using heuristics between the levels.

4.1 Decomposition of a temporal logic formula

Many existing uses of importance splitting employ a natural notion of levels inherent in a specific problem. Systems that do not have an inherent notion of level may be given quasi-natural levels by 'lumping' states of the model into necessarily consecutive states of an abstracted model. This technique is used in the dining philosophers example in Section 6.2.

For the purposes of statistical model checking, it is necessary to link levels to temporal logic. The following subsections describe various ways a logical formula may be decomposed into subformulae that may be used to form a level-based score function. The techniques may be used independently or combined with each other to give the score function greater resolution. Hence, the term 'property' used below refers both to the overall formula and its subformulae.

Since importance splitting depends on successively reaching levels, the initial estimation problem tends to become one of reachability (as in the case of numerical model checking algorithms). We observe from the following subsections that this does not necessarily limit the range of properties that may be considered.

Simple decomposition When a property φ is given as an explicit conjunction of n sub-formulae, i.e., $\varphi = \bigwedge_{j=1}^{n} \psi_j$, a simple decomposition into nested properties is obtained by $\varphi_i = \bigwedge_{j=1}^{i} \psi_j, \forall i \in \{1, \dots, n\}$, with $\varphi_0 \equiv \top$. The associativity and commutativity of conjunction make it possible to choose an arbitrary order of sub-formulae, with the possibility to choose an order that creates levels with equal conditional probabilities. Properties that are not given as conjunctions may be re-written using DeMorgan's laws in the usual way.

Natural decomposition Many rare events are defined with a natural notion of level, i.e., when some quantity in the system reaches a particular value. In physical systems such a quantity might be a distance, a temperature or a number of molecules. In computational systems, the quantity might refer to a loop counter, a number of software objects, or the number of available servers, etc.

Natural levels are thus defined by nested atomic properties of the form $\varphi_i = (l > l_i), \forall i \in \{0, \dots, n\}$, where l is a state variable, $l_0 = 0 < l_1 < \cdots < l_n$ and $\omega \models \varphi_n \iff l \geq l_n$. When rarity increases with decreasing natural level, the nested properties have the form $\varphi_i = l > l_i, \forall i \in \{0, \dots, n\}$, with $l_0 = \max(l) > l_1 > \cdots > l_n$, such that $\omega \models \varphi_n \iff l \leq l_n$.

Time may be considered as a natural level if it also happens to be described by a state variable, however in the following subsection it is considered in terms of the bound of a temporal operator.

Decomposition of temporal operators The following Propositions hold:

1. $(\varphi_n \Rightarrow \varphi_{n-1}) \Longrightarrow (\mathbf{F}^{\leq t}\varphi_n \Rightarrow \mathbf{F}^{\leq t}\varphi_{n-1})$
2. $(\varphi_n \Rightarrow \varphi_{n-1}) \Longrightarrow (\mathbf{G}^{\leq t}\varphi_n \Rightarrow \mathbf{G}^{\leq t}\varphi_{n-1})$
3. $(\varphi_n \Rightarrow \varphi_{n-1}) \Longrightarrow (\mathbf{X}\varphi_n \Rightarrow \mathbf{X}\varphi_{n-1})$
4. $(\varphi_n \Rightarrow \varphi_{n-1} \wedge \psi_m \Rightarrow \psi_{m-1}) \Longrightarrow (\varphi_n \mathbf{U} \psi_m \Rightarrow \varphi_{n-1} \mathbf{U} \psi_{m-1})$
5. $(\varphi_n \Rightarrow \varphi_{n-1}) \Longrightarrow (\mathbf{F}^{\leq t}\mathbf{G}^{\leq s}\varphi_n \Rightarrow \mathbf{F}^{\leq t}\mathbf{G}^{\leq s}\varphi_{n-1})$
6. $(\varphi_n \Rightarrow \varphi_{n-1}) \Longrightarrow (\forall \omega \models \mathbf{G}^{\leq t}\varphi_n : \exists t' \geq t \mid \omega \models \mathbf{G}^{\leq t'}\varphi_{n-1})$
7. $(\varphi_n \Rightarrow \varphi_{n-1}) \Longrightarrow (\forall \omega \models \mathbf{F}^{\leq t}\varphi_n : \exists t' \leq t \mid \omega \models \mathbf{F}^{\leq t'}\varphi_{n-1})$
8. $(t' \geq t) \Longrightarrow (\mathbf{F}^{\leq t}\mathbf{G}^{\leq s}\varphi_n \Rightarrow \mathbf{F}^{\leq t'}\mathbf{G}^{\leq s}\varphi_n)$
9. $(s' \leq s) \Longrightarrow (\mathbf{F}^{\leq t}\mathbf{G}^{\leq s}\varphi_n \Rightarrow \mathbf{F}^{\leq t}\mathbf{G}^{\leq s'}\varphi_n)$
10. $(t' \geq t \wedge s' \leq s) \Longrightarrow (\mathbf{F}^{\leq t}\mathbf{G}^{\leq s}\varphi_n \Rightarrow \mathbf{F}^{\leq t'}\mathbf{G}^{\leq s'}\varphi_n)$
11. $(\varphi_n \Rightarrow \varphi_{n-1}) \Longrightarrow (\forall \omega \models \mathbf{F}^{\leq t}\mathbf{G}^{\leq s}\varphi_n : \exists t' \leq t \wedge s' \geq s \mid \omega \models \mathbf{F}^{\leq t'}\mathbf{G}^{\leq s'}\varphi_{n-1})$

Temporal decomposition From Proposition 6, properties having the form $\varphi = \mathbf{G}^t\psi$ may be decomposed in terms of t. For an arbitrary suffix $\omega_{\geq k} = s_k \overset{t_k}{\rightarrow} s_{k+1} \overset{t_{k+1}}{\rightarrow} s_{k+2} \overset{t_{k+2}}{\rightarrow} \cdots$, we have $(\omega_{\geq k} \models \mathbf{G}^t\psi) \leftrightarrow (\omega_{\geq k} \models \psi) \wedge (\omega_{\geq k+1} \models \psi) \wedge \cdots \wedge (\omega_{k+m} \models \psi)$, for some m such that $\sum_{j=k}^{m+k} t_j \leq t \wedge \sum_{j=k}^{m+k+1} t_j > t$. This has the form required for a simple decomposition, giving nested properties of the form $\varphi_i = \mathbf{G}^{l_i}\psi, \forall i \in \{1, \dots, n\}$, where $l_1 = 0 < l_2 < \cdots < l_n = t$, with $\varphi_0 \equiv \top$.

Properties having the form $\varphi = \mathbf{F}^t\psi$ evaluate to disjunctions in terms of time. From Proposition 7, it is plausible to construct nested properties of the form $\varphi_i = \mathbf{F}^{t+l_i}\psi, \forall i \in \{1, \dots, n\}$, with $l_1 > l_2 > \cdots > l_n = 0$ and $\varphi_0 \equiv \top$. Some caution is required if t is the value given in the overall property. If trace ω satisfies $\mathbf{F}^{t'}$ but not $\mathbf{F}^t$, any prefix of ω does not satisfy $\mathbf{F}^t$. The requirement for $\mathbf{F}^{t'}$ to have a lower score than $\mathbf{F}^t$ conflicts with the requirement of a score function $\forall i, j \in \{0, \dots, |\omega|\} : i < j \implies \Phi(\omega_{\leq i}) \leq \Phi(\omega_{\leq j})$.

Heuristic decomposition The decomposition of a property into logical levels may not necessarily result in an adequate score function: there may be insufficient levels, the levels may be irrelevant to the overall property or the levels may not evenly distribute the probability. In such cases it may be desirable to define intermediate levels based on heuristics – approximate correlations between a path and its probability to satisfy the property. For example, $\varphi_i = \mathbf{F}^{t+l_i}\psi$ may not form legitimate nested properties with positive l_i, but may nevertheless be used as a heuristic with $l_i \in [-t, 0]$.

Note that a heuristic score function that respects Definition 2 will give an unbiased estimate when used with an unbiased importance splitting algorithm. The effectiveness of a heuristic is dependent on how well it correlates path prefixes with the probability of eventually satisfying the overall property.

5 Importance splitting algorithms

We give two importance splitting pseudo-algorithms; one with fixed levels defined a priori and one that finds optimal levels adaptively. N denotes the number of simulations performed at each level. Levels, denoted τ, are defined as values of score function $\Phi(\omega)$, where ω is a path. τ_k is the k^{th} level and ω_i^k is the i^{th} simulation on level k. $\tilde{\gamma}_k$ is the estimate of γ_k, the k^{th} conditional probability $\mathrm{P}(\Phi(\omega) \geq \tau_k \mid \Phi(\omega) \geq \tau_{k-1})$.

5.1 Fixed level algorithm

The fixed level algorithm follows from the general description given in Section 3.2. Its advantages are that it is simple, it has low computational overhead and the resulting estimate is unbiased. Its disadvantage is that the levels must often be guessed by trial and error – adding to the overall computational cost.

In Algorithm 1, $\tilde{\gamma}$ is an unbiased estimate (see, e.g., [7]). Furthermore, from Proposition 3 in [3], we can deduce the following $(1-\alpha)$ confidence interval:

$$CI = \left[\tilde{\gamma}\left(\frac{1}{1+\frac{z_\alpha \sigma}{\sqrt{N}}}\right), \tilde{\gamma}\left(\frac{1}{1-\frac{z_\alpha \sigma}{\sqrt{N}}}\right)\right] \qquad \text{with} \qquad \sigma^2 \geq \sum_{k=1}^{M} \frac{1-\gamma_k}{\gamma_k}, \tag{5}$$

where z_α is the $1-\frac{\alpha}{2}$ quantile of the standard normal distribution. Hence, with confidence $100(1-\alpha)\%$, $\gamma \in CI$. σ is reduced by making all γ_k equal and large. For given γ, this implies increasing M, further motivating fine grained score functions. When it is not possible to define γ_k arbitrarily, the confidence interval may nevertheless be reduced by increasing N. The inequality for σ arises because the independence of initial states diminishes with increasing levels: unsuccessful traces are discarded and new initial states are drawn from successful traces. Several possibilities have been provided to minimise this dependence effect in [3]. In the following, for sake of simplicity, we assume that this goal is achieved. In the confidence interval, σ is estimated by the square root of $\sum_{k=1}^{M} \frac{1-\tilde{\gamma}_k}{\tilde{\gamma}_k}$.

```
Algorithm 1: Fixed levels
  Let $(\tau_k)_{1 \leq k \leq M}$ be the sequence of thresholds
  Let $stop$ be a termination condition
  $\forall j \in \{1, \dots, N\}$, set $\tilde{\omega}_j^1 = \emptyset$
  for $1 \leq k \leq M$ do
      $\forall j \in \{1, \dots, N\}$, using prefix $\tilde{\omega}_j^k$, generate path $\omega_j^k$ until $(\Phi(\omega_j^k) \geq \tau_k) \vee stop$
      $I_k = \{\forall j \in \{1, \dots, N\} : \Phi(\omega_j^k) \geq \tau_k\}$
      $\tilde{\gamma}_k = \frac{|I_k|}{N}$
      $\forall j \in I_k$, $\tilde{\omega}_j^{k+1} = \omega_j^k$
      $\forall j \notin I_k$, let $\tilde{\omega}_j^{k+1}$ be a copy of $\omega_i^k$ with $i \in I_k$ chosen uniformly randomly
  $\tilde{\gamma} = \prod_{k=1}^{M} \tilde{\gamma}_k$
```

5.2 Adaptive level algorithm

The cost of finding good levels must be included in the overall computational cost of importance splitting. An alternative to trial and error is to use an adaptive level algorithm that discovers its own optimal levels.

```
Algorithm 2: Adaptive levels
  Let $\tau_\varphi = \min\{\Phi(\omega) \mid \omega \models \varphi\}$ be the minimum score of paths that satisfy $\varphi$
  Let $N_k$ be the pre-defined number of paths to keep per iteration
  $k = 1$
  $\forall j \in \{1, \dots, N\}$, generate path $\omega_j^k$
  repeat
      Let $T = \left\{\Phi(\omega_j^k), \forall j \in \{1, \dots, N\}\right\}$
      Find minimum $\tau_k \in T$ such that $|\{\tau \in T : \tau > \tau_k\}| \geq N_k$
      $\tau_k = \min(\tau_k, \tau_\varphi)$
      $I_k = \{j \in \{1, \dots, N\} : \Phi(\omega_j^k) > \tau_k\}$
      $\tilde{\gamma}_k = \frac{|I_k|}{N}$
      $\forall j \in I_k$, $\omega_j^{k+1} = \omega_j^k$
      for $j \notin I_k$ do
          choose uniformly randomly $l \in I_k$
          $\tilde{\omega}_j^{k+1} = \max_{|\omega|} \left\{\omega \in pref(\omega_l^k) : \Phi(\omega) < \tau_k\right\}$
          generate path $\omega_j^{k+1}$ with prefix $\tilde{\omega}_j^{k+1}$
      $M = k$
      $k = k + 1$
  until $\tau_k > \tau_\varphi$;
  $\tilde{\gamma} = \prod_{k=1}^{M} \tilde{\gamma}_k$
```

Algorithm 2 is an adaptive level importance splitting algorithm based on [4]. It works by pre-defining a fixed number N_k of simulation traces to retain at each level. With the exception of the last level, the conditional probability of each level is then nominally N_k/N.

Use of the adaptive algorithm may lead to gains in efficiency (no trial and error, reduced overall variance), however the final estimate has a bias of order $\frac{1}{N}$, i.e., $\mathrm{E}(\tilde{\gamma}) = \gamma + \mathcal{O}(N^{-1})$. The overestimation (potentially not a problem when estimating rare critical failures) is negligible with respect to σ, such that the confidence interval remains that of the fixed level algorithm. Furthermore, under some regularity conditions, the bias can be asymptotically corrected. The estimate of γ has the form $r_0{\gamma_0}^{M_0}$, with $M_0 = M-1$, $r_0 = \gamma{\gamma_0}^{-M_0}$ and $\frac{E[\tilde{\gamma}]-\gamma}{\gamma} \sim \frac{M_0}{N}\frac{1-\gamma_0}{\gamma_0}$ when N goes to infinity. Using the expansion

$$\tilde{\gamma} = \gamma\left(1 + \frac{1}{\sqrt{N}}\sqrt{M_0\frac{1-\gamma_0}{\gamma_0} + \frac{1-r_0}{r_0}}Z + \frac{1}{N}M_0\frac{1-\gamma_0}{\gamma_0} + o\left(\frac{1}{N}\right)\right),$$

with Z a standard normal variable, $\tilde{\gamma}$ is corrected by dividing it by $1+\frac{M_0(1-\gamma_0)}{N\gamma_0}$. See [3] for more details.

6 Case study

We have adapted models from the literature to illustrate the use of importance splitting with statistical model checking. All simulations were performed using our statistical model checking platform PLASMA in which the previous algorithms have been implemented [11].

6.1 Biochemical network

The network of chemical reactions given below is typical of biochemical systems and demonstrates the potential of SMC to handle the enormous state spaces of biological models.

We consider a well stirred chemically reacting system comprising five reactants (molecules of type A, B, C, D, E), a dimerisation reaction (6) and two decay reactions (7,8).

$$A + B \xrightarrow{1} C \tag{6}$$

$$C \xrightarrow{1} D \tag{7}$$

$$D \xrightarrow{1} E \tag{8}$$

The semantics of (6) 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 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 1. A typical simulation run is illustrated in Figure 1. A and B combine rapidly to form C, that 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 total number of states is less than 10^9, but beyond the current practical capability of exhaustive probabilistic model checking. It is possible

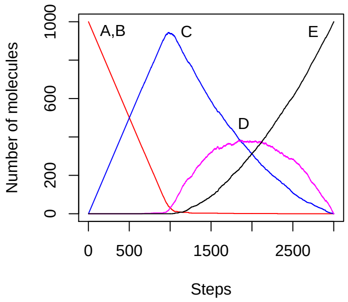

Figure 1. A typical stochastic simulation trace of reactions (6-8).

Probability	Estimate	$\sigma_{estimator}$
$\mathrm{P}(D > 390)$	0.182	0.012
$\mathrm{P}(D > 400 \mid D > 390)$	0.299	0.021
$\mathrm{P}(D > 410 \mid D > 400)$	0.201	0.019
$\mathrm{P}(D > 420 \mid D > 410)$	0.134	0.017
$\mathrm{P}(D > 430 \mid D > 420)$	0.088	0.016
$\mathrm{P}(D > 440 \mid D > 430)$	0.057	0.015
$\mathrm{P}(D > 450 \mid D > 440)$	0.035	0.012
$\mathrm{P}(D > 460 \mid D > 450)$	0.021	0.009
$\mathrm{P}(D > 460)$	8.1×10^{-9}	1.29×10^{-8}

Table 1. Chemical network conditional probability estimates based on 1000 runs of Algorithm 1 using $N = 1000$. $\sigma_{estimator}$ is estimated using the sample means.

for the number of molecules of D to reach 1000, however $D > 400$ is unusual. We thus define a suitably rare property to be $\varphi = \mathbf{F}^t D > 460$, with t initially 3000 steps, chosen to be adequately long. To apply Algorithm 1, we set $N = 1000$ and define a nested sequence of properties $\varphi_0 = \top$, $\varphi_i = \mathbf{F}^t D \geq \tau_i$, with $\tau_1 = 390$, $\tau_2 = 400$, $\tau_3 = 410$, $\tau_4 = 420$, $\tau_5 = 430$, $\tau_6 = 440$, $\tau_7 = 450$ and $\tau_8 = 460$. The score function is thus a mapping from paths to τ. τ_1 was found by trial and error, chosen to produce sufficient occurrences of the property on the first level. The other values are equally spaced.

We executed the algorithm 1000 times using the parameters given above. The results are given in Table 1. The standard deviation of the estimator, $\sigma_{estimator}$, is estimated in each case using the sample mean. An individual estimate is achieved with 8000 simulation runs; approx. 1.5×10^4 times fewer than the expected number to see a single instance of the rare property.

Algorithm 1 estimates $\mathrm{P}(D > 460) \approx 8.1 \times 10^{-9}$ with 8 levels, implying an optimal (to minimise variance) per-level conditional probability of approx. 0.097. Based on 100 executions, with $N = 1000$ and N_k thus set to 97, Algorithm 2 chose average levels $\hat{\tau}_1 = 396.0$, $\hat{\tau}_2 = 414.5$, $\hat{\tau}_3 = 426.3$, $\hat{\tau}_4 = 434.6$, $\hat{\tau}_5 = 441.8$, $\hat{\tau}_6 = 448.3$, $\hat{\tau}_7 = 454.1$ and $\hat{\tau}_8 = 459.0$. There is apparently some scope with this score function to increase the number of levels and thus increase the confidence of the estimate according to (5). This is left to a future investigation.

To compare the estimates of Algorithm 2 and Algorithm 1, we set $N = 1000$ and $N_k = 100$, giving a nominal conditional probability of 0.1 per level. The average levels chosen by Algorithm 2 under these circumstances were $\hat{\tau}_1 = 395.8$, $\hat{\tau}_2 = 414.0$, $\hat{\tau}_3 = 425.4$, $\hat{\tau}_4 = 433.7$, $\hat{\tau}_5 = 440.8$, $\hat{\tau}_6 = 447.3$, $\hat{\tau}_7 = 453.1$ and $\hat{\tau}_8 = 458.2$. These levels have fractionally closer spacing than those with $N_k = 97$, reflecting the marginally increased nominal per-level probability. With 1000 executions, Algorithm 2 estimates $\mathrm{P}(D > 460) \approx 1.4 \times 10^{-8}$, compared to the estimate of 8.1×10^{-9} with Algorithm 1. Given the estimated standard deviation of the fixed level estimator, this empirical difference is ascribed to statistical variance rather than the overestimate predicted by theory. Furthermore, a direct computation of the 95% confidence interval of Algorithm 1 shows that the estimate of Algorithm 2 is into it ($CI = [5 * 10^{-9}; 2.4 * 10^{-8}]$).

6.2 Dining philosophers

We construct a rare event based on the well known probabilistic solution [16] of Dijkstra's dining philosophers problem [8]. In this example, there are no natural counters to exploit, so levels must be constructed by considering 'lumped' states.

A number of philosophers sit at a circular table with an equal number of chopsticks; a chopstick being placed within reach of two adjacent philosophers. Philosophers think and occasionally wish to eat from a communal bowl. To eat, a philosopher must independently pick up two chopsticks: one from the left and one from the right. Having eaten, the philosopher replaces the chopsticks and returns to thinking. A problem of concurrency arises because a philosopher's neighbour(s) may have already taken the chopstick(s). Lehmann and Rabin's solution [16] is to allow the philosophers to make probabilistic choices.

We consider a model of 100 'free' philosophers [16]. The number of states in the model is approx. 10^{95}; 10^{15} times more than the estimated number of protons in the universe. The possible states of an individual philosopher can be abstracted to those shown in Fig. 2.

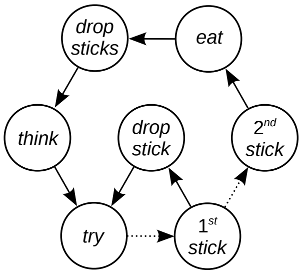

Figure 2. An abstract model of a dining philosopher.

Probability	Estimate	$\sigma_{estimator}$
P(*try*)	0.055	0.007
P(1^{st}*stick*|*try*)	0.029	0.006
P(2^{nd}*stick*|1^{st}*stick*)	0.017	0.005
P(*eat*|2^{nd}*stick*)	0.010	0.005
P(*drop sticks*|*eat*)	0.005	0.004
P(*drop sticks*)	1.7×10^{-9}	1.95×10^{-9}

Table 2. Dining philosophers conditional probability estimates based on 100 runs of Algorithm 1 with $N = 1000$. $\sigma_{estimator}$ is estimated using the sample means.

Thinking is the initial state of all philosophers. The transitions denoted by dotted lines in Figure 2 are dependent on the availability of chopsticks. All transitions are controlled by stochastic rates and made in competition with the transitions of other philosophers. With increasing numbers of philosophers, it is increasingly unlikely that a specific philosopher will be satisfied (i.e., that the philosopher will reach the state *drop sticks*) within a given number of steps from the initial state. We thus define a rare property $\varphi = \mathbf{F}^t$*drop sticks*, with t initially 7, denoting the property that a given philosopher will reach state *drop sticks* within 7 steps. Thus, using the states of the abstract model, we decompose φ into nested properties $\varphi_0 = \top$, $\varphi_1 = \mathbf{F}^t$*try*, $\varphi_2 = \mathbf{F}^t 1^{st}$*stick*, $\varphi_3 = \mathbf{F}^t 2^{nd}$*stick*,

$\varphi_4 = \mathbf{F}^t \mathsf{eat}$ and $\varphi_5 = \mathbf{F}^t \mathsf{drop\ sticks}$. The score function, not used explicitly here, is that defined by (4).

We executed Algorithm 1 100 times and obtained the results given in Table 2. The final estimate is achieved with approx. 10^5 fewer simulations than would be expected to see a single occurrence of the property using simple Monte Carlo.

6.3 Repair model

We consider a repair model from the rare event literature (Ex. 1 in [19]), which represents a class of systems that is known to be challenging for parametrised importance sampling; the use of 'group repair' causes them to be 'unbalanced' [19] and renders simple biasing schemes unable to bound the relative error [21].

The model comprises three types of components, with n components per type, that may fail and be repaired at certain probabilistic rates. Each type of component has a different rate of failing and components fail independently. The initial state has no failed components. Repairs are prioritised: components of type 1 are repaired before those of type 2 and type 2 are repaired before type 3. There is a common repair rate, but types 1 and 2 are repaired in groups (all failed components are repaired in one event) while type 3 are repaired singly.

We consider the *total failure entrance probability* (the probability that all components fail, without the system returning to the initial state) expressed as $\gamma = \mathrm{P}(\omega \models \mathit{init} \wedge \mathbf{X}(\neg \mathit{init} \mathbf{U}^t \mathit{failure}))$, with t infinite. Let fail_1, fail_2 and fail_3 denote the instantaneous number of failed components of types 1, 2 and 3, respectively, then init is defined as $\mathit{fail}_1 = 0 \wedge \mathit{fail}_2 = 0 \wedge \mathit{fail}_3 = 0$ and $\mathit{failure}$ is defined as $\mathit{fail}_1 = n \wedge \mathit{fail}_2 = n \wedge \mathit{fail}_3 = n$. We set $n = 4$ to create a model with a rare event that is nevertheless tractable to numerical analysis. We thus find that $\gamma = 1.177 \times 10^{-7}$ to four significant figures.

The property $\varphi = \mathit{init} \wedge \mathbf{X}(\neg \mathit{init} \mathbf{U}^t \mathit{failure})$ has the form of a conjunction, but a simple decomposition is trivial. Using Proposition 3 we can decompose $\mathbf{X}$ and using Proposition 4 we can decompose $\mathbf{U}$. init is a conjunction, but is used negated so can not be usefully decomposed. $\mathit{failure}$ can be decomposed as a simple conjunction or in terms of natural levels of failed components. We combine these and consider nested properties based on the total number of failed components $\mathit{totalfail} = \mathit{fail}_1 + \mathit{fail}_2 + \mathit{fail}_3$. The score function is then just a mapping from paths to $\mathit{totalfail}$.

We thus define levels $\tau_0 = 0$, $\tau_1 = 2, \ldots,\ \tau_i = i + 1, \ldots,\ \tau_{11} = 12$ and construct nested properties of the form $\varphi_i = \mathit{init} \wedge \mathbf{X}(\neg \mathit{init} \mathbf{U}^t \mathit{totalfail} \geq \tau_i)$. We applied Algorithm 1 100 times and achieved the results shown in Table 3. Using the numerical model checker PRISM[1] to calculate the true probabilities, we calculate the standard deviations of our estimators ($\sigma_{estimator}$). We conclude that we are able to accurately estimate γ with approx. 800 fewer simulations than would be expected to produce a single example of the rare property.

The results are illustrated in Fig. 3, where the inset *box and whisker* plot shows the overall performance of the importance splitting estimator with respect

[1] www.prismmodelchecker.org

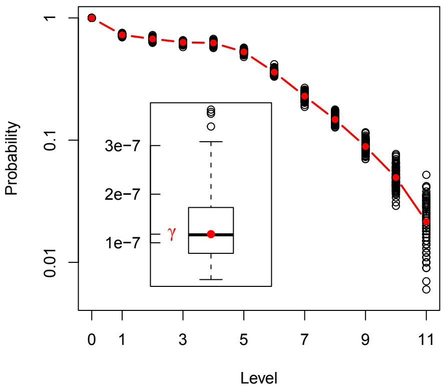

Figure 3. Estimated (black) and true (red) conditional probabilities for repair model (line only to guide the eye). Inset, overall estimate (black line) and true value (red dot).

Probability	Estimate	$\sigma_{estimator}$
$\mathrm{P}(\varphi_1 \mid \varphi_0)$	0.725	0.015
$\mathrm{P}(\varphi_2 \mid \varphi_1)$	0.673	0.016
$\mathrm{P}(\varphi_3 \mid \varphi_2)$	0.628	0.015
$\mathrm{P}(\varphi_4 \mid \varphi_3)$	0.622	0.019
$\mathrm{P}(\varphi_5 \mid \varphi_4)$	0.529	0.015
$\mathrm{P}(\varphi_6 \mid \varphi_5)$	0.360	0.017
$\mathrm{P}(\varphi_7 \mid \varphi_6)$	0.231	0.015
$\mathrm{P}(\varphi_8 \mid \varphi_7)$	0.149	0.011
$\mathrm{P}(\varphi_9 \mid \varphi_8)$	0.091	0.010
$\mathrm{P}(\varphi_{10} \mid \varphi_9)$	0.050	0.010
$\mathrm{P}(\varphi_{11} \mid \varphi_{10})$	0.023	0.009
$\mathrm{P}(\omega \models \varphi_{11})$	1.34×10^{-7}	8.12×10^{-8}

Table 3. Estimated conditional and overall probabilities for repair model, based on 100 runs of Algorithm 1 with $N = 1000$. $\sigma_{estimator}$ is calculated w.r.t. the true values.

to the true value of γ. The use of a logarithmic scale serves to demonstrate how the relative error increases with decreasing estimated probability, motivating the need to find optimal levels. Given the infinite time horizon of the property in this example, we hypothesise that it might be possible to use temporal decomposition to increase the granularity of the score function and thus balance the conditional probabilities of the levels. This is left to future work.

7 Conclusion

We have introduced the notion of using importance splitting with statistical model checking to verify rare properties. We have described how such properties must be decomposed to facilitate importance splitting and have demonstrated the procedures on several examples. We have described two importance splitting algorithms that may be constrained to give results within confidence bounds. Overall, we have shown that the application of importance splitting to statistical model checking has great potential.

References

1. C. Baier and J.-P. Katoen. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.
2. B. Barbot, S. Haddad, and C. Picaronny. Coupling and importance sampling for statistical model checking. In C. Flanagan and B. König, editors, *TACAS*, volume 7214 of *LNCS*, pages 331–346. Springer, 2012.
3. F. Cérou, P. Del Moral, T. Furon, and A. Guyader. Sequential Monte Carlo for rare event estimation. *Statistics and Computing*, 22:795–808, 2012.

4. F. Cérou and A. Guyader. Adaptive multilevel splitting for rare event analysis. *STOCHASTIC ANALYSIS AND APPLICATIONS*, 25:417–443, 2007.
5. H. 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.
6. E. M. Clarke, Jr., O. Grumberg, and D. A. Peled. *Model checking.* MIT Press, Cambridge, MA, USA, 1999.
7. P. Del Moral. *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications.* Probability and Its Applications. Springer, 2004.
8. E. W. Dijkstra. Hierarchical ordering of sequential processes. *Acta Informatica*, 1:115–138, 1971.
9. D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81:2340–2361, 1977.
10. P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. Multilevel splitting for estimating rare event probabilities. *Oper. Res.*, 47(4):585–600, Apr. 1999.
11. C. Jegourel, A. Legay, and S. Sedwards. A Platform for High Performance Statistical Model Checking – PLASMA. In C. Flanagan and B. König, editors, *TACAS*, volume 7214 of *LNCS*, pages 498–503. Springer, 2012.
12. C. Jegourel, A. Legay, and S. Sedwards. Cross-Entropy Optimisation of Importance Sampling Parameters for Statistical Model Checking. In P. Madhusudan and S. A. Seshia, editors, *CAV*, volume 7358 of *LNCS*, pages 327–342. Springer, 2012.
13. H. Kahn. Random sampling (Monte Carlo) techniques in neutron attenuation problems. *Nucleonics*, 6(5):27, 1950.
14. H. Kahn and T. E. Harris. Estimation of Particle Transmission by Random Sampling. In *Applied Mathematics*, volume 5 of *series 12*. National Bureau of Standards, 1951.
15. H. Kahn and A. W. Marshall. Methods of Reducing Sample Size in Monte Carlo Computations. *Operations Research*, 1(5):263–278, November 1953.
16. D. Lehmann and M. O. Rabin. On the Advantage of Free Choice: A Symmetric and Fully Distributed Solution to the Dining Philosophers Problem (Extended Abstract). In *Proc. 8th Ann. Symposium on Principles of Programming Languages*, pages 133–138, 1981.
17. N. Metropolis and S. Ulam. The Monte Carlo Method. *Journal of the American Statistical Association*, 44(247):335–341, September 1949.
18. D. Reijsbergen, P.-T. de Boer, W. Scheinhardt, and B. Haverkort. Rare event simulation for highly dependable systems with fast repairs. *Performance Evaluation*, 69(7–8):336 – 355, 2012.
19. A. Ridder. Importance sampling simulations of markovian reliability systems using cross-entropy. *Annals of Operations Research*, 134:119–136, 2005.
20. M. N. Rosenbluth and A. W. Rosenbluth. Monte Carlo Calculation of the Average Extension of Molecular Chains. *Journal of Chemical Physics*, 23(2), February 1955.
21. P. Shahabuddin. Importance Sampling for the Simulation of Highly Reliable Markovian Systems. *Management Science*, 40(3):333–352, 1994.
22. M. Villén-Altamirano and J. Villén-Altamirano. RESTART: A Method for Accelerating Rare Event Simulations. In J. W. Cohen and C. D. Pack, editors, *Queueing, Performance and Control in ATM*, pages 71–76. Elsevier, 1991.
23. A. Wald. Sequential Tests of Statistical Hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186, June 1945.