11-2018

# On the sequential massart algorithm for statistical model checking

Cyrille JEGOUREL

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

Jin Song DONG

# On the Sequential Massart Algorithm for Statistical Model Checking

Cyrille Jegourel[1(✉)], Jun Sun[1], and Jin Song Dong[2]

[1] Singapore University of Technology and Design, Singapore, Singapore
[2] Griffith University, Mount Gravatt, Australia
cyrille.jegourel@gmail.com, sunjunhqq@gmail.com, dongjs1@gmail.com

**Abstract.** Several schemes have been provided in Statistical Model Checking (SMC) for the estimation of property occurrence based on predefined confidence and *absolute* or *relative* error. Simulations might be however costly if many samples are required and the usual algorithms implemented in statistical model checkers tend to be conservative. Bayesian and rare event techniques can be used to reduce the sample size but they can not be applied without prerequisite or knowledge about the system under scrutiny. Recently, sequential algorithms based on Monte Carlo estimations and Massart bounds have been proposed to reduce the sample size while providing guarantees on error bounds which has been shown to outperform alternative frequentist approaches [15]. In this work, we discuss some features regarding the distribution and the optimisation of these algorithms.

## 1 Introduction

Statistical Model Checking (SMC) [22] is a formal verification method used to estimate quantitative properties of probabilistic systems by simulations sampled from an executable model of the system. Unlike other probabilistic Model Checking techniques, the results are not exact but given within predefined precision and confidence bounds that rely in general on the Monte Carlo method [5,18]. An important issue is to design algorithms that provide enough statistical evidence about the probabilistic occurrence of properties.

SMC was initially proposed to address the problem of verifying whether a property probability exceeds a threshold or not. This problem can be solved by using the sequential probability ratio test in hypothesis testing [21,22]. Other issues have been considered since, notably the estimation of the probability that a system property holds. In spite of similarities, the two problems are different and in what follows, we focus on the estimation problem. The need of rigorous sampling schemes have been addressed from the early days of SMC [10,22] to the more recent [8,9,15] just to cite a few. A key feature in designing a sampling procedure is to determine the number of simulations necessary to generate an estimation within acceptable margins of error and confidence.


T. Margaria and B. Steffen (Eds.): ISoLA 2018, LNCS 11245, pp. 287–304, 2018.
https://doi.org/10.1007/978-3-030-03421-4_19

In many case studies, reducing the sample size while rigorously guaranteeing the control of these error bounds is critical. For example, Secure Water Treatment (SWaT)[1] is a scaled-down but fully operational water treatment testbed at the Singapore University of Technology and Design, capable of producing five gallons of safe drinking water per minute. Probabilistic SWaT models have been designed to understand the response of SWaT to a variety of cyber attacks [4]. However, the simulations are costly and time-consuming. Then checking whether the probabilistic model is a faithful representation of this system is critical and must be done efficiently, under the constraint that the system is executed as few as possible.

Several classes of schemes such as Bayesian SMC [23], or rare event simulation techniques [12,13] have been considered to address this problem. However, in Bayesian SMC, the probability to estimate must be given by a prior random variable whose density is based on previous experiments and knowledge about the system. Rare event techniques can not be easily deployed for general problems and for arbitrary class of probabilistic systems. Also, these techniques require either the full knowledge of the probabilistic distribution of the system or the design of an accurate score function. Finally, the error bounds remain approximate in rare event simulation. These limitations motivate the recourse to sequential algorithms based on exact error bounds for frequentist estimations. This work is limited to this class of schemes.

In [10], the authors discussed the notion of *absolute* and *relative* margin of error for SMC. The absolute error is defined as the magnitude of the difference between a probability and an estimation of this probability whereas the relative error is defined as the absolute error divided by the magnitude of the probability. To guarantee that the absolute error is bounded, they introduced a procedure relying on the Okamoto bound[2] that, given fixed confidence and error parameters, determines *a priori* the number of Bernoulli samples required, which is independent of the probability to estimate. Supporting relative errors (i.e., errors which depend on the probability to quantify) is more difficult, although theoretical bounds exist. The relative error was initially handled by Dagum et al.'s algorithm [7].

In [15], new sequential sampling schemes based on Massart bounds and exact confidence intervals were proposed to handle absolute and relative error problems and were compared with other approaches, including some that have not been necessarily used in the context of SMC. We refer the readers to [15] for a comparison among these algorithms. The results were promising as they show that the Massart sequential sampling schemes outperformed the standard algorithms implemented in statistical model checkers like PRISM [16], PLASMA [11], APMC [10], COSMOS [2] and UPPAAL-SMC [8]. It is worth saying that these sequential Massart algorithms are not limited to a particular class of models and could be easily implemented in any of these statistical model checkers.

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

[2] The Okamoto bound is sometimes called the Chernoff bound in the literature.

In this work, we take the opportunity to discuss some features of the sequential Massart algorithms that were not initially considered and to explain with more details on how to set up the algorithms. First of all, given error and confidence parameters, $\epsilon$ and $\delta$, it is necessary to provide a third parameter $\alpha$ such that $0 < \alpha < \delta$. In [15], $\alpha$ was set by default at $\delta/2$. In an expanded version [14], we showed empirically that $\alpha$ could not be optimised a priori since it is dependent on the (unknown) probability to estimate. In this work, we give guidelines about setting $\alpha$ up and we show that the gains in terms of sample size reduction are quite significant if $\alpha$ is set up closer to zero.

These algorithms also require the recursive computation of confidence intervals. For the sake of rigorousness, these confidence intervals were initially chosen to be exact confidence intervals. However, these intervals are always rather conservative. Instead, approximate confidence intervals are in general easier to compute, their precision is better in the sense that their width is narrower but their confidence is not always strictly guaranteed. In what follows, we will consider two alternative approximate confidence intervals to measure empirically the impact on the sample sizes and the reliability of our algorithms.

Finally, an important aspect of SMC is that its performance can be improved by distributing the simulations on a multi-threaded system once the sample size of the experiment has been determined. For sequential algorithms, extra work must be done since the sample size is unknown a priori. In this work, we introduce a simple procedure to handle this issue.

In Sect. 2, we formally state the absolute and relative specifications that must be fulfilled by the SMC schemes. We also recall the basics of Monte Carlo estimation and Okamoto and Massart bounds. In Sect. 3, we present the sequential Massart algorithms. We discuss the coverage parameter in Sect. 4. We show in Sect. 5 the impact of approximate confidence intervals on the sampling size reduction. In Sect. 6, we propose a simple algorithm to distribute our sequential algorithms in a multi-threaded system. Section 7 concludes the article.

## 2 Background

In the following, a stochastic system $\mathcal{S}$ is interpreted as a set of interacting components in which the state is determined randomly with respect to a global probability distribution. Let $(\Omega, \mathcal{F}, \mu)$ be the probability space induced by the system with $\Omega$ a set of finite paths with respect to system's property $\phi$, $\mathcal{F}$ a $\sigma$-algebra of $\Omega$ and $\mu$ the probability distribution defined over $\mathcal{F}$.

### 2.1 Absolute and Relative Error Specifications

Given a probabilistic system $\mathcal{S}$, a property $\phi$ and a probability $\gamma$, we write $\mathcal{S} \models Pr(\phi) = \gamma$ if and only if the probability that a random execution of $\mathcal{S}$ satisfies $\phi$ is equal to $\gamma$. In principle, if $\gamma$ is unknown, we can apply analytical methods to determine this value. However, due for example to numerical imprecisions, we often relax the constraints over $\gamma$ and introduce the following notations:

$$\mathcal{S} \models^{a}_{\epsilon} Pr(\phi) = \gamma \quad \text{and} \quad \mathcal{S} \models^{r}_{\epsilon} Pr(\phi) = \gamma \tag{1}$$

The left formula means that a random execution of $\mathcal{S}$ satisfies $\phi$ with probability $\gamma$ plus or minus an absolute error $\epsilon$, i.e. $Pr(\phi) \in [\gamma - \epsilon, \gamma + \epsilon]$. The right formula means that a random execution of $\mathcal{S}$ satisfies $\phi$ with probability $\gamma$ up to some relative error $\epsilon$, i.e. $Pr(\phi) \in [(1-\epsilon)\gamma, (1+\epsilon)\gamma]$.

SMC applies to an executable system $\mathcal{S}$ and a property $\phi$ whose truth value can be determined in finite time. In SMC, the satisfaction of property $\phi$ is quantified by a Bernoulli random variable of unknown mean $\gamma$. This mean is then approximated using a Monte Carlo estimation scheme. The output of the scheme is thus not an exact value but an approximate one, given within certain error bounds and a confidence parameter $\delta$ that is the probability of outputting a false estimate. SMC thus requires a sampling scheme which outputs, after $n$ samples, an estimate $\hat{\gamma}_n$ close to $\gamma$ up to some absolute or relative $\epsilon$-based error with probability greater or equal than $1 - \delta$. Formally, we write:

$$\mathcal{S} \models^{a}_{\epsilon,\delta} Pr(\phi) = \hat{\gamma}_n \quad \text{or} \quad \mathcal{S} \models^{r}_{\epsilon,\delta} Pr(\phi) = \hat{\gamma}_n \tag{2}$$

if and only if an algorithm outputs estimators while guaranteeing:

$$Pr(|\hat{\gamma}_n - \gamma| > \epsilon) \leq \delta \tag{3}$$

or respectively:

$$Pr(|\hat{\gamma}_n - \gamma| > \epsilon\gamma) \leq \delta. \tag{4}$$

We call (3) the absolute error specification and (4) the relative error specification.

### 2.2 Monte Carlo Estimation

Let $\omega$ be a path sampled from space $\Omega$ with respect to distribution $\mu$; $z$ be a function from $\Omega$ to $\{0, 1\}$ assigning 1 if $\omega$ satisfies property $\phi$ and 0 otherwise; and $\gamma$ be the probability that an arbitrary path of the system satisfies $\phi$. In SMC, the behaviour of function $z$ is interpreted as a Bernoulli random variable $Z$ with mean parameter $\gamma$. By definition, the average value $\gamma$ is the integral of function $z$ with respect to distribution $\mu$ over space $\Omega$: $\gamma = \mathrm{E}_\mu[Z] = \int_\Omega z(\omega)\,\mathrm{d}\mu(\omega)$ and an estimator $\hat{\gamma}_n$ is given by the Monte Carlo method by drawing $n$ independent samples $\omega_i \sim \mu$, $i \in \{1, \ldots, n\}$, as follows:

$$\hat{\gamma}_n = \frac{1}{n} \sum_{i=1}^{n} z(\omega_i) \approx \mathrm{E}_\mu[Z] \tag{5}$$

Let $m = \sum_{i=1}^{n} z(\omega_i)$ be the number of successes and $\sigma^2 = \gamma(1-\gamma)$ the variance of $Z$. In what follows, for sake of simplicity, we use both notations $\hat{\gamma}_n$ and $m/n$ to denote the estimate.

The purpose of the algorithms presented in Sect. 3 is to fulfil Specification (3) or (4) with as few samples as possible. In other words, their goal is to improve the performance of statistical model checkers with algorithms that output reliable Monte Carlo estimates, in terms of precision and confidence. For this purpose, they make use of the bounds below.

### 2.3 Okamoto and Massart Bounds

In the literature, the Chernoff bounds [5] refer to exponential decreasing bounds, in the number of simulations, of the probability of deviation between a Monte Carlo estimate and its mean. Tighter bounds have been established since, notably in [17]. Note that in their original respective works, these bounds are only one-sided. In what follows, we give the two-sided versions of these bounds, for which the proofs can be found in the expanded version of [15][3].

**Absolute Error Bounds.** Though the seminal work is due to Chernoff [5], the following two-sided absolute error bound has been stated for binomial distributions by Okamoto in [19].

**Theorem 1 (Okamoto bound).** *For any $\epsilon$, $0 < \epsilon < 1$, we have the following inequality:*

$$Pr(|\hat{\gamma}_n - \gamma| > \epsilon) \leq 2\exp(-2n\epsilon^2) \tag{6}$$

Given $\epsilon$, $\delta$, writing out $\delta = 2\exp(-2n\epsilon^2)$, the Okamato bound can be used to determine a minimal number $n$ of simulations to perform a Monte Carlo plan fulfilling the absolute error specification (3). The main advantage of the Okamoto bound is that it does not depend on $\gamma$, the value to estimate. However, the bound is very conservative and in many cases, a much lower sample size would achieve the same absolute error specification.

Massart established in [17] a sharper bound that holds if the absolute error $\epsilon$ is lower than probabilities $\gamma$ and $1 - \gamma$.

**Theorem 2 (Absolute Error Massart bound).** *For all $\gamma$ such that $0 < \gamma < 1$ and any $\epsilon$ such that $0 < \epsilon < \min(\gamma, 1-\gamma)$, we have the following inequality:*

$$Pr(|\hat{\gamma}_n - \gamma| > \epsilon) \leq 2\exp\left(-n\epsilon^2 h_a(\gamma, \epsilon)\right) \tag{7}$$

$$\text{where} \quad h_a(\gamma, \epsilon) = \begin{cases} 9/2\ ((3\gamma + \epsilon)(3(1-\gamma) - \epsilon))^{-1} & if\ 0 < \gamma < 1/2 \\ 9/2\ ((3(1-\gamma) + \epsilon)(3\gamma + \epsilon))^{-1} & if\ 1/2 \leq \gamma < 1 \end{cases}$$

Figure 1 shows the number of samples per probability necessary to satisfy an absolute error specification defined by $\epsilon = 0.01$ and $\delta = 0.05$ according to the Okamoto and the Massart bounds. For values close to the boundaries, we can see that the Okamoto bound is very conservative in comparison of the Massart bound. However, the two bounds are similar for $\gamma = 1/2$.

[3] A journal version with the proofs is currently submitted [14]. The proofs are also available here: https://www.researchgate.net/publication/317823195_Sequential_Schemes_for_Frequentist_Estimation_of_Properties_in_Statistical_Model_Checking.

**Relative Error Bounds.** In practice, the absolute error is set independently of $\gamma$. However, it could be that the approximation is meaningless, especially if the absolute error is large with respect to $\gamma$. In this case, setting a relative error that remains 'small' with respect of $\gamma$ may be adequate. The Massart bound has a two-sided relative form.

**Theorem 3 (Relative Error Massart bound).** *For $\gamma$, $0 < \gamma < 1$ and any $\epsilon$, $0 < \epsilon < (1-\gamma)/\gamma$, we have the following inequality:*

$$Pr(|\hat{\gamma}_n - \gamma| \geq \epsilon\,\gamma) \leq 2\exp\left(-n\epsilon^2 h_r(\gamma,\epsilon)\right) \tag{8}$$

$$\textit{with} \quad h_r(\gamma,\epsilon) = \begin{cases} 9\gamma/2\ ((3+\epsilon)(3-\gamma(3+\epsilon)))^{-1} & \textit{if } 0 < \gamma < 1/2 \\ 9\gamma/2\ ((3-\epsilon)(3-\gamma(3-\epsilon)))^{-1} & \textit{if } 1/2 \leq \gamma < 1 \end{cases}$$

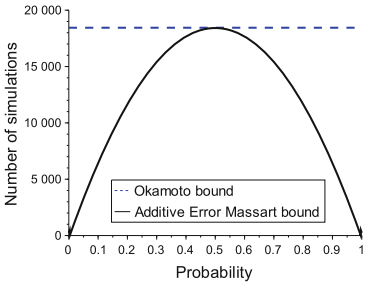



**Fig. 1.** Okamoto (dash) and Massart (plain) bounds with absolute error $\epsilon = 0.01$ and confidence parameter $\delta = 0.05$.

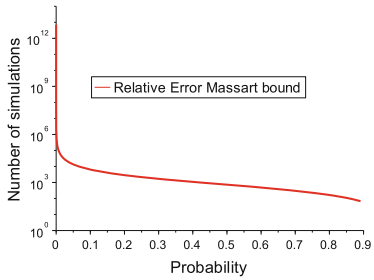



**Fig. 2.** Massart bounds with relative error $\epsilon = 0.1$ and confidence parameter $\delta = 0.05$.

Figure 2 shows the number of samples per probability necessary to satisfy a relative error specification defined by $\epsilon = 0.1$ and $\delta = 0.05$ according to the relative error Massart bound. As expected, the sample size explodes when $\gamma$ tends to zero.

### 2.4 Bounds with Coverage

In contrast to the Okamoto bound, the Massart bounds depend on $\gamma$ and they are thus not directly applicable since $\gamma$ is the probability that we want to estimate. However, one may still exploit some information about $\gamma$. For example, depending on the problem, one may know or numerically evaluate with certainty a rough interval in which $\gamma$ evolves. In what follows, we denote $C(\gamma, I)$ the notional coverage of $\gamma$ by a confidence interval $I$, that is the probability that $I$ contains $\gamma$.

**Theorem 4 (Absolute Error Massart Bound with coverage).** *Let $a$ and $b$ be the lower and upper bounds of a confidence interval $I$ and $I^c$ be the complement of $I$ in $[0,1]$:*

$$Pr\left(|\hat{\gamma}_n - \gamma| > \epsilon\right) \leq 2\exp\left(-n\epsilon^2 h_a(x,\epsilon)\right) + C(\gamma, I^c) \tag{9}$$

*where function $h_a$ is defined in Theorem 2 and $x = b$ if $b < 1/2$, $x = a$ if $a > 1/2$ and $x = 1/2$ if $1/2 \in I$.*

By default, $a = 0$, $b = 1$, $C(\gamma, [0,1]^c) = 0$ and the above theorem is consistent with the Okamoto bound. We remark that even if an accurate estimation of $\gamma$ is not feasible within a reasonable time, Theorem 4 can exploit coarse but exact bounds $a, b$, calculated analytically. In that case, we would have $C(\gamma, [a,b]^c) = 0$. Finally, a similar theorem involving relative error can be established.

**Theorem 5 (Relative Error Massart Bound with coverage).** *Let $a$ be the lower bound of a confidence interval $I = [a,1]$ and $h_r$ defined as in Theorem 3.*

$$Pr\left(|\hat{\gamma}_n - \gamma| > \epsilon\,\gamma\right) \leq 2\exp\left(-n\epsilon^2 h_r(a,\epsilon)\right) + C(\gamma, [0,a[) \tag{10}$$

Both theorems state that the probability of absolute or relative error is bounded by the respective Massart bound applied over the most pessimistic value of a confidence interval plus the probability that the interval does not contain $\gamma$. We deduce from both theorems the following sample-size result:

**Theorem 6.** *Let $\alpha < \delta$ such that $C(\gamma, I^c) < \alpha$. (i) Under the conditions of Theorem 4, a Monte Carlo algorithm $\mathcal{A}$ that outputs an estimate $\hat{\gamma}_n$ fulfils Specification (3) if $n > \frac{1}{h_a(x,\epsilon)\epsilon^2} \log \frac{2}{\delta-\alpha}$.*

*(ii) Similarly, under the conditions of Theorem 5, a Monte Carlo algorithm $\mathcal{A}$ that outputs an estimate $\hat{\gamma}_n$ fulfils Specification (4) if $n > \frac{1}{h_r(a,\epsilon)\epsilon^2} \log \frac{2}{\delta-\alpha}$.*

The bounds of Theorem 6 are more conservative than the bounds induced by Theorems 2 and 3 because the Massart bounds are evaluated using the most pessimistic value of the confidence interval $[a,b]$. In addition, our bound also takes into account the probability that $\gamma$ is not in $I$, implying that an additional number of samples are required in the final sample size. In the absolute error case, if a confidence interval $I$ containing $1/2$ is determined, applying the previous theorem is unnecessary because the sample size is simply bounded with respect to the Okamoto bound. Similarly, if $a$ (or $b$) is lower-bounded (or respectively upper-bounded) by $1/2$ but still close to $1/2$, the Okamoto bound is likely better. However, if $\gamma$ is closer to 0 or 1, the logarithmic extra number of samples is largely compensated by the evaluation of the Massart bound in $a$ or $b$.

## 3 Sequential Massart Algorithm

In this section, we recall the sequential Massart schemes for the absolute and relative error specifications. Both of them require three inputs: an error parameter

$\epsilon$, and two confidence parameters $\delta$ and $\alpha$ such that $\alpha < \delta$. To avoid confusion between $\delta$ and $\alpha$, below we call $\alpha$ the coverage parameter.

After each sample, we update a Monte Carlo estimator and a $(1-\alpha)$-confidence interval for $\gamma$. Then, the most pessimistic bound of the confidence interval is used in the Massart function to compute a new minimal sample size $n$ that satisfies Theorem 6. The process is repeated until the calculated sample size is lower than or equal to the current number of runs. In the pseudo-code of Algorithms 1 and 2, keywords GENERATE corresponds to a sample path generation and function CONFINT to the evaluation of the confidence interval (two-sided in the absolute error scheme but only one-sided in the relative error scheme). Theorems 4 and 5 guarantee the correctness of our schemes since, for any couple $(m, n)$, if we are able to compute a $(1-\alpha)$-confidence interval $I$ and its coverage, the deviation probability is bounded by $\delta$ defined as the sum of the coverage plus the Massart function evaluated at the most pessimistic value of $I$.

**Absolute Error Sequential Algorithm.** We initiate the algorithm with an interval $I_0$ in which $\gamma$ belongs (by default, $I_0 = [0, 1]$) and a worst-case $(\epsilon, \delta)$-sample size $n_0 = M$ with $M = \lceil \frac{1}{2\epsilon^2} \log \frac{2}{\delta} \rceil$ determined by the Okamoto bound (where $\lceil . \rceil$ denotes the ceiling function). Once a trace $\omega^{(k)}$ is generated and monitored, the number of successes with respect to property $\phi$ and the total number of traces are updated. Then, a $(1-\alpha)$-confidence interval $I_k$ is evaluated. Iteration after iteration, the interval width tends to shorten and becomes more and more accurate. Theorem 6-(i) is applied to determine a new sample size $n_k$, bounded from above by $M$ if necessary. These steps are repeated until $k \geq n_k$ at which Specification (3) is rigorously fulfilled.

**Relative Error Sequential Algorithm.** We first assume the existence, in a practical case study, of a threshold $\gamma_{min}$, supposedly low, corresponding to a tolerated precision error (e.g. a floating-point approximation). Estimating a value below $\gamma_{min}$ is then unnecessary. The maximal number of simulations is consequently bounded by the maximal Massart bound, $M = \lceil \frac{1}{\epsilon^2 h_r(\gamma_{min}, \epsilon)} \log \frac{2}{\delta} \rceil$. The relative error scheme is similar to the absolute error scheme. Note however that it is only necessary to determine a lower bound of $I_k$ since $h_r$ is a decreasing function in $\gamma$. Then, we determine a one-sided $(1-\alpha)$-interval of shape $[a_k, 1]$. Theorem 6-ii is applied to determine a new sample size $n_k$, upper bounded by $M$ if $a_k < \gamma_{min}$ and the steps are repeated until $k \geq n_k$. If the final output $\hat{\gamma}_k$ is higher than $\gamma_{min}$, Specification (4) is rigorously fulfilled. Otherwise, we can still output that $\gamma$ is lower than $\gamma_{min}$ with probability greater that $1 - \delta$.

## 4 Discussion on the Coverage Parameter

Coverage parameter $\alpha$ must be chosen such that $0 < \alpha < \delta$. Note that the sample sizes at which Specifications (3) and (4) are fulfilled are guaranteed to be lower or equal than the Okamoto and the maximal Massart bounds.

**Algorithm 1.** Absolute Error Sequential Algorithm

**Data**:
$\epsilon$, $\delta$, $\alpha$ : the original parameters
$M = \lceil \frac{1}{2\epsilon^2} \log \frac{2}{\delta} \rceil$: the Okamoto bound
$k = 0$
$m = 0$: the number of successes
$n_k = M$
$I_k = [a_k, b_k] = [0, 1]$: the initial interval in which $\gamma$ is known to belong

**while** $k < n_k$ **do**
  $k \leftarrow k + 1$
  GENERATE $\omega^{(k)}$
  $z(\omega^{(k)}) = \mathbb{1}(\omega^{(k)} \models \phi)$
  $m \leftarrow m + z(\omega^{(k)})$
  $I_k \leftarrow \text{CONFINT}(m, k, \alpha)$
  **if** $1/2 \in I_k$ **then**
    $n_k = M$
  **else if** $b_k < 1/2$ **then**
    $n_k = \lceil \frac{2}{h_a(b_k, \epsilon)\epsilon^2} \log \frac{2}{\delta - \alpha} \rceil$
  **else**
    $n_k = \lceil \frac{2}{h_a(a_k, \epsilon)\epsilon^2} \log \frac{2}{\delta - \alpha} \rceil$
  $n_k \leftarrow \min(n_k, M)$

**Output**: $\hat{\gamma}_k = m/k$

**Algorithm 2.** Relative Error Sequential Algorithm

**Data**:
$\epsilon$, $\delta$, $\alpha$, $\gamma_{min}$ : the original parameters
$M = \lceil \frac{1}{\epsilon^2 h_r(\gamma_{min}, \epsilon)} \log \frac{2}{\delta} \rceil$
$k = 0$
$n_k = M$
$I_k = [a_k, 1] = [\gamma_{min}, 1]$: the initial interval in which $\gamma$ is supposed to belong

**while** $k < n_k$ **do**
  $k \leftarrow k + 1$
  GENERATE $\omega^{(k)}$
  $z(\omega^{(k)}) = \mathbb{1}(\omega^{(k)} \models \phi)$
  $m \leftarrow m + z(\omega^{(k)})$
  $I_k \leftarrow \text{CONFINT}(m, k, \alpha)$
  **if** $\gamma_{min} \geq a_k$ **then**
    $n_k = M$
  **else**
    $n_k = \lceil \frac{1}{\epsilon^2 h_r(a_k, \epsilon)} \log \frac{2}{\delta - \alpha} \rceil$
  $n_k \leftarrow \min(n_k, M)$

**Output**: $\hat{\gamma}_k = m/k$

If $\alpha$ tends to zero, the $(1-\alpha)$-confidence interval converges to $[0,1]$. In the absolute error case, since $1/2$ belongs to the confidence interval, $h_a(x,\epsilon) = h_a(1/2,\epsilon) = 2$. Then, according to Theorem 6-(i), Specification (3) is fulfilled when $n$ is greater than $\frac{1}{2\epsilon^2}\log\frac{2}{\delta}$, that is equivalent to the Okamoto bound. In the relative error case, the sample size fulfilling Specification (4) tends to infinity because $h_r(a,\epsilon)$ tends to zero when $a$ tends to zero. As mentioned previously, $n$ however can be bounded in practice by $M = \lceil \frac{1}{\epsilon^2 h_r(\gamma_{min},\epsilon)}\log\frac{2}{\delta}\rceil$. In both cases, setting $\alpha$ too close to zero thus does not improve the predetermined bounds. Similarly, when $\alpha$ tends to $\delta$, $\log\frac{2}{\delta-\alpha}$ tends to infinity. Consequently, the sample sizes are respectively bounded by the Okamato bound and $M$ in the absolute and relative error case.

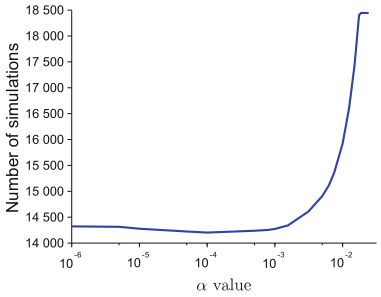
However, determining *a priori* a value for $\alpha$ that would minimise the sample size on average is a conundrum. A closed-form expression would depend on $\epsilon$ and $\delta$, but also on probability of interest $\gamma$. Given $\epsilon = 0.01$ and $\delta = 0.05$, Fig. 3 shows for different probabilities the sample size (averaged over 150 experiments) necessary to achieve Specifications (3) and (4) with various $\alpha$. In the absolute error case, the minimal sample size is empirically achieved for $10^{-4} < \alpha < 10^{-3}$ given $\gamma = 0.02$ and for $0.01 < \alpha < 0.015$ given $\gamma = 0.25$. Similarly, in the relative error case, the minimal sample size is achieved for $0.0015 < \alpha < 0.003$ given $\gamma = 0.1$ and for $0.006 < \alpha < 0.0125$ given $\gamma = 0.7$.

Since $\gamma$ impacts the choice of an optimal $\alpha$ but is unknown, it is not possible to optimise $\alpha$ a priori. Though the empirical observations cannot be generalised to any triples $(\epsilon,\delta,\gamma)$, it is worth remarking that all our results suggest a quicker convergence to the maximal bound when $\alpha$ converges to $\delta$ than when $\alpha$ converges to zero. This comes from the logarithmic speed of convergence in $\alpha$ of the confidence interval to the estimate given fixed number of samples and successes.

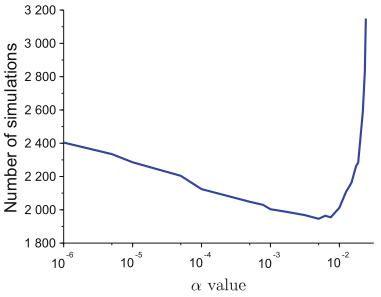
Figure 4 shows how the bounds of a (Wald) $(1-\alpha)$-confidence interval evolves when $\alpha$ varies between $10^{-5}$ and 0.05, given two different estimates $m/n$. The figure would be similar with any other intervals described in Sect. 5. When $\alpha$ is low, the variations in the bounds of the confidence interval are more important. But when $\alpha$ tends to 0.05, the variations are smoother and the width of the intervals does not vary much. So, the Massart function at the bounds of the confidence interval does not vary much as well in this case.

## 5 Approximate Versus Exact Confidence Intervals

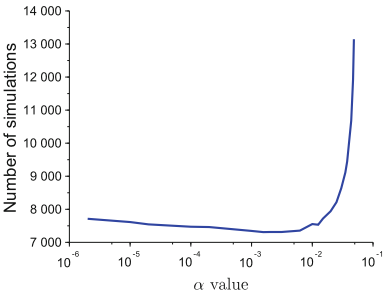
At line 6 of Algorithms 1 and 2, we iteratively compute an intermediate $(1-\alpha)$-confidence interval for $\gamma$. For the sake of rigorousness, we initially used exact Clopper-Pearson confidence intervals [6]. This confidence interval is directly derived from the binomial distribution and not from its approximation. It guarantees that the actual coverage is always equal to or above the nominal confidence level. In others words, a $(1-\alpha)$-Clopper-Pearson confidence interval $I_{CP}$ guarantees that $C(\gamma, I_{CP}) \geq 1-\alpha$ and its closed-form expression can be easily computed: $I_{CP} = [\,\beta^{-1}\left(\frac{\alpha}{2}, m, n-m+1\right), \beta^{-1}(1-\frac{\alpha}{2}, m+1, n-m)\,]$ with $\beta^{-1}(\alpha,u,v)$ being the $\alpha$-th quantile of a Beta distribution parametrised by $u$ and



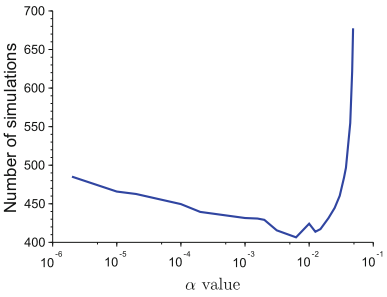

(a) Absolute error $\epsilon = 0.01$ and $\gamma = 0.25$.




(b) Absolute error $\epsilon = 0.01$ and $\gamma = 0.02$.

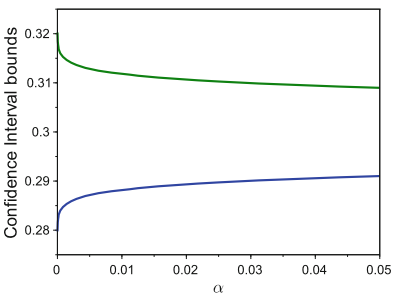



(c) Relative error $\epsilon = 0.1$ and $\gamma = 0.1$

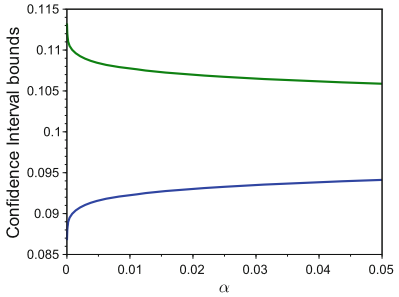



(d) Relative error $\epsilon = 0.1$ and $\gamma = 0.7$.

**Fig. 3.** Number of simulations for $\alpha$




(a) $m = 3000$ and $n = 10000$.




(b) $m = 1000$ and $n = 10000$.

**Fig. 4.** Lower and upper bounds of (Wald) $(1 - \alpha)$-confidence intervals given different number of successes $m$ and a fixed number of samples $n$.

$v$. Unfortunately, to quote [3], "the Clopper-Pearson interval $I$ is wastefully conservative and is not a good choice for practical use, unless strict adherence to the prescription $C(\gamma, I) \geq 1 - \alpha$ is demanded". In our algorithms this prescription is necessary to rigorously apply Theorems 4 and 5.

### 5.1 Approximate Confidence Intervals

However, given a confidence interval, we evaluate a worst-case value of the Massart bound. Since our approach is likely to be conservative, it remains interesting to replace the computation of exact confidence intervals by simpler approximations.

**The Wald confidence interval** is the most standard approximate confidence interval. Denoting $\Phi(.)$ the standard normal distribution function and $z_{\delta/2} = \Phi^{-1}(1 - \delta/2)$ the $(1 - \delta/2)$th quantile of the normal distribution, the notional $(1 - \delta)$-confidence interval for $\gamma$ is given by $I = \left[\hat{\gamma}_n - z_{\delta/2}\frac{\sigma}{\sqrt{n}}, \hat{\gamma}_n + z_{\delta/2}\frac{\sigma}{\sqrt{n}}\right]$, by virtue of the central limit theorem. However, in practice, $\sigma^2$ is replaced by a sample approximation $\hat{\sigma}_n^2 = \hat{\gamma}_n(1-\hat{\gamma}_n)/n$ (and if $n$ is small, $z_{\delta/2}$ by $t_{\delta/2,n-1}$ the quantile of the Student's $t$-distribution with $n - 1$ degrees of freedom). Then, the Wald approximate $(1 - \delta)$-confidence interval $\tilde{I}_W$ is given by:

$$\tilde{I}_W = \left[\hat{\gamma}_n - z_{\delta/2}\hat{\sigma}_n, \hat{\gamma}_n + z_{\delta/2}\hat{\sigma}_n\right] \tag{11}$$

The coverage of $\gamma$ by Wald interval $\tilde{I}_W$, may be significantly below the (desired) notional coverage: $C(\gamma, \tilde{I}_W) < C(\gamma, I) = 1 - \delta$. More details about this topic are available in [3].

**The Agresti-Coull confidence Interval** consists of replacing the number of samples $n$ by $n + z_\delta^2$ and the number of successes $m$ by $m + z_\delta^2/2$ in the Wald confidence interval (11):

$$\tilde{I}_{AC} = \left[\frac{m + z_\delta^2/2}{n + z_\delta^2} \pm z_{\delta/2}\sqrt{\frac{1}{n + z_\delta^2}\frac{m + z_\delta^2/2}{n + z_\delta^2}\left(1 - \frac{m + z_\delta^2/2}{n + z_\delta^2}\right)}\right] \tag{12}$$

This approximate confidence interval is recommended in several textbooks [3,20] to overcome the flaws of the Wald interval. Its coverage remains excellent, even close to probabilities 0 and 1 and may represent a good compromise between exactness and conservativeness (see [3] for more details).

### 5.2 Absolute Error Scheme Results

We repeated each set of experiments 200 times with the three different confidence intervals for several values of $\gamma$, $\epsilon$ and $\delta$. We set $\alpha = \delta/50 = 10^{-3}$ in these experiments. We estimated the empirical coverage by the number of times Specification (3) is fulfilled divided by 200 and computed the average, the standard

deviation and the extrema values of the estimations $\hat{\gamma}$ and the sample size. For the sake of clarity, as our results are consistent for all $\epsilon$, $\delta$ and are symmetric with respect to $\gamma = 1/2$, we summarize the most relevant results for $\epsilon = 0.01$, $\delta = 0.05$ and $0 < \gamma \leq 1/2$ in Table 1.

**Table 1.** Results of the absolute error scheme with $\epsilon = 0.01$ and $\delta = 0.05$

| $\gamma$ | 0.005 | 0.01 | 0.02 | 0.05 | 0.1 | 0.3 | 0.5 |
|---|---|---|---|---|---|---|---|
| Coverage (Wald), $\alpha = \delta/50$ | 1 | **0.925** | 0.97 | 0.995 | 0.98 | 0.995 | 0.99 |
| $\hat{\gamma}$ min (Wald) | 0 | 0 | 0 | 0.039 | 0.089 | 0.29 | 0.49 |
| $\hat{\gamma}$ max (Wald) | 0.011 | 0.022 | 0.028 | 0.057 | 0.113 | 0.314 | 0.512 |
| $\bar{N}$ (Wald) | 480 | 835 | 1629 | 3765 | 6893 | 15687 | 18445 |
| Coverage (AC), $\alpha = \delta/50$ | 1 | 0.995 | 0.995 | 1 | 0.99 | 0.995 | 0.98 |
| $\hat{\gamma}$ min (AC) | 0 | 0 | 0.01 | 0.041 | 0.088 | 0.29 | 0.487 |
| $\hat{\gamma}$ max (AC) | 0.011 | 0.018 | 0.029 | 0.058 | 0.111 | 0.31 | 0.511 |
| $\bar{N}$ (AC) | 710 | 1047 | 1753 | 3782 | 6874 | 15692 | 18445 |
| Coverage (CP), $\alpha = \delta/50$ | 1 | 1 | 0.995 | 0.995 | 0.99 | 0.99 | 1 |
| $\hat{\gamma}$ min (CP) | 0 | 0.003 | 0.01 | 0.04 | 0.089 | 0.29 | 0.488 |
| $\hat{\gamma}$ max (CP) | 0.009 | 0.015 | 0.028 | 0.058 | 0.114 | 0.311 | 0.508 |
| $\bar{N}$ (CP) | 971 | 1318 | 2031 | 4095 | 7192 | 15826 | 18445 |

Replacing Clopper-Pearson intervals by Agresti-Coull intervals (respectively denoted CP and AC in Tables 1 and 2) has no negative impact on the coverage of the experiment, while the ratio of the average sample sizes obtained with the Agresti-Coull and Clopper-Pearson intervals decreases from 1 to 0.73. This illustrates the wasteful amount of samples using the Clopper-Pearson method, especially for the small probabilities. Regarding the Wald confidence interval, the results are in general even better. However, we remark that for one set of experiments ($\gamma = 0.01, \epsilon = 0.01, \delta = 0.05, \alpha = 10^{-3}$), the empirical coverage were below the theoretical level $(1 - \delta)$ (in bold red in Table 1). This illustrates one of the issues encountered when using the Wald interval: the interval is too narrow. Then, the Massart function is evaluated on a too optimistic extremal point of the Wald interval that prematurely causes the termination of Algorithm 1. In order to optimise the performance of our algorithm, we thus recommend the use of the Agresti-Coull confidence interval.

### 5.3 Relative Error Scheme Results

As for the absolute error algorithm, we repeated our relative error scheme 200 times per set of experiments with Wald and Clopper-Pearson intervals. We have not reported the empirical coverage since the empirical coverages were all equal to 1. This suggests that our relative error scheme remains conservative, even

**Table 2.** Sample size average of the relative error schemes, given $\epsilon$ and $\delta$.

| $\gamma$ | 0.9 | 0.7 | 0.5 | 0.3 | 0.1 | 0.05 | 0.01 | 0.001 |
|---|---|---|---|---|---|---|---|---|
| $\bar{N}$ Wald, $(\epsilon, \delta, \alpha) = (0.05, 0.01, 0.001)$ | 573 | 2016 | 4617 | 10508 | 39927 | 83858 | 437847 | 4400530 |
| $\bar{N}$ CP, $(\epsilon, \delta, \alpha) = (0.05, 0.01, 0.001)$ | 648 | 2119 | 4701 | 10686 | 40303 | 84880 | 438929 | 4438120 |
| $\bar{N}$ Wald, $(\epsilon, \delta, \alpha) = (0.1, 0.01, 0.001)$ | 148 | 548 | 1220 | 2734 | 10204 | 21502 | 111522 | 1121966 |
| $\bar{N}$ CP, $(\epsilon, \delta, \alpha) = (0.1, 0.01, 0.001)$ | 204 | 583 | 1273 | 2822 | 10484 | 21930 | 112880 | 1135687 |
| $\bar{N}$ Wald, $(\epsilon, \delta, \alpha) = (0.1, 0.05, 0.001)$ | 94 | 361 | 828 | 1838 | 6922 | 14642 | 75644 | 761563 |
| $\bar{N}$ CP, $(\epsilon, \delta, \alpha) = (0.1, 0.05, 0.001)$ | 156 | 431 | 905 | 1970 | 7333 | 15310 | 67511 | 789934 |
| $\bar{N}$ Wald, $(\epsilon, \delta, \alpha) = (0.05, 0.05, 0.001)$ | 374 | 1366 | 3132 | 7162 | 27200 | 57368 | 298397 | 3004281 |
| $\bar{N}$ CP, $(\epsilon, \delta, \alpha) = (0.05, 0.05, 0.001)$ | 471 | 1489 | 3296 | 7422 | 27951 | 58724 | 301258 | 3043438 |

if we replace the exact confidence interval by an approximation. The sample sizes are always lower with the Wald intervals. However, they tend to become similar when $\gamma$ tends to zero since the lower bound of the respective intervals are alike. We have not performed our relative error scheme with the Agresti-Coull confidence intervals since the Agresti-Coull interval contains $\tilde{I}_W(\gamma)$ and is less conservative than $I_{CP}(\gamma)$. The results would have thus been similar. Given these results, we also recommend the use of an approximate confidence interval, the Agresti-Coull confidence interval being a good compromise between rigorousness and performance.

Last but not least, it is worth recalling that the coverage of the Agresti-Coull confidence interval remains conservative for probability values lower than 0.1 or greater than 0.9. In between, it is possible to find couples $(n, \gamma)$ for which the coverage of the interval is below the desired $1 - \alpha$ level. But, as far we know, this remains rare and the distance between the coverage and $1 - \alpha$ never exceeds 1% in the literature (e.g. [1,3]).

## 6 Distributing the Algorithms

The standard absolute error Monte Carlo scheme can be easily distributed. Indeed, once the sample size has been calculated with the Okamoto bound, the simulations are executed independently of each other. In what follows, we call 'server' the root node of a network of computational devices and 'client' the leaf nodes. In a multi-thread system, the clients correspond to independent computational threads on a machine. In a multi-client network, the server globally manages the estimation and the clients perform the simulations. Each client executes a number of traces equal to the Okamoto bound divided by the number of threads used by the server (assuming for the sake of simplicity that the remainder is equal to zero). Once the client has finished its simulation task, it communicates the number of successes to the server. The server centralises the information from all the clients and the estimator is computed at the level of the server.

But for sequential algorithms, the sample size is a priori unknown and the estimator should be updated on-the-fly until Specification (3) or (4) holds. In

**Algorithm 3.** Distributed Absolute Error Sequential Algorithm

**Data**:
$\epsilon$, $\delta$, $\beta$ : the original parameters
$M = \lceil \frac{1}{2\epsilon^2} \log \frac{2}{\delta} \rceil$: the Okamoto bound
$K = 0$
$m = 0$: the number of successes
$n_K = M$
$I_K = [a_K, b_K] = [0, 1]$: the initial interval to which $\gamma$ is known to belong

$j = 0$
**while** $K < n_K$ **do**
  Server sends $m$, $K$ and $M^{(j)} = \frac{M-K}{r^{(j)}}$ to $r^{(j)}$ clients.
  Each client $i$, $1 \leq i \leq r^{(j)}$, samples at most $\frac{M-K}{r^{(j)}}$ traces.
  $m_i = 0$
  $k_i = 0$
  **while** $K + k_i < \frac{n_{K+k_i}}{r^{(j)}}$ *and* $k_i < \frac{M-K}{r^{(j)}}$ **do**
    $k_i \leftarrow k_i + 1$
    GENERATE $\omega^{(k_i)}$
    $z(\omega^{(k_i)}) = \mathbb{1}(\omega^{(k_i)} \models \phi)$
    $m_i \leftarrow m_i + z(\omega^{(k_i)})$
    $I_{K+k_i} \leftarrow$ CONFIDENCE INTERVAL$(m + m_i, K + k_i, \beta)$
    **if** $1/2 \in I_{K+k_i}$ **then**
      $n_{K+k_i} = M$
    **else if** $b_{K+k_i} < 1/2$ **then**
      $n_{K+k_i} = \lceil \frac{2}{h_a(b_{K+k_i}, \epsilon)\epsilon^2} \log \frac{2}{\delta - \beta} \rceil$
    **else**
      $n_{K+k_i} = \lceil \frac{2}{h_a(a_{K+k_i}, \epsilon)\epsilon^2} \log \frac{2}{\delta - \beta} \rceil$
    $n_{K+k_i} \leftarrow \min(n_{K+k_i}, M)$
  $m = m + \sum_{i=1}^{r^{(j)}} m_i$
  $K = K + \sum_{i=1}^{r^{(j)}} k_i$
  $n_K \leftarrow$ UPDATE$(m, K, \epsilon, \delta, \beta)$
  $j = j + 1$

**Output**: $\hat{\gamma}_K = m/K$

what follows, we propose a distributed algorithm for the absolute error that reduces the amount of central processing and reduces the amount of time due to communication between the clients and the server.

### 6.1 A Distributed Version of the Absolute Error Scheme

The following idea can be easily adapted to the relative error scheme. For the sake of readability, we only explain how to distribute our absolute error sampling scheme. Initially the server computes the Okamato bound, divides the simulation work between $r^{(0)}$ clients and sends to the clients the parameters of the algorithm

$\epsilon, \delta, \beta$, the current number of successes $m = 0$ and samples $K = 0$ and the maximal number of samples $M^{(0)} = M/r^{(0)}$ that each client, indexed by $i$, may perform.

Each client executes simulations as in Algorithm 1 but stops as soon as its sample size $k_i$ is greater than $n_{k_i}/r^{(0)}$. Once all the clients communicated their local number of successes and samples, the server updates $m = \sum_{i=1}^{r^{(0)}} m_i$, $K = \sum_{i=1}^{r^{(0)}} k_i$ and computes a global $n_K$ to check whether $K < n_K$ or not. If $K \geq n_K$ holds, then the server outputs $\hat{\gamma}_K = m/K$ and Specification (3) is fulfilled. The simulations are all independent and the clients do not communicate with the other clients their local results. Then, since the server waits for all the clients' local results before updating $n_K$, the correctness of the algorithm is preserved. The idea behind stopping client $i$ once $k_i > n_{k_i}/r^{(0)}$ is the following: if all the clients (roughly) communicate the same number of successes and samples, $m \approx r^{(0)} m_i$, $K \approx r^{(0)} k_i$ and consequently $K \geq n_K$. However, if the local results are very different, it could be that $K < n_K$. Then, the server divides the maximal remaining samples $M - K$ between all the available clients $r^{(1)}$ and sends them the updated values of $m$ and $K$. The procedure is repeated until $K \geq n_K$. Note that the number of available clients $r^{(j)}$ may change from one step to another.

**Gain in Time.** This distributed version of the algorithm potentially involves several rounds of communication between a server and the clients. However, the number of rounds $j$ likely remains small. For the sake of simplicity, we assume that the number of clients $r$ is constant. Let $c$ be the cost in time of the communication between a server and a client and $d$ be the average cost of one execution trace. We can reasonably assume that the cost of the intermediate calculations is negligible in comparison of $c$ and $d$ and that $d$ is significantly greater than $c$. Then, the amount of time taken by the whole experiment is roughly $jcr + dK/r$ instead of $dK$ where the overhead cost $jcr$ due to communication is largely compensated by the gain due to the division of $dK$ by $r$.

## 7 Conclusion

In this work we discussed several optimisations and features for the sequential Massart algorithm introduced in [15]. In particular, it appears that in practice, using approximate instead of exact confidence intervals in the algorithm facilitates at least faster preliminary analysis. Moreover, the Agresti-Coull confidence interval reduces the sample size without significant impact on the coverage. Also, even if setting up optimally the coverage parameter a priori is not possible, it seems likely to set it up closer to zero than $\delta$. Last but not least, we showed that the schemes can be efficiently distributed on high performance parallel computational architectures.

**Acknowledgment.** This work was supported in part by the National Research Foundation (NRF), Prime Minister's Office, Singapore, under its National Cybersecurity

R&D Programme (Award No. NRF2014NCR-NCR001-040) and administered by the National Cybersecurity R&D Directorate.

## References


1. Agresti, A., Caffo, B.: Simple and effective confidence intervals for proportions and differences of proportions result from adding two successes and two failures. Am. Stat. **54**(4), 280–288 (2000)
2. Ballarini, P., Barbot, B., Duflot, M., Haddad, S., Pekergin, N.: HASL: a new approach for performance evaluation and model checking from concepts to experimentation. Perform. Eval. **90**, 53–77 (2015)
3. Brown, L., Cai, T., DasGupta, A.: Interval estimation for a binomial proportion. Stat. Sci. **16**(2), 101–133 (2001)
4. Chen, Y., Poskitt, C.M., Sun, J.: Learning from mutants: using code mutation to learn and monitor invariants of a cyber-physical system. In: SP, pp. 648–660 (2018)
5. Chernoff, H.: 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. Clopper, C.J., Pearson, E.S.: The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika **26**, 404–413 (1934)
7. Dagum, P., Karp, R.M., Luby, M., Ross, S.M.: An optimal algorithm for Monte Carlo estimation. SIAM J. Comput. **29**(5), 1484–1496 (2000)
8. David, A., Larsen, K.G., Legay, A., Mikucionis, M., Poulsen, D.B.: Uppaal SMC tutorial. STTT **17**(4), 397–415 (2015)
9. Grosu, R., Peled, D., Ramakrishnan, C.R., Smolka, S.A., Stoller, S.D., Yang, J.: Using statistical model checking for measuring systems. In: Margaria, T., Steffen, B. (eds.) ISoLA 2014. LNCS, vol. 8803, pp. 223–238. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45231-8_16
10. Hérault, T., Lassaigne, R., Magniette, F., Peyronnet, S.: Approximate probabilistic model checking. In: Steffen, B., Levi, G. (eds.) VMCAI 2004. LNCS, vol. 2937, pp. 73–84. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24622-0_8
11. Jegourel, C., Legay, A., Sedwards, S.: A platform for high performance statistical model checking – PLASMA. In: Flanagan, C., König, B. (eds.) TACAS 2012. LNCS, vol. 7214, pp. 498–503. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-28756-5_37
12. Jegourel, C., Legay, A., Sedwards, S.: Importance splitting for statistical model checking rare properties. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 576–591. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39799-8_38
13. Jegourel, C., Legay, A., Sedwards, S.: Command-based importance sampling for statistical model checking. Theor. Comput. Sci. **649**, 1–24 (2016)
14. Jegourel, C., Sun, J., Dong, J.S.: Sequential schemes for frequentist estimation of properties in statistical model checking (Journal version). Currently submitted
15. Jegourel, C., Sun, J., Dong, J.S.: Sequential schemes for frequentist estimation of properties in statistical model checking. In: Bertrand, N., Bortolussi, L. (eds.) QEST 2017. LNCS, vol. 10503, pp. 333–350. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-66335-7_23
16. Kwiatkowska, M.Z., Norman, G., Parker, D.: PRISM 2.0: a tool for probabilistic model checking. In: QEST, pp. 322–323. IEEE (2004)
17. Massart, P.: The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality. Ann. Probab. **18**, 1269–1283 (1990)

18. Metropolis, N., Ulam, S.: The Monte Carlo method. J. Am. Stat. Assoc. **44**(247), 335–341 (1949)
19. Okamoto, M.: Some inequalities relating to the partial sum of binomial probabilities. Ann. Inst. Stat. Math. **10**, 29–35 (1958)
20. Samuels, M.L., Witmer, J.W.: Statistics for the Life Sciences, 2nd edn. Prentice Hall, Englewood Cliffs (1999)
21. Wald, A.: Sequential tests of statistical hypotheses. Ann. Math. Stat. **16**(2), 117–186 (1945)
22. Younes, H.: Verification and planning for stochastic processes with asynchronous events. Ph.D. thesis, Carnegie Mellon University (2004)
23. Zuliani, P., Platzer, A., Clarke, E.M.: Bayesian statistical model checking with application to stateflow/simulink verification. FMSD **43**(2), 338–367 (2013)