Review on Monte Carlo Simulation Stopping Rules: How Many Samples Are Really Enough?

Martin Bicher[1*], Matthias Wastian[2], Dominik Brunmeir[2], Niki Popper[1,2,3]

[1] Institute for Information Software Engineering, TU Wien, Favoritenstraße 9–11, 1040 Vienna, Austria; *martin.bicher@tuwien.ac.at*

[2]dwh GmbH, Neustiftgasse 57–59,1070 Vienna, Austria;

[3]DEXHELPP, Association for Decision Support for Health Policy and Planning, Neustiftgasse 57–59, 1070 Vienna, Austria;

SNE 32(1), 2022, 1-8, DOI: 10.11128/sne.32.on.10591
Received: 2020-11-10 (selected EUROSIM 2019 Postconference Publication); Revised: 2021-05-10; Accepted: 2021-06-15
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. Due to extensive usage of stochastic simulation models correct execution of Monte Carlo simulation has become more and more important. Hereby the unknown real mean of the simulation result is estimated by the sample mean of a large number of simulation evaluations. Unfortunately, this procedure is often done carelessly. Modellers commonly use replication counts without scientific justification and sometimes underestimate the consequences of a bad or even wrong choice: if it is chosen too small, the sample mean is not a representative approximation for the regarded mean, and not only the simulation output, but also any kind of simulation analysis will not be representative at all. If the number is chosen too high, the Monte Carlo experiment will consume unnecessary computation time, which could, exemplarily, be invested into deeper model analysis instead. In this work, we present four methods that allow calculating an optimal replication number for Monte Carlo simulation and getting an image about the error between the estimated and the real mean value. The methods are furthermore evaluated on a simple case study, a stochastic cellular automaton model for simulation of an infectious disease.

Introduction

During the last decades stochastic microscopic simulation methods like agent-based modelling or discrete event simulation established as standard tools for decision support. Even though the actual distribution of the simulation results of such models carries extremely valuable information, usually the mean value of the simulation's state variable is regarded as *the* simulation output and approximated using the sample mean of Monte Carlo (MC) experiments with the simulation.

Unfortunately, this extremely important procedure is often regarded as trivial and therefore sidelined. Modellers often feel so annoyed by waiting for dozens of simulation runs to be finished and calculating the sample mean that they do not want to spend any effort on carefully choosing the only parameter of this procedure, namely the number of repetitions. Sentences like "*We repeated the simulation M times to flatten stochastic effects*" are often found in literature without any scientific justification, why M should be a meaningful number of repetitions. Some examples:

- [1] *Ten replications ... were performed ... Increasing the number of replications shows no significant effect on this error.*
- [2] *The averaging ... was carried out on* 500 *MC runs.*
- [3] *To reflect randomness 10 replicate runs were undertaken for each scenario, where the ... Random Seed parameter is changed in each replicate.*

It seems as if modellers are not really conscious about the consequences of a bad or even wrong choice: if M is chosen too small, the sample mean is not a representative approximation for the regarded mean, and not only the simulation output, but also any kind of simulation analysis (parameter variations, scenario tests, etc.) is not representative at all.

If the number is chosen too high, the Monte Carlo experiment will consume unnecessary computation time, which could, exemplary, be invested into deeper model analysis instead.

Summarising, modellers use highly sophisticated methods for identification, optimisation or calibration of model parameters, but still use a more or less arbitrary repetition count, probably some suitable power of ten, to perform the Monte Carlo experiment with the simulation. To overcome this imbalance, this work centers around the question: *How many times do I really need to repeat the simulation to make sure, the mean-value estimator lies within a certain tolerance from the real mean?*

Interpreting the single simulation-run results in a Monte Carlo simulation as a sequence of i.i.d. random numbers, the problem turns into a solely theoretic one. Hereby, the research question can be viewed completely detached from the process of Monte Carlo simulation and, on the first glance, appears like a problem that mathematicians should have solved long time ago: How well does the sample mean approximate the real one? Indeed a lot of researchers have successfully worked on it and found a lot of different formulas that describe the difference between those two quantities (see [4, 5, 6, 7, 8, 9]). What seems positive at first, contains one large disadvantage: *there is no unique answer to this problem*!

Consequently, modellers who start taking care about the correct number of replications for their simulation, not only have to search for a possibly very complicated formula, they also have to evaluate which of the derived concepts is the "best" or most "correct". In the course of this work, we take a closer look at the two most fundamental concepts to answer the research question and evaluated their value when applied to a Monte Carlo simulation case study.

1 Methods

Before going into details of the two mentioned methods, we clarify some notation which we will use henceforth. First of all, we will consider the output of the simulation as a random variable $X \in \mathbb{R}$ and define $X_i, i \in \mathbb{N}$ as i.i.d. copies of it. Clearly, this restricts the simulation output to be a scalar number, but most of the ideas can be extended to time-series or multidimensional output as well. As X is defined as the output of a simulation it is fair to assume that $\mu := \mathbb{E}(X)$ and $\sigma^2 := \mathbb{V}(X)$ are finite real numbers.

All methods in this work consider the empiric mean

$$\overline{X}_M := \frac{1}{M}\sum_{i=1}^{M}. \tag{1}$$

and the corresponding error $|\overline{X}_M - \mu|$, which itself is a random variable. Therefore, we cannot ask for a useful upper and lower bound for it, but seek for a confidence interval with predefined width δ_{abs} and confidence level p so that

$$P\left(|\overline{X}_M - \mu| \leq \delta_{\text{abs}}\right) \geq p. \tag{2}$$

Suppose, we are able to find a formula that describes the relation between sample size M, probability p and allowed absolute error δ_{abs}, we are able stop the Monte Carlo simulation whenever the allowed error δ_{abs} and the failure probability $(1-p)$ are sufficiently small. We combined these thoughts in a so called stopping rule:

Definition 1 (Stopping Rule)**.** *A real-valued function*

$$f : \mathbb{R}^+ \times (0,1) \times \mathbb{N} \to \mathbb{R} : (\delta_{abs}, p, M) \mapsto f(\delta_{abs}, p, M, \cdot)$$

is called stopping rule for the Monte Carlo simulation if

$$f(\delta_{abs}, p, M, \cdot) \geq 0 \Rightarrow P(\mu \in [\overline{X}_M - \delta_{abs}, \overline{X}_M + \delta_{abs}]) \geq p. \tag{3}$$

We will call the smallest positive integer M for which $f(\delta_{abs}, p, M, \cdot) \geq 0$ the stopping index of the stopping rule and use the label

$$M_{stop} = M_{stop}(\cdot). \tag{4}$$

The $\cdot$ notation indicates, that it is possible, that stopping index and stopping rule may depend on additional parameters, like (sample-) moments of the distribution (see later).

As mentioned, there are several approaches how such a stopping rule can be derived. We will take a look at the two fundamental ones.

1.1 Chebyshev Inequality Stopping Rule

The Chebyshev inequality or Bienaymé–Chebyshev inequality [10] was applied for determination of replication numbers in, exemplary, [11]. This inequality has become a standard tool in stochastics and gives a connection between a random number's variance and its expected value:

$$P(|X - \mathbb{E}(X)| \geq k) \leq \frac{\mathbb{V}(X)}{k^2}. \tag{5}$$

This inequality is valid for any random number X with finite expected value and variance and is independent of the distribution. Therefore it can be applied for the sample mean of independent simulation experiments. As $\mathbb{V}(\overline{X}) = \frac{\mathbb{V}(X)}{M} = \frac{\sigma^2}{M}$ and $\mathbb{E}(\overline{X}) = \mathbb{E}(X) = \mu$, we get

$$P(|\overline{X}_M - \mu| \geq k) \leq \frac{\sigma^2}{Mk^2} \Rightarrow P(|\overline{X}_M - \mu| < k) \geq 1 - \frac{\sigma^2}{Mk^2}. \tag{6}$$

With $k = \delta_{\text{abs}}$ and $1 - \frac{\sigma^2}{Mk^2} = p$ we get, that

Corollary 1 (Chebyshev Stopping Rule)**.**

$$f(\delta_{abs}, p, M, \sigma^2) = (1-p) - \frac{\sigma^2}{M\delta_{abs}^2}, \tag{7}$$

is a stopping rule for the Monte Carlo simulation, if X has a bounded first and second moment.

Unfortunately, the Chebyshev inequality is only sharp in very rare cases. Hence, we expect that the rule usually overestimates the iteration count.

1.2 Gauss-Distribution Stopping Rule

The second and even more frequently used stopping rule (e.g. [12] p. 119) is based on the Central Limit Theorem (CLT)

$$\frac{\overline{X}_M - \mu}{\sqrt{M}\sigma} \xrightarrow[M\to\infty]{P} Y \sim \mathcal{N}(1,0). \tag{8}$$

Consequently, in case M is large enough, $\overline{X}_M$ can be imagined as normally- (Gaussian-) distributed with parameters μ and $\sigma/\sqrt{M}$. Hence, we may use the percentiles of the standard normal distribution to estimate the probability. Let

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-s^2/2} ds \tag{9}$$

stand for the probability function of the normal distribution, then

$$P\left(\frac{\overline{X}_M - \mathbb{E}(\overline{X}_M)}{\sqrt{\mathbb{V}(\overline{X}_M)}} \leq k\right) = P\left(\frac{\sqrt{M}(\overline{X}_M - \mu)}{\sigma} \leq k\right) \approx \Phi(k). \tag{10}$$

and thus

$$P\left(|\overline{X}_M - \mu| \leq \frac{k\sigma}{\sqrt{M}}\right) \approx \Phi(k) - \Phi(-k) = 1 - 2\Phi(-k). \tag{11}$$

We conclude that

Corollary 2 (Gaussian Stopping Rule)**.**

$$f(\delta_{abs}, p, M, \sigma^2) = (1-p) - 2\Phi\left(-\frac{\sqrt{M}\delta_{abs}}{\sqrt{\sigma^2}}\right) \tag{12}$$

is an asymptotic stopping rule for the Monte Carlo simulation, in case X has a bounded first and second moment and M is sufficiently large.

Interestingly, the speed of convergence of the sample mean quantiles towards the quantiles of the Normal distribution depends on the skewness of the distribution of X (see Edgeworth Extension, [13] p. 538) and specifically if the distribution is symmetric or not. Consequently, the Gaussian estimator needs to be used with care if

- the number of samples M is small (e.g. if the allowed error is comparably large), and
- the distribution of X is skewed.

In these cases the Gaussian stopping rule might eventually underestimate the iteration count.

1.3 Estimation of the Variance

At this stage, neither of the two stopping rules can be applied directly as both depend on the unknown variance σ^2 of the random number, i.e. the fluctuations of the simulation output. Essential to overcome this problem is, that we do not necessarily need to know these quantities precisely, but need a feasible approximation for it. We need to make sure that $\forall p, \delta_{\text{abs}}, M : (\delta_{\text{abs}}, p, M, \sigma^2) \leq 0 \Rightarrow f(\delta_{\text{abs}}, p, M, s^2) \leq 0$ for a variance estimator s^2. In this case, the function remains a stopping rule and $M_{\text{stop}}(\sigma^2) \leq M_{\text{stop}}(s^2)$. This is achieved if either

(a) $s^2 \geq \sigma^2$ as the variance occurs in the denominator of both stopping rules, or

(b) $|s^2 - \sigma^2|$ is small enough to make sure that $M_{\text{stop}}(s_2) = M_{\text{stop}}(\sigma^2)$.

Ad (a) In case $X \in [a,b]$ for some known a,b, it is possible to state a crude upper bound for the variance. Knowing that the expected value μ minimises the mean-quadratic error $f(t) = \mathbb{E}(X-t)^2$, we get

$$\mathbb{V}(X) = \mathbb{E}(X-\mu)^2 \leq \mathbb{E}\left(X - \frac{b+a}{2}\right)^2 \leq \mathbb{E}\left(b - \frac{b+a}{2}\right)^2 = \frac{(b-a)^2}{4} =: s_c^2. \quad (13)$$

Unfortunately s_c^2 might overestimate the real variance by a large margin. Therefore the iteration count might also be overestimated. Moreover, if we cannot find an upper or lower bound for the simulation result, we cannot state a suitable upper bound for the variance as well which only leaves variant (b).

Ad (b) The most reliable estimator for the real variance is, of course, the sample variance

$$s_N^2 := \frac{1}{N-1}\sum_{i=1}^{N}\left(X_i - \overline{X}_N\right)^2, \quad (14)$$

which can be determined by N individual simulation replications. It makes sense including the evaluation of the sample variance into the Monte Carlo simulation itself, i.e. using $N = M$. Unfortunately, the sample variance gives no guarantee that the real variance is not underestimated, even if the iteration count is very large, which essentially makes every stopping rule an asymptotic one.

1.4 Application of Stopping Rules

Using crude variance bound s_c^2 mentioned above makes it possible to apply any of the two defined stopping rules (7) and (12) in advance. It is presented as pseudo-code in Algorithm 1.

Algorithm 1 Monte Carlo Simulation with A-Priori Variance Estimate

Require: $p \in [0,1], \delta_{\text{abs}} > 0, X \in [a,b]$
  $s_c^2 \leftarrow \frac{(b-a)^2}{4}$
  $M_{\text{stop}} \leftarrow \mathbf{argmin}_{M\in\mathbb{N}}\{f(\delta_{\text{abs}}, p, M, s_c^2) \geq 0\}$
  $X' \leftarrow 0$
  for $i \in [1, \ldots, M_{\text{stop}}]$ **do**
    $X_i \leftarrow$ Simulation()
    $X' \leftarrow X' + X_i$
  end for
  return X'/M_{stop}

Before stating the corresponding algorithm for applying stopping rules with sample moments, we need to mention, that both, sample mean and sample variance can be update dynamically. Clearly,

$$\overline{X}_{M+1} = \frac{M}{M+1}\overline{X}_M + \frac{1}{M+1}X_{i+1}. \quad (15)$$

Hence, $\overline{X}_{M+1}$ can be calculated from $\overline{X}_M$ with reasonable effort. Unfortunately this strategy can only be applied indirectly to the estimator of the sample variance. With $\overline{X^2}_M = \frac{1}{M}\sum_{i=1}^{M} X_i^2$,

$$\overline{s^2}_M = \frac{1}{M-1}\sum_{i=1}^{M}(X_i - \overline{X}_M)^2 = \frac{M}{M-1}\left(\overline{X^2}_M - \overline{X}_M^2\right). \quad (16)$$

As both, $\overline{X}$ and $\overline{X^2}$ can be updated on-the-fly using (15), also s^2 can be updated dynamically.

We conclude Algorithm 2 which is also proposed exemplarily in [12].

Algorithm 2 Monte Carlo Simulation with Dynamic Variance Estimate

Require: $p \in [0,1], \delta_{\text{abs}} > 0, M_0 \in \mathbb{N}$
  $\overline{X}_0 \leftarrow 0$
  $\overline{X^2}_0 \leftarrow 0$
  for $i \in \mathbb{N}$ **do**
    $X_i \leftarrow$ Simulation()
    $\overline{X}_i \leftarrow \frac{i-1}{i}\overline{X}_{i-1} + \frac{1}{i}X_i$
    $\overline{X^2}_i \leftarrow \frac{i-1}{i}\overline{X^2}_{i-1} + \frac{1}{i}X_i^2$
    if $i > M_0$ **then**
      $s_i^2 \leftarrow \frac{i}{i-1}\left(\overline{X^2}_i - \overline{X}_i^2\right)$
      if $f(\delta_{\text{abs}}, p, M, s_M^2) \geq 0$ **then**
        return $\overline{X}_i$
      end if
    end if
  end for

The initial guard $M_0 \geq 1$ prevents that the algorithm terminates prematurely. Quick analysis makes clear that the loop will run through three phases.

1. For $i < M_0$, the loop runs through a warm-up phase and is not allowed to stop. It prevents premature termination of the algorithm by a gross underestimation of the sample variance, caused by a few simulation results that lie very close together.
2. For $M_0 \leq i \ll M_{\text{stop}}(\sigma^2)$ the sample variance s_i has stopped fluctuating and represents a feasible approximation for σ^2. Therefore, $i \ll M_{\text{stop}}(s_i^2)$ holds as well.
3. For i close to $M_{\text{stop}}(\sigma^2)$ the sample variance approximates the real variance really well. Therefore, $M_{\text{stop}}(\sigma^2) \neq M_{\text{stop}}(s_i^2)$ only in very rare cases.

Consequently, the choice of M_0 is not too critical and its main purpose is preventing "accidents" like

$$X_1 \approx X_2 \Rightarrow \sigma^2 \gg s_2^2 \approx 0 \Rightarrow f(p, \delta_{\text{abs}}, 2, s_2^2) \geq 0 \Rightarrow M_{\text{stop}}(s_2) = 2. \quad (17)$$

Ross [12] proposes $M_0 \geq 100$.

Algorithm 2 is sometimes found in a slightly different version wherein the sample moments are not updated continuously, but in batches of predefined size (see [14]). What seems like an insignificant mathematical detail at first, is relevant for parallel computing.

2 Case Study

In order to test both algorithms and both stopping rules, we decided to use a simple, academic, yet for this purpose quite representative cellular automata (CA) model for simulation of a susceptible-infectious-recovered (SIR) epidemic. It brings the classic SIR differential equation model by Kermack and McKendrick [15] into a spatial context and can be thought of depicting a bacterial infection that spreads on a rectangular Petri-dish. It unites ideas of typical agent-based SIR models like [16] and models of bacteria growth on a homogeneous surface like [17]

- The cell space of the CA is chosen as a rectangular grid with $m \times n$ cells.
- The state mapping of the CA maps each cell onto any of three states: $0 \cong$ susceptible, $1 \cong$ infectious, $2 \cong$ recovered. For the sake of readability we say, a cell is in e.g. the infectious-state, if the state mapping currently maps it onto 1.
- The simulation starts with one randomly chosen cell in the infectious state, while all other $m \cdot n - 1$ cells are susceptible.
- The CA uses the Von Neumann neighbourhood, i.e. a cell's neighbourhood consists of the four cells directly above, below, left and right of it. The neighbourhood is restricted at the borders.
- The CA is updated synchronously using the following two stochastic rules:
  - A cell in the susceptible state becomes infected with probability α, if it has at least one infectious neighbour.
  - A cell in the infectious state recovers with probability β.
- The time-update terminates as soon as the number of infectious cells in the CA reaches zero. We regard the *maximum number of infectious cells* observed during the iteration as the outcome of the model.

Figure 1 shows snapshots of the CA at three different times during the execution. The infection spreads radially from the point of initial infection and leaves interesting patterns. Figure 2 shows the number of infected cells as a function of time for twenty different simulation runs. Easily seen, the outcome X of the simulation run is highly irregular and has an unusual distribution. In particular, about every twelfth simulation run the disease did not break out at all, leading to $X = 1$.

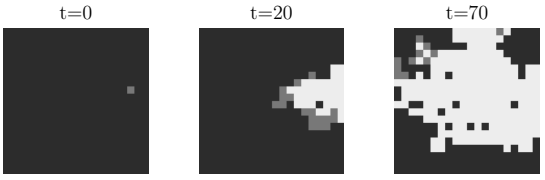

Figure 1: Three snapshots of the CA for different times during execution with $\alpha = \beta = 0.2$ and 20×20 cells. Infectious cells are marked red, susceptible blue and recovered ones yellow.

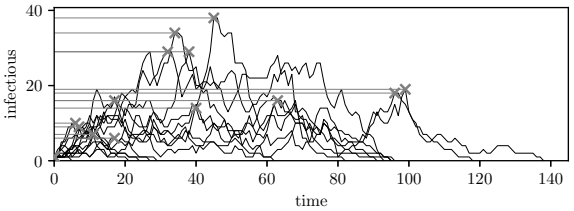

Figure 2: Results of twenty simulation runs with $\alpha = \beta = 0.2$ and 20×20 cells. The outcome value of every run, i.e. point with the highest value of infectious cells, is marked.

We chose this model for our test scenario as it is both simple and to some extent realistic. Hereby we do not mean, that the stated model is a realistic model for disease spread, but that the distribution of the model's outcome could be something we would also receive from a fully validated model. This feature distinguishes this study from tests found in standard literature about stopping of Monte Carlo experiments.

They use either classic, theoretical distributions (e.g. Exponential-, Pareto-, Uniform, Normal-inv. Gaussian-distribution are compared in [6]) or test the methods in the context of Monte Carlos integration (e.g. [8]).

3 Testing of Algorithms and Stopping Rules

3.1 Test Definition

As test scenario, $\alpha = \beta = 0.2$ as well as $N = M = 20$ were fixed. Both Monte Carlo algorithms and both stopping rules were applied in a Monte Carlos setting themselves with 10000 replications each.

In order to evaluate the failures of the algorithms, knowledge about the real mean value would be necessary. As the distribution of the simulation result cannot be determined analytically, we executed the simulation one million times and used the results of this procedure for bootstrapping. As this ridiculously large number is definitely large enough to approximate the distribution of X accurately, we replace the original simulation by drawing from a sampled list. This procedure has two key advantages:

- In contrast to the actual simulation, we know the moments of the distribution. They are precisely the empiric mean and variance of the huge data sample.
- Randomly drawing from the list executes much faster than evaluating the simulation. As we need to run each Monte Carlo test 10000 times, this is an essential feature.

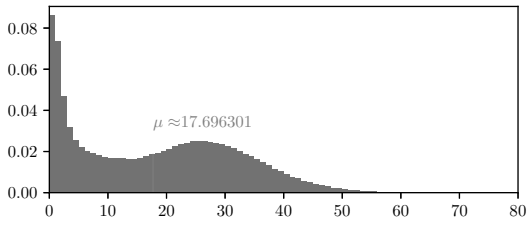

Figure 3: Histogram of the maximum number of infectious cells of one million simulation runs. The mean value is marked red.

The resulting distribution is seen in Figure 3. It has

$$\mu = \overline{X}_{10^6} = 17.696301,$$
$$\sigma^2 = s^2_{10^6} = 186.49197791739908. \quad (18)$$

To use Algorithm 1 and evaluate the stopping index M_{stop} in advance it is necessary to calculate an upper bound of the variance. As described this is only possible, if lower and upper bounds for the random number are available. Clearly, 1 poses for a lower bound of X as the maximum number of infectious cells cannot be smaller than its initial value. The total number of cells $m \times n = 400$ is a very pessimistic upper bound for X. Yet, in advance, it might be the only one available. We get

$$s_c^2 = \frac{(400-1)^2}{4} = 39800.25. \quad (19)$$

3.2 Test Results

Table 1 shows the results of tests with different values for p and δ_{abs}. The third column shows the sample mean of the stopping index $\overline{M_{\text{stop}}}$ and its sample standard deviation gained from 10000 reruns of the Monte Carlo algorithm with different random number settings. Note, that M_{stop} is deterministic in Algorithm 1, while it varies in Algorithm 2 depending on the value of the sample variance. The fourth column shows, how many of the 10000 algorithm executions failed to satisfy $\left|\overline{X}_{M_{\text{stop}}} - \mu\right| \leq \delta_{\text{abs}}$. As the confidence interval is supposed to make sure that the probability for such a failure is smaller than $1-p$, we define that the algorithm/stopping-rule failed as a whole, if the number of failures exceeded $(1-p)\cot 10000$.

The results of the case study are fascinating and allow highly interesting conclusions. First of all, the huge difference between the forecasted values for M_{stop} is remarkable. We find that the value for M_{stop} predicted by Algorithm 1 with the Chebyshev stopping rule is more than 1000 times higher than the average value gained from Algorithm 2 with the Gauss rule. In general, stopping indices predicted by Algorithm 1 are more than 200 times larger than those of Algorithm 2 which is, of course, a consequence of the crude variance bound being more than 200 times larger than the real variance.

In terms of comparing the two stopping rules we find that the stopping indices of the Chebyshev stopping rule are 5 to 8 times higher than the ones predicted by the Gauss stopping rule. This results from the general weakness of the Chebyshev inequality.

testcase	algorithm	stopping rule	$\overline{M_{\text{stop}}}$ ($\pm$ std)	failures (allowed)	failed?
	Algorithm 1	Chebyshev	199002	0 (500)	No
$p = 0.95$	Algorithm 1	Gauss	38223	0 (500)	No
$\delta_{\text{abs}} = 2$	Algorithm 2	Chebyshev	931.67($\pm$31.56)	1 (500)	No
	Algorithm 2	Gauss	178.49($\pm$14.11)	548 (500)	Yes
	Algorithm 1	Chebyshev	398003	0(250)	No
$p = 0.975$	Algorithm 1	Gauss	49988	0(250)	No
$\delta_{\text{abs}} = 2$	Algorithm 2	Chebyshev	1864.37($\pm$45.57)	0(250)	No
	Algorithm 2	Gauss	233.72($\pm$16.18)	273(250)	Yes
	Algorithm 1	Chebyshev	796005	0 (500)	No
$p = 0.95$	Algorithm 1	Gauss	152892	0 (500)	No
$\delta_{\text{abs}} = 1$	Algorithm 2	Chebyshev	3729.72($\pm$64.13)	0 (500)	No
	Algorithm 2	Gauss	715.76($\pm$28.12)	510 (500)	Yes

Table 1: Test results of 10000 Monte Carlo simulations stopped with the specified algorithm and stopping rule.

Finally, Algorithm 2 applied with the Gauss stopping rule failed in all three cases. Although they almost matched with the predicted ones, the resulting failure counts were slightly too high in all three test cases indicating that $P(|\overline{X}_{M_{\text{stop}}} - \mu| \leq \delta_{\text{abs}})$ is actually smaller than p. Therefore, the confidence level of the interval is not as high as required.

In summary, having a really non-asymptotic stopping algorithm for a Monte Carlo simulation comes at a price which is by far too high. On the one hand, performing hundreds of thousands of simulation runs is simply not necessary and too costly. Hence, the proposed dynamic algorithm with the sample variance is more practicable. On the other hand, asymptotic stopping rules may lead to an unreliable confidence interval.

4 Conclusion

In this work we presented and performed basic tests for the two most popular stopping rules for Monte Carlo simulation and investigated them in the context of two stopping algorithms. Considering the results presented in the last section, all four investigated methods essentially failed either by requiring too many or too little simulation runs. While the Gaussian approach needs to be corrected to become non-asymptotic, the Chebyshev concept needs to be made sharper. We found several attempts for both ideas in literature [9, 4, 6], yet the evaluation of the corresponding stopping strategies is work-in-progress.

We demonstrated that the seemingly simple problem of stopping Monte Carlo simulation at the right time is still not solved satisfactorily. Although the methods presented in this work are surely not optimal, we would yet still recommend using one of them in favour of just "guessing" a feasible iteration count.

References

[1] Duguay C, Chetouane F. Modeling and improving emergency department systems using discrete event simulation. *Simulation*. 2007;83(4):311–320.

[2] Purtov AN, Mamonova MV, Prudnikov VV, Prudnikov PV. Monte Carlo simulation of aging phenomena in multilayer magnetic structures. In: *EPJ Web of Conferences*, vol. 185. EDP Sciences. 2018; p. 3008.

[3] Saroj S, Rajput SJ. Composite smart mesoporous silica nanoparticles as promising therapeutic and diagnostic candidates: recent trends and applications. *Journal of Drug Delivery Science and Technology*. 2018; 44:349–365.

[4] Audibert JY, Munos R, Szepesvári C. Tuning bandit algorithms in stochastic environments. In: *International conference on algorithmic learning theory*. Springer. 2007; pp. 150–165.

[5] Maurer A, Pontil M. Empirical Bernstein bounds and sample variance penalization. *arXiv preprint arXiv:09073740*. 2009;.

[6] Bayer C, Hoel H, Von Schwerin E, Tempone R. On Nonasymptotic Optimal Stopping Criteria In Monte Carlo Simulations. *Siam Journal On Scientific Computing*. 2014;36(2):17. A869–A885.

[7] Gilman MJ. A Brief Survey of Stopping Rules in Monte Carlo Simulations. In: *Proceedings of the Second Conference on Applications of Simulations*. 1968; pp. 16–20.

[8] Hickernell F, Choi SC, Jiang L, Jiménez Rugama L. Monte Carlo Simulation, Automatic Stopping Criteria For. In: *Wiley StatsRef: Statistics Reference Online*. John Wiley & Sons. 2018;.

[9] Hickernell FJ, Jiang L, Liu Y, Owen AB. Guaranteed Conservative Fixed Width Confidence Intervals via Monte Carlo Sampling. In: *Monte Carlo and Quasi-Monte Carlo Methods 2012*, edited by Dick J, Kuo FY, Peters GW, Sloan IH, pp. 105–128. Berlin, Heidelberg: Springer Berlin Heidelberg. 2013;.

[10] Bienaymé IJ. *Considérations à l'appui de la découverte de Laplace sur la loi de probabilité dans la méthode des moindres carrés*. Imprimerie de Mallet-Bachelier. 1853.

[11] Ballio F, Guadagnini A. Convergence assessment of numerical Monte Carlo simulations in groundwater hydrology. *Water resources research*. 2004;40(4).

[12] Ross S. *Simulation*. Knovel Library. Elsevier Science. 2012.

[13] Feller W. *An introduction to probability theory and its applications*, vol. 2. John Wiley & Sons. 1957.

[14] Xie W, Nelson BL, Barton RR. Statistical uncertainty analysis for stochastic simulation. In: *Departmant of Industrial Engineering and Management Sciences, Northwestern University Working paper*. IEEE. 2014;.

[15] Kermack WO, McKendrick AG. A Contribution to the Mathematical Theory of Epidemics. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*. 1927;115(772):700–721.

[16] Miksch F, Haim C, Schneckenreither G, Breitenecker F. Comparison of Differential Equations and Cellular Automata for Epidemic Simulation. In: *ERK 2013 Proceedings.*, 22. Portoroz, Slovenia. 2013; pp. 137–140.

[17] Krawczyk K, Dzwinel W, Yuen DA. Nonlinear development of bacterial colony modeled with cellular automata and agent objects. *International Journal of Modern Physics C*. 2003;14(10):1385–1404.