ACS SyntheticBiology

pubs.acs.org/synthbio Research Article

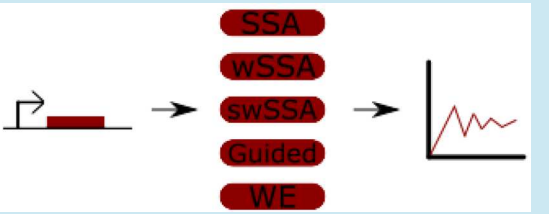
A Comparison of Weighted Stochastic Simulation Methods for the Analysis of Genetic Circuits

Mohammad Ahmadi,[#] Payton J. Thomas,[#] Lukas Buecherl, Chris Winstead, Chris J. Myers, and Hao Zheng*

Cite This: *ACS Synth. Biol.* 2023, 12, 287−304 Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: Rare events are of particular interest in synthetic biology because rare biochemical events may be catastrophic to a biological system by, for example, triggering irreversible events such as off-target drug delivery. To estimate the probability of rare events efficiently, several weighted stochastic simulation methods have been developed. Under optimal parameters and model conditions, these methods can greatly improve simulation efficiency in comparison to traditional stochastic simulation. Unfortunately, the optimal parameters and conditions cannot be deduced *a priori*. This paper presents a critical survey of weighted stochastic simulation methods. It shows that the methods considered here cannot consistently, efficiently, and exactly accomplish the task of rare event simulation without resorting to a computationally expensive calibration procedure, which undermines their overall efficiency. The results suggest that further development is needed before these methods can be deployed for general use in biological simulations.



KEYWORDS: *stochastic simulation, rare event simulation, importance sampling, weighted ensemble, genetic circuits, stochastic chemical kinetics*

INTRODUCTION

Despite occurring with low frequency, rare events can have devastating effects on biological systems. For example, rare biochemical events have been demonstrated to contribute to cancerous phenotypes by inactivating tumor-suppressing genes.[1] The issue of rare events is of particular interest in synthetic biology, where genetic circuits must reliably produce desired outputs to be of use. Unfortunately, genetic circuits are known to experience rare deviations from their expected behavior called *glitches*.[2,3]

Although glitches may occur rarely, populations of cells may each express genetic circuits continuously for a prolonged period, increasing the probability that a glitch eventually occurs. Though glitches often persist only transiently, they still pose a threat because unwanted genetic circuit outputs could produce irreversible cellular responses such as early or off-target release of a therapeutic molecule. Methods to study rare biochemical events are, therefore, a necessity in genetic circuit design.

Exact trajectories of biochemical reaction networks may be determined with *molecular dynamics*, wherein, given the initial position and momentum of each atom in the system, the complete state of the system can be determined at any time.[4] Unfortunately, such methods are computationally intractable for most systems. Instead, *stochastic chemical kinetics* (SCK), which assumes a chemical system as homogeneous and well-stirred, may be used to generate many potential trajectories for a system and approximate the probability of some event occurring.[5] This is traditionally done using the *stochastic simulation algorithm* (SSA).[6,7]

Probability estimation for rare events can be problematic for stochastic simulation because the number of trajectories that must be generated to approximate the probability of a rare event may be computationally prohibitive. To address this issue, a variety of stochastic simulation algorithms have been developed that utilize *importance sampling* (IS) techniques to more efficiently estimate the probability of rare events.[8−10] In this paper, three such algorithms are examined to determine whether they are suitable for regular use in the evaluation of robust genetic circuit designs.

The first algorithm that is examined is the *weighted stochastic simulation algorithm* (wSSA),[8] which first applied IS techniques to biochemical network simulation. The second algorithm that is examined is the *state-dependent weighted stochastic simulation algorithm* (swSSA).[9] The third algorithm that is examined is the *guided weighted stochastic simulation algorithm* (Guided wSSA).[10,11] While other IS-based algorithms exist, these three together form a representative sample of IS methods in SCK simulation.

Received: October 18, 2022
Published: December 30, 2022

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

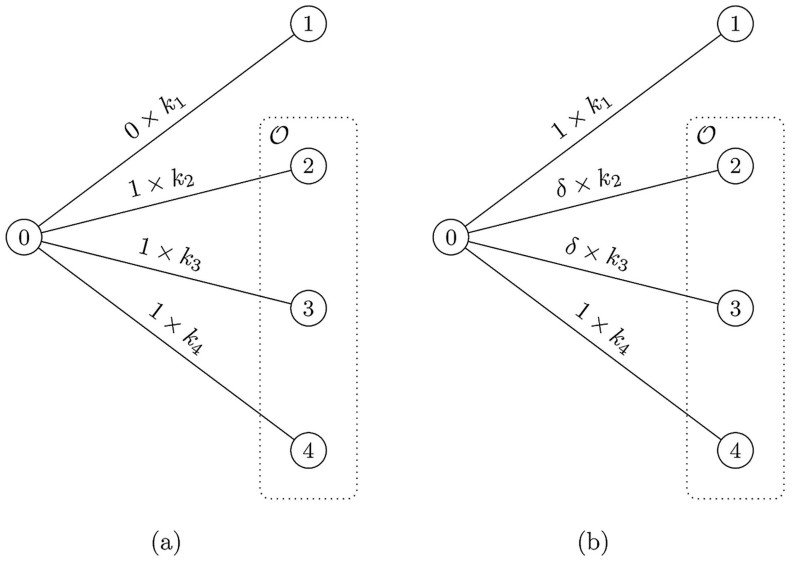


Figure 1. A simple chemical reaction network with $O = \{2, 3, 4\}$. Two wSSA biasing scenarios are shown: (a) optimal wSSA biasing and (b) poor wSSA biasing.

The performance of these algorithms is analyzed and compared on six biochemical reaction networks, varying in complexity from a simple biochemical network to a complete genetic circuit design.

In addition to IS methods, a technique known as *weighted ensemble* (WE) simulation, which is akin to stratified sampling[12,13] is examined. In contrast to wSSA, swSSA and Guided wSSA, WE does not apply biasing to any reaction rates during simulation. Individual transitions are generated using the SSA. WE tries to estimate the probability of rare events by giving more computational resources (i.e., CPU time) to trajectories that are more likely to reach the target states. It achieves this by generating parallel simulation paths, dividing the state-space into bins. Simulation paths are duplicated or terminated to maintain a uniform exploration across the bins.[12−14]

This paper makes the following claims and contributions: In the Mathematical Preliminaries section, it is shown what an ideal importance sampling scheme looks like. Then, using a simple example, it is shown how a poor biasing scheme can result in *false convergence*—an incorrect estimate with low sample variance. In this situation, a biasing scheme can *seem* near-optimal, as it appears to converge to a narrow confidence interval, when it is in fact a poor biasing scheme producing unreliable estimates. The mathematical analysis applies to the original wSSA algorithm. Other IS algorithms (swSSA and Guided wSSA) are assessed experimentally to determine if they provide more reliable estimators. The performance of the IS-based and WE methods are compared on four different examples by computing the computational gain each method achieves over SSA. The results show that false convergence happens in all four experiments with IS-based algorithms given that poor biasing parameters are selected, therefore the practitioners should use caution when applying those method, as it is possible for a biasing scheme that seems near-optimal to produce a low-variance estimate that is orders of magnitude off from the true probability. The WE method offers a more reliable alternative in the cases considered, although WE has lower performance than wSSA in terms of computational gain achieved over SSA and has higher additional runtime complexity.

MATHEMATICAL PRELIMINARIES

This paper makes two central and related claims. The first claim is that wSSA and associated Importance Sampling methods may provide unreliable results when biasing parameters are "poorly" selected. The second claim is that, for "poor" biasing cases, the sample variance is not a reliable indicator of accuracy, and the associated confidence interval may be incorrect. Prior literature advised using the wSSA confidence interval as an accuracy test,[15] but our analysis shows that the confidence interval can be unhelpful or misleading for poor biasing designs.

When wSSA is applied to complex models, it may be difficult or intractable to determine if the biasing is poor or not, so a narrow confidence interval cannot be interpreted as evidence of accuracy. To assist the experimenter, we offer some analysis to distinguish the features that may contribute to erroneous wSSA estimates. A simple example is analyzed in this section to reveal the cause of significant estimation errors. Similar patterns of error are observed experimentally for more complex models in subsequent sections of the paper.

First, we review the well-known conditions for an optimal importance sampling scheme. Then we will show how those conditions may be violated in some wSSA cases, resulting in significant errors. Consider a random experiment X with M discrete outcomes x_i, $i = 1, 2, ..., M$ (in the context of chemical reaction networks, an "outcome" refers to a distinct sequence of reaction events). Each outcome occurs with probability $p_i = \Pr(X = x_i)$. A subset of the outcomes succeeds in reaching a chosen objective, defined by an *objective set* O. The experimenter seeks to determine $P_O = \Pr(X \in O)$ through statistical sampling. The Monte Carlo estimate is determined by

$$\hat{P}_O = \frac{1}{N}\sum_{k=1}^{N} I_O(x^{(k)})$$

where N is the number of samples, I_O is a binary-valued indicator function for membership in O, and $x^{(k)}$ is the k^{th} random sample.

If P_O is very small, the experimenter may apply importance sampling to obtain the estimate using a small number of samples. Instead of sampling the x_i with their "native" probabilities p_i, the

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

experimenter imposes an alternative sampling distribution g_i. Each obtained sample is corrected by applying a sample weight $w^{(k)} = I_O(x_k)p^{(k)}/g^{(k)}$, where $p^{(k)}$ and $g^{(k)}$ are the native and biased probabilities of the obtained sample $x^{(k)}$, respectively. Then the Importance Sampling estimate is determined by

$$\tilde{P}_O = \frac{1}{\tilde{N}}\sum_{k=1}^{\tilde{N}} w^{(k)}$$

where $\tilde{N}$ is the number of samples used for the Importance Sampling experiment.

An optimal sample distribution g^* exists mathematically, although it is usually unknown to the experimenter:

$$g_i^* = \begin{cases} \dfrac{p_i}{P_O}, & x_i \in O \\ 0, & \text{otherwise} \end{cases}$$

With this optimal distribution, zero-weight samples are never observed, and every nonzero sample has the same weight, equal to the true probability sought by the experimenter:

$$w_i^* = \begin{cases} P_O, & x_i \in O \\ 0, & \text{otherwise} \end{cases}$$

Since the experimenter cannot access the optimal sampling distribution, a "good" sampling distribution is sought, which should have features similar to the optimal solution. Specifically, $g_i \propto p_i$ when $x_i \in O$, and g_i should be minimized when $x_i \notin O$. In simpler terms, a "good" sampling distribution should somehow exclude zero-weight samples (i.e., failures) without disturbing the distribution of the remaining outcomes.

A poor sampling distribution occurs when an event x_i has a relatively high native probability p_i, but is suppressed to a low sampling probability g_i. In this situation, the sample weight $w_i = p_i/g_i$ can be quite large, but occurs with frequency below $1/\tilde{N}$, so it is not likely to be observed in the experiment. The probability mass associated with x_i is effectively hidden from the experiment, resulting in both an underestimation and a deceptively low sample variance. The resulting sampling scheme is not merely suboptimal, it can produce severely distorted results without showing any evidence of inaccuracy.

To demonstrate the effects of both "good" and poor sampling distributions, we consider the almost-trivial time-abstracted reaction network shown in Figure 1. This network has four reactions with propensities a_1, a_2, a_3, and a_4 equal to 100, 10^{-2}, 10^{-2}, and 1, respectively. The probability of reaction R_i, hence reaching state i, is $p_i = a_i/a_0$, where $a_0 = \sum_{j=1}^{4} a_j = 101.02$. The probability of reaching the objective set is $P_O = p_2 + p_3 + p_4 = 0.0101$. The experimenter plans to estimate P_O using no more than $\tilde{N} = 1000$ samples.

An optimal sampling distribution is demonstrated in Figure 1a, where a_1 is fully suppressed. Then the sample probabilities are

$$g_i^* = \begin{cases} 0 & i = 1 \\ \dfrac{a_i}{\sum_{j=2}^{4} a_j} & i > 1, \end{cases}$$

and the corresponding nonzero sample weights are

$$w_i^* = \frac{p_i}{g_i} = \frac{a_i \sum_{j=2}^{4} a_j}{a_i a_0} = P_O$$

With this sampling distribution, every weighted sample is equal to the precise probability, so the sample variance is zero and a single sample is sufficient to get the correct result.

Next, to demonstrate a poor sampling distribution, shown in Figure 1b, suppose the experimenter chooses to enhance reactions R_2 and R_3 by a factor $\delta = 10^5$, so that they occur much more often than the unhelpful reaction R_1. For whatever reason (perhaps an oversight), the experimenter does not apply any biasing to R_4. Then the sampling distribution becomes

$$g_1 = \frac{a_1}{b_0} = 0.0476$$

$$g_2 = \frac{\delta a_2}{b_0} = 0.476$$

$$g_3 = \frac{\delta a_3}{b_0} = 0.476$$

$$g_4 = \frac{a_4}{b_0} = 4.76 \times 10^{-4},$$

where $b_0 = a_1 + \delta a_2 + \delta a_3 + a_4 = 2101$. Due to the reaction biasing, both R_1 and R_4 are suppressed by the factor $b_0/a_0 = 20.8$. In the case of R_1, this means the experiment produces fewer zero-weight samples, which is desirable. In the case of R_4, however, it means that a highly probable outcome in the objective set is now unlikely to be observed in the experiment. The associated probability mass resides in the sample weights:

$$w_1 = \frac{p_1}{g_1} = 0$$

$$w_2 = \frac{p_2}{g_2} = \frac{b_0}{\delta a_0} = 2.08 \times 10^{-4}$$

$$w_3 = \frac{p_3}{g_3} = \frac{b_0}{\delta a_0} = 2.08 \times 10^{-4}$$

$$w_4 = \frac{p_4}{g_4} = \frac{b_0}{a_0} = 20.8.$$

After collecting $\tilde{N} = 1000$ samples, we expect about 476 of the weighted samples to be w_2, another 476 to be w_3, and the remaining samples are expected to be zero. The most likely estimate is therefore $\tilde{P}_O \approx 1.98 \times 10^{-4}$, underestimating the true probability by a factor of 51. For this statistical experiment, the standard-deviation confidence interval would be $\pm 1.4 \times 10^{-6}$, giving the false appearance of an accurate estimation.

Although underestimation is the most likely outcome in this scenario, it is possible for wSSA to obtain other estimation results. To evaluate the range of possible experimental results, ten thousand wSSA experiments were simulated using the same biasing procedure, each with the same sample size $\tilde{N} = 1000$. The plot in Figure 2 shows a histogram of estimation results as well as the 99.5% confidence interval for each result. The confidence interval is visualized as a shaded rectangle around the estimation. In 62% of experiments, the wSSA result is an underestimate at $1.98 \times 10^{-4} \pm 5 \times 10^{-6}$. In these cases the 95% confidence interval is quite narrow, barely perceptible in Figure

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

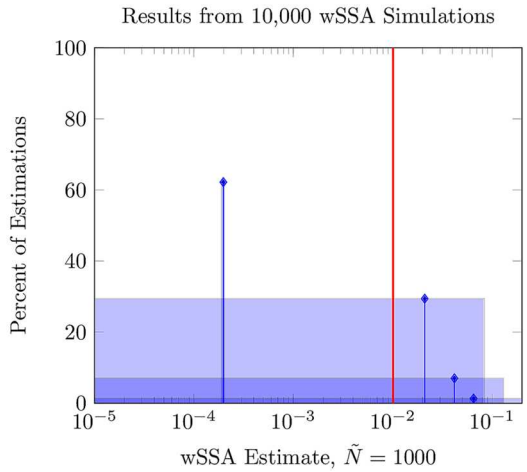


Figure 2. Results from ten thousand wSSA estimations with poor reaction biasing. The height of each data point indicates the percent of cases that obtained that estimate. The blue rectangle indicates the worst 99.5% confidence interval for that group of estimations. The exact probability is indicated by the thick vertical line.

2. In the remaining 38% of experiments, the results are overestimates with very wide confidence intervals spanning from 0 to 0.07 or higher. In all cases, the estimation is wrong by at least an order of magnitude.

It remains true that the wSSA estimation converges toward the exact P_O as $\tilde{N}$ goes to infinity, even in a "poor" biasing scenario. For a finite sample size, the estimation can be quite far from the exact value, and it may not be possible to predict the required sample size to obtain a desired accuracy. The primary hazard of wSSA is underestimation with low sample variance. We may conclude that overestimates reveal themselves via high sample variance, and when a low sample variance occurs it should be interpreted as a lower bound and not necessarily a precise estimation.

To apply this example to more complex models, we may consider each outcome in Figure 1 as a set of reaction paths rather than a single reaction sequence. In this interpretation, R_1 represents the set of paths that fail to reach the objective. R_2 and R_3 represent sets of paths reaching O that are enhanced by the experimenter, and R_4 represents a set of paths reaching O that are not biased or are inadvertently suppressed by the biasing procedure. This could correspond, for example, to a large multiplicity of low-probability paths that escape the experimenter's awareness. In that situation, even with a good-faith biasing effort the contribution of those paths is concealed from the wSSA result.

In the sequel we present several examples of wSSA and derived algorithms applied to more complicated models. In those examples we observe the phenomena analyzed here: underestimation with low sample variance, and infrequent occurrence of large-weight samples. We attempt to assess whether these problems are resolved by improved wSSA algorithms, and we assess the computational cost of those improvements in comparison to alternatives, namely SSA and Weighted Ensemble algorithms.

RESULTS AND DISCUSSION

Experiment Setup and Evaluation. The setup used to perform experiments in the following subsections and the statistics used to compare their performance are described in this section.

Running wSSA and swSSA requires the selection of a set of reactions to be biased prior to the simulation. For an event $X \xrightarrow{t \le T} \theta$ (population of species X reaching θ within T time units), this is done by analyzing the model's reaction network and identifying reactions that would help the simulation reach a state where X = θ and those reactions that would diminish the possibility of reaching the target state within the time-bound T.

To simplify things, a single biasing parameter is used for all selected reactions in wSSA and swSSA methods. For wSSA, a single value $\delta \in (0, 1)$ is selected. The propensity of reactions that are selected to be biased upward is multiplied by $\frac{1}{\delta}$ and the propensity of reactions that are selected to be biased downward is multiplied by δ. Similarly, a single value $\gamma_{\max}$ is used for the swSSA method. The propensity of reactions that are selected to be biased upward can be maximally increased by a factor of $\gamma_{\max}$ and the propensity of reactions that are selected to be biased downward can be maximally decreased by a factor of $\gamma_{\max}$. For each model, the wSSA and swSSA methods are first run for a range of δ and $\gamma_{\max}$ values, and the reported probability for each of these values is compared to the exact probability of the event. A value for the biasing parameter is then selected from the range of values producing an acceptable probability estimate (one close to the exact probability) for further experimentation. For cases where exact probability of the event is not known, a probability estimate produced by running SSA simulation is considered exact.

In various places, confidence intervals and standard errors are used to compare the performance of different methods. This requires calculating the variance for the estimator. For IS-based methods, N simulation runs are performed, resulting in N run-weight samples. The probability is reported as the mean of these N samples and standard error and confidence interval are calculated using the variance of these N samples. For the WE method, an ensemble of M simulations is run for each model. For each simulation, the sum of the weights of the trajectories reaching the target bin is collected. The probability is then reported as the mean of these M samples and the standard error and confidence interval are calculated using the variance of these M samples.

To compare the performance of these methods on each model, a computational gain achieved over SSA is reported for each method. For a probability estimate $\hat{p}$ with standard error $\frac{\sigma}{\sqrt{n}}$, Gillespie et al.[15] report that given $\hat{p} \ll 1$, the number of SSA simulations required to produce an estimate with the same level of accuracy can be estimated as $n^{\mathrm{SSA}} = \frac{\hat{p}}{(\sigma / \sqrt{n})^2}$. Running N IS-based simulations, the computational gain over SSA is calculated as $\frac{n^{\mathrm{SSA}}}{N}$.

In order to calculate the computational gain for the WE method, first the number of SSA runs that is computationally equivalent to the performed WE run is calculated. Given a WE setup with polling-time τ, each trajectory segment is simulated for τ units of time before the WE dynamics is applied. Given that the time-bound in the event of interest is T and assuming that the ensemble of WE simulations simulates N_{segment} trajectory segments in total, $N_{\mathrm{equivalent}} = N_{\mathrm{segment}} \times \frac{\tau}{T}$ gives the number of SSA runs that are computationally equivalent to running the ensemble of M weighted ensemble simulations. The computa-

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

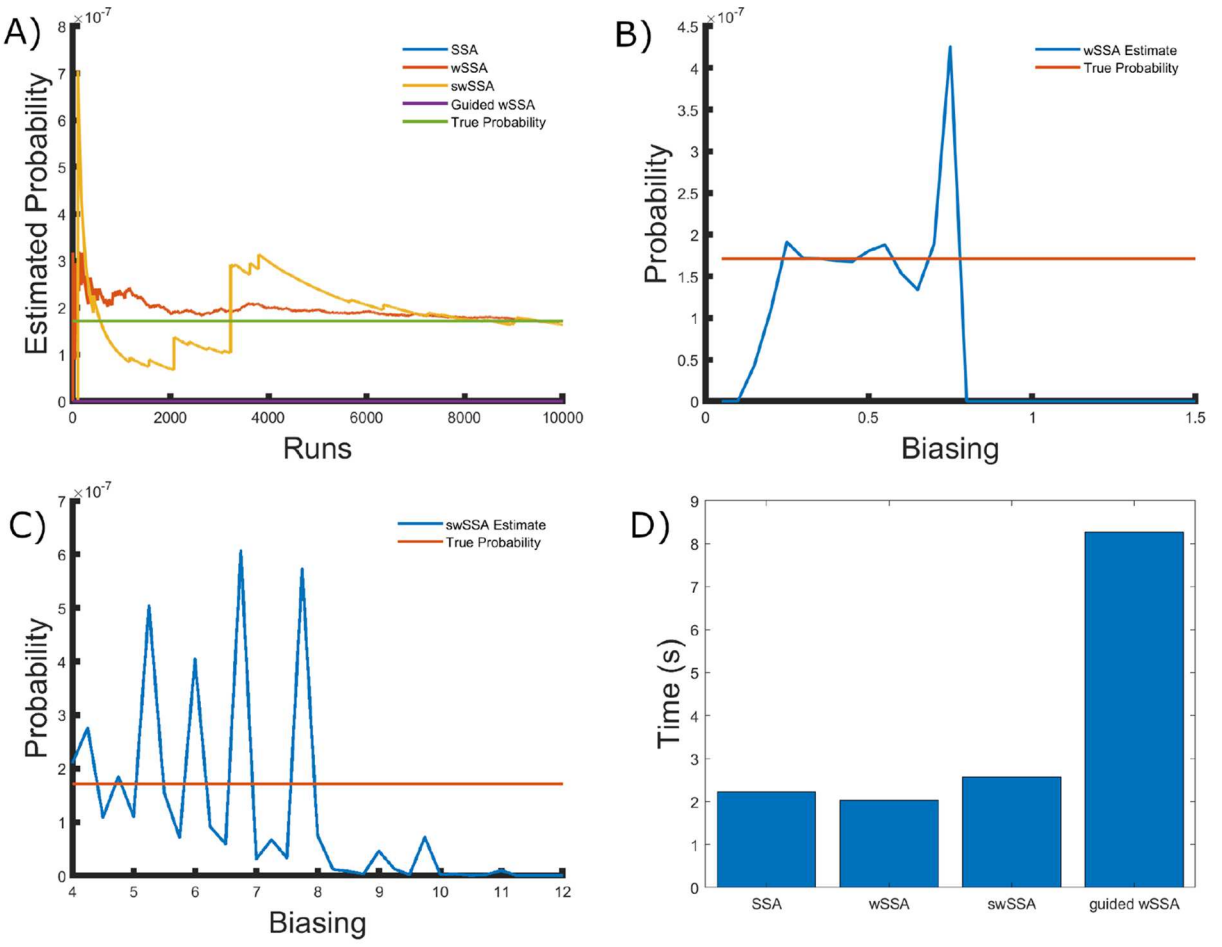


Figure 3. Analysis of the enzymatic futile cycle model. (A) Comparison of estimated probability to true probability for all four algorithms over 10^4 runs (with optimal biasing parameters). (B) Probability estimated by wSSA after 10^4 runs as δ is varied. Note that δ values in (0,1.5) were used despite $\delta > 1$ corresponding to reciprocal weighting. This is to demonstrate that, for some values of δ, the motivated scheme presented here performs no better than unmotivated biasing. (C) Probability estimated by swSSA after 10^4 runs as γ_{max} is varied. (D) Comparison of time to complete 10^3 runs for each algorithm. Time comparison results for all models were produced on a computer with an Intel i7 4-core 2.11 GHz processor and 8 GB of RAM, running Windows 10 Pro (v21H1), with optimal biasing parameters.

tional gain of the WE method over SSA is then calculated as $\frac{n^{SSA}}{N_{equivalent}}$.

In the following, the results for applying wSSA, swSSA, Guided wSSA, and WE to four chemical reaction network models are presented.

- *Enzymatic Futile Cycle* model was previously studied in wSSA, swSSA, and WE contexts showing the effectiveness of these methods in estimating rare-events, but those studies did not discuss the effect of choosing suboptimal biasing parameters on the accuracy of the estimate. Guided wSSA does not require selection of biasing parameters prior to running a simulation. In this experiment the performance of Guided wSSA is compared to that of WE and other IS-based methods and the effect of selecting suboptimal biasing parameters on the accuracy of the estimate is studied.
- *Simplified Motility Regulation* model was previously studied in Guided wSSA context, and it was shown that Guided wSSA can achieve significant computational gain over SSA without providing any biasing parameters prior to the simulation. In this experiment, the performance of wSSA, swSSA, and WE methods using near-optimal biasing parameters are compared to the performance of Guided wSSA. Also, the effect of selecting suboptimal biasing parameters on the accuracy of wSSA and swSSA methods is studied on this model.
- Since the objective of this paper is to determine if weighted stochastic simulation methods are suitable to be used for the analysis of genetic circuits, two genetic digital logic circuits showing glitching behavior were selected and experimented with. The performance of all IS-based and WE methods using near-optimal biasing parameters is studied on *Genetic Circuit 0 × 8E* and *Genetic Circuit 0 × 8E_TI* models. Also, the effects of selecting nonoptimal biasing parameters on the performance of wSSA and swSSA methods are studied on each of these two models.

Enzymatic Futile Cycle. The enzymatic futile cycle example is a simple six-reaction biochemical network given as follows:

$$R_1\colon S_1 + S_2 \xrightarrow{k1} S_3$$

$$R_2\colon S_3 \xrightarrow{k2} S_1 + S_2$$

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

$$R_3\text{: } S_3 \xrightarrow{k3} S_1 + S_5$$

$$R_4\text{: } S_4 + S_5 \xrightarrow{k4} S_6$$

$$R_5\text{: } S_6 \xrightarrow{k5} S_4 + S_5$$

$$R_6\text{: } S_6 \xrightarrow{k6} S_4 + S_2$$

where

$$k_1 = k_2 = k_4 = k_5 = 1, \quad k_3 = k_6 = 0.1$$

The initial state of the model is set as follows:

$$S_1(0) = S_4(0) = 1, \quad S_2(0) = S_5(0) = 50$$

$$S_3(0) = S_6(0) = 0$$

This system is representative of the futile cycle motif, which is common in natural biochemical systems, such as GTPase cycles, MAP Kinase cascades, and glucose mobilization.[16] This model serves as an archetypal naturally occurring biochemical system with few reactions.

The rare event of interest is S_5 falling to 25 molecules or fewer before 100 time units have passed, which is unlikely because the symmetry of the initial molecule counts and reaction rate constants keeps the system near its initial state with high probability. The exact probability of this event was reported by Kuwahara and Mura[8] to be 1.71×10^{-7}. This value is considered exact for all analyses.

The wSSA enhances the probability of observing a rare-event by biasing individual reaction rates upward or downward. In general, each reaction may be assigned a unique bias parameter, and each parameter is chosen prior to simulation. This paper uses a single wSSA biasing parameter, $\delta \in (0, 1)$, to simplify biasing optimization to a one-dimensional problem.

The swSSA method, like wSSA, biases reactions upward or downward using parameters selected prior to simulation. Unlike the wSSA, the swSSA applies biased rates selectively. For every reaction, two parameters are defined: (1) a relative propensity threshold beyond which biasing is applied, and (2) maximum biasing allowed for that reaction. This paper uses a fixed threshold of $\rho_D^0 = 0.15$ for reactions that are selected to be discouraged and a fixed threshold of $\rho_E^0 = 0.5$ for reactions that are selected to be encouraged. A single value for maximum biasing, $\gamma^{\max}$, is also selected for all selected reactions in a model in order to simplify biasing optimization to a one-dimensional problem.

In the enzymatic futile cycle model, the production of S_2 may be increased and the production of S_5 decreased to increase the probability that S_5 falls to 25. This gives the following wSSA weighting scheme:

$$b_1(\mathbf{x}) = a_1(\mathbf{x}), \quad b_2(\mathbf{x}) = a_2(\mathbf{x}), \quad b_3(\mathbf{x}) = \delta a_3(\mathbf{x})$$

$$b_4(\mathbf{x}) = a_4(\mathbf{x}), \quad b_5(\mathbf{x}) = a_5(\mathbf{x}), \quad b_6(\mathbf{x}) = \frac{1}{\delta} a_6(\mathbf{x})$$

Selecting R_6 to be encouraged and R_3 to be discouraged gives the following swSSA weighting scheme (f and g functions are described in detail in the swSSA subsection):

$$b_1(\mathbf{x}) = a_1(\mathbf{x}), \quad b_2(\mathbf{x}) = a_2(\mathbf{x}),$$
$$b_3(\mathbf{x}) = g(\rho_D^0, \gamma^{\max}, \mathbf{x}) a_3(\mathbf{x})$$
$$b_4(\mathbf{x}) = a_4(\mathbf{x}), \quad b_5(\mathbf{x}) = a_5(\mathbf{x}),$$
$$b_6(\mathbf{x}) = f(\rho_E^0, \gamma^{\max}, \mathbf{x}) a_6(\mathbf{x})$$

The Guided wSSA requires no hand-picked reaction weighting. Within this framework, to anecdotally demonstrate convergence comparisons, sample estimated probability plots were produced for each algorithm (Figure 3A). Additionally, wSSA and swSSA convergence was compared for various biasing parameters (Figure 3B,C). A runtime comparison was performed so that the number of runs could be compared for different algorithms (Figure 3D), and run weight mean and variance was compared for each algorithm at 10^3 runs (Table 1).

Table 1. Estimated Probability, Run Weight Variance, and 95% Confidence Interval for Each Simulation Method for the Enzymatic Futile Cycle Model at 10^3 Runs ($\mu_{\text{exact}} = 1.71 \times 10^{-7}$)[a]

Algorithm	μ_w	σ_w^2	95% CI
SSA	0	0	[0, 0]
wSSA	1.4753×10^{-7}	9.1059×10^{-13}	$[8.83 \times 10^{-8}, 2.06 \times 10^{-7}]$
swSSA	6.2517×10^{-8}	1.0426×10^{-12}	$[0, 1.25 \times 10^{-7}]$
Guided wSSA	0	0	[0, 0]

[a]Note that a tight confidence interval containing the true probability indicates near-optimal biasing, while zero probability and zero variance indicate poor biasing, as the error state was never reached. Near-optimal biasing parameters were used ($\delta = 0.5$, $\gamma_{\max} = 2.5$).

These results show that the wSSA with optimal biasing parameters converges much faster than other methods, but is very sensitive to biasing. The swSSA also converges quickly as well, but is slightly less sensitive to biasing. The Guided wSSA and SSA both converge poorly, and the Guided wSSA has much poorer runtime performance than any other method. A small but nonzero run weight variance might suggest that a weighting scheme is close to the ideal importance density, which would assign every run a weight equivalent to the true probability.

Next the WE method is applied to the example system. WE facilitates estimating the probability of rare events by dividing the model's state-space into bins and distributing trajectories to the bins that are likely to be undersampled. WE periodically checks the state of the simulation and enforces a fixed number of trajectories to be assigned to each bin. This limits the number of trajectories in highly reachable bins so that more computational resources can be dedicated to bins with lower reachability. Trajectories in low-probability bins are duplicated until the bin size reaches the fixed number assigned to it.

This paper implements the following WE binning procedure for enzymatic futile cycle model based on coordinate S_5: one bin containing states where $S_5 > 50$. Twenty-five bins for $S_5 \in \{50, 49, ..., 26\}$. An absorbing bin for the target states, $S_5 = 25$.

WE enforces each populated bin to contain a fixed number of trajectories by checking the bins after τ units of time. So, in addition to dividing the state-space into bins, a time period, τ, and the size of each bin should be set prior to the simulation. Running 1000 SSA runs of 100 time units each, the average reaction time for the enzymatic futile cycle model was estimated to be 0.2328 time units. The time period τ should be some value greater than this to minimize unnecessary polling of the bins. τ is set to 1 for this model.

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

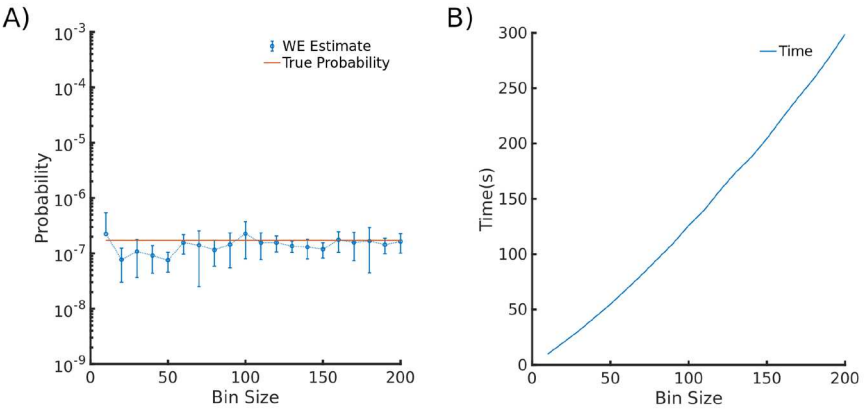


Figure 4. Analysis of the enzymatic futile cycle model. (A) 95% confidence interval constructed by running an ensemble of 15 WE simulations as bin size is varied from 10 to 200 with a step of 10. (B) Comparison of time to complete 15 WE simulations as bin size is varied from 10 to 200 with a step of 10.

Table 2. Number of Simulations, Estimated Probability, One-Standard-Error, Equivalent Number of SSA Runs Required to Produce the Same Level of Accuracy and the Gain Achieved over Brute-Force SSA for Each Simulation Method for the Enzymatic Futile Cycle Model ($p_{exact} = 1.71 \times 10^{-7}$)[a]

Method	N	$\hat{p}$	σn	n^{SSA}	Gain (n^{SSA}/N)
wSSA	10^5	1.73×10^{-7}	3.29×10^{-9}	1.60×10^{10}	1.60×10^5
swSSA	10^5	1.95×10^{-5}	1.45×10^{-8}	9.30×10^8	9.30×10^3
Guided wSSA	10^5	0	0		
WE	9.36×10^4	1.49×10^{-7}	2.23×10^{-8}	3.01×10^8	3.21×10^3

[a]Near-optimal biasing parameters were used ($\delta = 0.5$, $\gamma_{max} = 2.5$) for IS-based methods. An ensemble of 200 WE simulations was run to produce the results with the bin size set to 30. Note that the number of simulations, N, for the WE method is the number of SSA simulations computationally equivalent to the WE run.

This paper assigns to each bin the same target number of trajectories, called the "bin size", in order to simplify parameter selection to a one-dimensional problem. The results attained by running 15 WE simulations with the described framework are shown in Figure 4. As expected, estimates are more precise with increasing bin size. As with the wSSA and swSSA, performance is sensitive to choice in bin size. Unlike the wSSA and swSSA, however, a well-performing set of parameters may cause poor runtime performance.

Table 2 compares the performance of IS-based and WE methods on this model by reporting the computational gain achieved over running SSA for each method. In order to derive the standard error for the WE method, an ensemble of 200 independent WE runs was simulated, and the standard error of these 200 samples is reported. Note that for IS-based methods, column N reports the number of simulation runs that produced the reported estimate. For WE method, column N reports the number of SSA runs that is computationally equivalent to the WE simulation. Near-optimal biasing parameters are used for wSSA and swSSA. The same described binning framework is used for the WE method with the bin size being set to 30.

Table 2 shows that wSSA and swSSA both achieve significant computational gain over SSA given that near-optimal biasing parameters are selected, but the accuracy of estimator in these two methods is extremely sensitive to biasing parameters (Figure 3B,C). Guided wSSA was not able to produce a single trajectory ending in a state where $S_5 = 25$ after simulating 100,000 trajectories. Weighted Ensemble provides a safer option, as it is easier to select a good set of parameters compared to IS-based methods but the computational gain achieved over SSA was less than wSSA and swSSA.

Simplified Motility Regulation. The simplified motility regulation model is a 12-reaction naturally occurring gene regulatory network given as follows:

$$R_1: \text{codY} \xrightarrow{k1} \text{codY} + \text{CodY}$$

$$R_2: \text{CodY} \xrightarrow{k2} \phi$$

$$R_3: \text{flache} \xrightarrow{k3} \text{flache} + \text{SigD}$$

$$R_4: \text{SigD} \xrightarrow{k4} \phi$$

$$R_5: \text{SigD} \xrightarrow{k5} \text{SigD} + \text{hag} + \text{Hag}$$

$$R_6: \text{Hag} \xrightarrow{k6} \phi$$

$$R_7: \text{SigD} + \text{hag} \xrightarrow{k7} \text{SigD\_hag}$$

$$R_8: \text{SigD\_hag} \xrightarrow{k8} \text{SigD\_hag}$$

$$R_9: \text{CodY} + \text{flache} \xrightarrow{k9} \text{CodY\_flache}$$

$$R_{10}: \text{CodY\_flache} \xrightarrow{k10} \text{CodY} + \text{flache}$$

$$R_{11}: \text{CodY} + \text{hag} \xrightarrow{k11} \text{CodY\_hag}$$

$$R_{12}: \text{CodY\_hag} \xrightarrow{k12} \text{CodY} + \text{hag}$$

where

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

$$k_1 = k_8 = k_{10} = k_{12} = 0.1, \quad k_2 = k_4 = k_6 = 0.0002$$

$$k_3 = k_5 = 1, \quad k_7 = k_{11} = 0.01, \quad k_9 = 0.02$$

The initial state of the model is set as follows:

$$codY(0) = flache(0) = SigD\_hag(0) = hag(0)$$
$$= CodY\_flache(0) = CodY\_hag(0) = 1$$

$$CodY(0) = SigD(0) = Hag(0) = 10$$

This system represents the genetic mechanism which regulates flagella formation in *Bacillus subtilis*.[17,18] This model serves as an archetypal simple gene regulatory network.

The rare event of interest is CodY reaching 20 molecules before 10 time units have passed. The probability of this event was estimated to be 2.161×10^{-7} with 10^7 runs of the wSSA with the biasing setup described below and $\delta = 0.3$. This value is similar to estimates presented by Gillespie and Golightly[10] for the same model, and is considered exact for all analyses.

In the simplified motility regulation model, codY transcription/translation, Cod_flache dissociation, and CodY_hag dissociation may be increased, and CodY degradation, CodY_flache association, and CodY_hag association may be decreased to increase the probability that CodY rises to 20. This gives the following wSSA weighting scheme:

$$b_1(\mathbf{x}) = \frac{1}{\delta}a_1(\mathbf{x}), \quad b_2(\mathbf{x}) = \delta a_2(\mathbf{x}), \quad b_3(\mathbf{x}) = a_3(\mathbf{x})$$

$$b_4(\mathbf{x}) = a_4(\mathbf{x}), \quad b_5(\mathbf{x}) = a_5(\mathbf{x}), \quad b_6(\mathbf{x}) = a_6(\mathbf{x})$$

$$b_7(\mathbf{x}) = a_7(\mathbf{x}), \quad b_8(\mathbf{x}) = a_8(\mathbf{x}), \quad b_9(\mathbf{x}) = \delta a_9(\mathbf{x})$$

$$b_{10}(\mathbf{x}) = \frac{1}{\delta}a_{10}(\mathbf{x}), \quad b_{11} = \delta a_{11}(\mathbf{x}), \quad b_{12}(\mathbf{x}) = \frac{1}{\delta}a_{12}(\mathbf{x})$$

Again, we fix $\rho_D^0 = 0.15$ for all reactions selected to be biased downward and $\rho_E^0 = 0.5$ for all reactions selected to be biased upward. This gives the following swSSA weighting scheme:

$$b_1(\mathbf{x}) = f(\rho_E^0, \gamma^{\max}, \mathbf{x})a_1(\mathbf{x}),$$
$$b_2(\mathbf{x}) = g(\rho_D^0, \gamma^{\max}, \mathbf{x})a_2(\mathbf{x}), \quad b_3(\mathbf{x}) = a_3(\mathbf{x})$$

$$b_4(\mathbf{x}) = a_4(\mathbf{x}), \quad b_5(\mathbf{x}) = a_5(\mathbf{x}), \quad b_6(\mathbf{x}) = a_6(\mathbf{x})$$

$$b_7(\mathbf{x}) = a_7(\mathbf{x}), \quad b_8(\mathbf{x}) = a_8(\mathbf{x}),$$
$$b_9(\mathbf{x}) = g(\rho_D^0, \gamma^{\max}, \mathbf{x})a_9(\mathbf{x})$$

$$b_{10}(\mathbf{x}) = f(\rho_E^0, \gamma^{\max}, \mathbf{x})a_{10}(\mathbf{x}),$$
$$b_{11} = g(\rho_D^0, \gamma^{\max}, \mathbf{x})a_{11}(\mathbf{x}),$$
$$b_{12}(\mathbf{x}) = f(\rho_E^0, \gamma^{\max}, \mathbf{x})a_{12}(\mathbf{x})$$

Within this framework, the same analyses performed on the previous two models were performed on the simplified motility regulation model (Table 3, Figure 5). The results of these analyses corroborate those of the analyses performed on the previous two models.

WE requires dividing the state space of the model into bins prior to simulation. This paper implements the following binning procedure for the Motility Regulation model based on coordinate *CodY*: One bin containing states where *CodY* < 10. Ten bins for *CodY* ∈ {10, 11, ..., 19}. An absorbing bin for the target states, *CodY* = 20.

Table 3. Estimated Probability and Run Weight Variance for Each Simulation Method for the Simplified Motility Regulation Model at 10^3 Runs ($\mu_{exact} = 2.161 \times 10^{-7}$)[a]

Algorithm	μ_w	σ_w^2	95% CI
SSA	0	0	[0, 0]
wSSA	6.3233×10^{-8}	6.4963×10^{-13}	$[1.32 \times 10^{-8}, 1.13 \times 10^{-7}]$
swSSA	3.3120×10^{-8}	2.5388×10^{-13}	$[1.89 \times 10^{-9}, 6.43 \times 10^{-8}]$
Guided wSSA	6.4949×10^{-7}	2.4402×10^{-10}	$[0, 1.61 \times 10^{-6}]$

[a]Note that a tight confidence interval containing the true probability indicates near-optimal biasing, while zero probability and zero variance indicate poor biasing, as the error state was never reached. Near-optimal biasing parameters were used ($\delta = 0.3$, $\gamma_{max} = 8.25$).

Running 1000 SSA runs of 10 time units each, the average reaction time for this model was estimated to be 0.5071 time units. Setting τ to some value greater than this will minimize unnecessary polling of the bins. τ is set to 1 for this model.

The results attained by running 100 WE simulations with the described framework is shown in Figure 6. As was the case with the IS-based methods, the results of these analyses corroborate those of the analyses performed on the previous two models.

Table 4 compares the performance of IS-based and WE methods on a simplified motility regulation model by reporting the computational gain achieved over running SSA for each method. In order to derive the standard error for the WE method, an ensemble of 500 independent WE runs was simulated, and the standard error of these 500 samples is reported. For IS-based methods, column *N* reports the number of simulation runs that produced the reported estimate. For the WE method, column *N* reports the number of SSA runs that is computationally equivalent to the WE simulation. Near-optimal biasing parameters are used for wSSA and swSSA. The same described binning framework is used for WE method with the bin size being set to 150.

Table 4 shows that all the methods provide a considerable computational gain over SSA. Running 5×10^5 simulations, Guided wSSA provides a computational gain of 330, while the selection of biasing parameters is done totally automatically. We again observe that the accuracy of the estimate with wSSA and swSSA methods are very sensitive to the selection of biasing parameter (Figure 5B,C). Weighted ensemble again proves to be a safer option as it is easier to select a good set of parameters compared to IS-based methods but the computational gain achieved over SSA was less than that for IS-based methods with near-optimal biasing parameters.

Genetic Circuit 0x8E. The third example is the genetic circuit 0x8E originally published by Nielsen et al.[19] The circuit is one of 60 genetic digital logic circuits designed using the *genetic design automation* (GDA) tool Cello.[19] The circuit has three input molecules *arabinose* (Ara, ChEBI = 17535), *isopropyl-beta-*D-thiogalactopyranoside (IPTG, ChEBI = 61448) and *acetylcholine* (aTc, ChEBI = 15355) and one output, the production of *yellow fluorescent protein* (*YFP*).

The inducer molecules have to be present for a prolonged time so that the input states can propagate through the different levels of logic. The circuit has three input molecules and therefore eight input combinations: *IPTG*, *aTc*, *Ara* = (0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1). A one

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

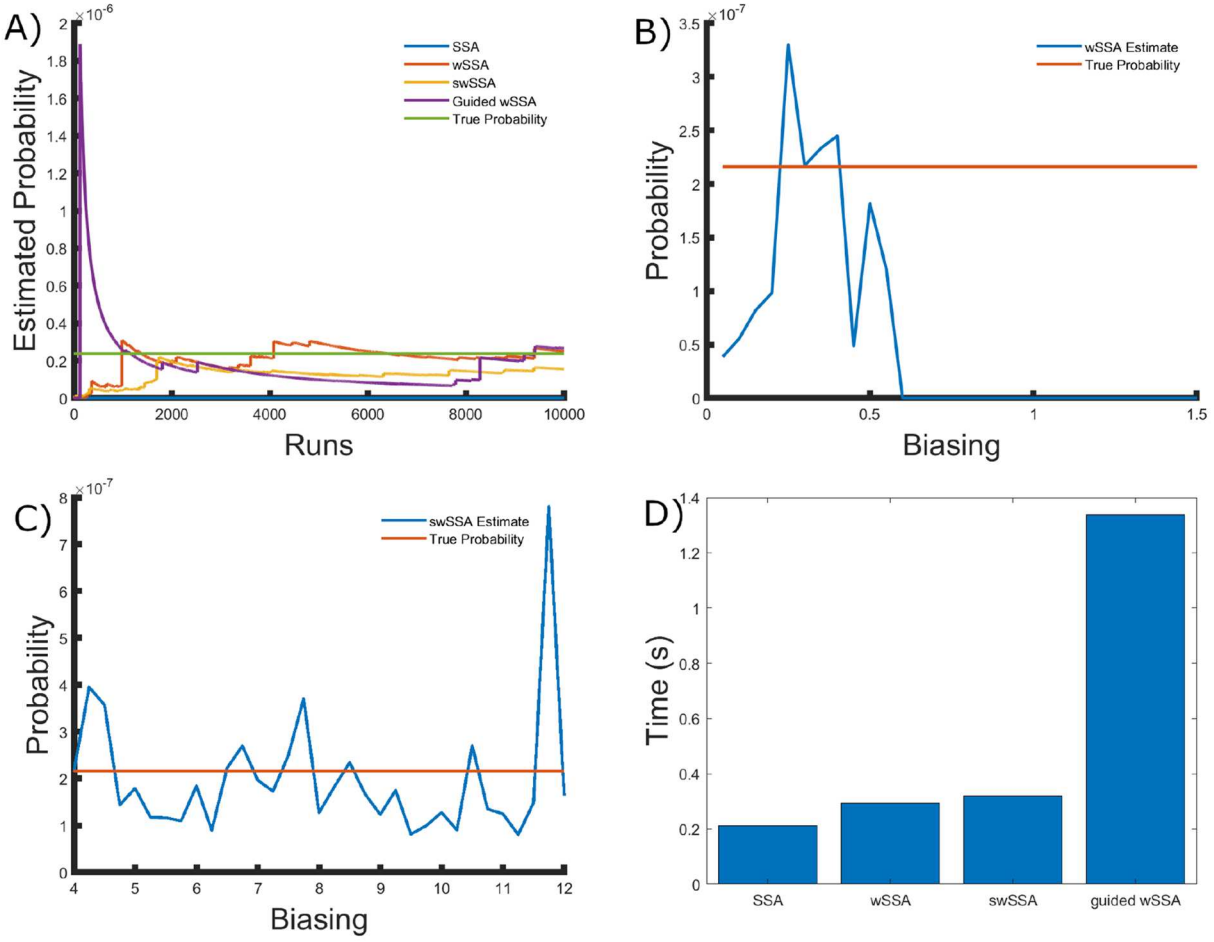


Figure 5. Analysis of the simplified motility regulation model. (A) Comparison of estimated probability to true probability for all four algorithms over 10^4 runs (with optimal biasing parameters). (B) Probability estimated by wSSA after 10^4 runs as δ is varied. Note that δ values in (0,1.5) were used despite $\delta > 1$ corresponding to reciprocal weighting. This is to demonstrate that, for some values of δ, the motivated scheme presented here performs no better than unmotivated biasing. (C) Probability estimated by swSSA after 10^4 runs as γ_{max} is varied. (D) Comparison of time to complete 10^3 runs for each algorithm (with $\delta = 0.3$, $\gamma_{max} = 8.25$).

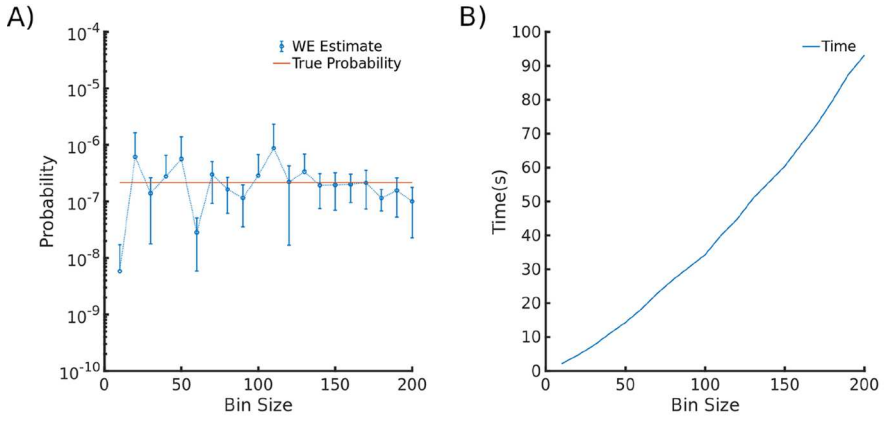


Figure 6. Analysis of the simplified motility regulation model. (A) 95% confidence interval constructed by running an ensemble of 100 WE simulations as bin size is varied from 10 to 200 with a step of 10. (B) Comparison of time to complete 100 WE simulations as bin size is varied from 10 to 200 with a step of 10.

represents a high input of 60 molecules and a zero represents a low input meaning no molecules of that input are available to the cell. For example, (0,0,1) means that zero molecules of *IPTG* or *aTc* are available but 60 molecules of *Ara*. Out of these eight different input combinations, four result in the production of *YFP*. For example, if (0,0,0) is applied, *YFP* is produced and the cell glows yellow. However, if (0,0,1) is applied, the circuit does not produce *YFP* and the cell is not glowing yellow.

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

Table 4. Number of Simulations, Estimated Probability, One-Standard-Error, Equivalent Number of SSA Runs Required to Produce the Same Level of Accuracy and the Gain Achieved over Brute-Force SSA for Each Simulation Method for the Simplified Motility Regulation Model ($p_{exact} = 2.161 \times 10^{-7}$)[a]

Method	N	$\hat{p}$	σn	n^{SSA}	Gain (n^{SSA}/N)
wSSA	5×10^5	2.07×10^{-7}	3.45×10^{-8}	1.74×10^8	3.49×10^2
swSSA	5×10^5	1.97×10^{-7}	1.74×10^{-8}	6.50×10^8	1.30×10^3
Guided wSSA	5×10^5	2.80×10^{-7}	4.12×10^{-8}	1.65×10^8	3.30×10^2
WE	4.09×10^5	3.75×10^{-7}	1.30×10^{-7}	2.20×10^7	5.37×10

[a]Near-optimal biasing parameters were used ($\delta = 0.3$, $\gamma_{max} = 8.25$) for IS-based methods. An ensemble of 500 WE simulations was run to produce the results with the bin size set to 150. Note that the number of simulations, N, for the WE method is the number of SSA simulations computationally equivalent to the WE run.

In their experiments on this circuit, the circuit exhibited a glitching behavior. If the circuit is in the state *IPTG*, *aTc*, *Ara* = (1, 0, 0), no *YFP* is produced. If the circuit is in the state *IPTG*, *aTc*, *Ara* = (1, 1, 1), there is also no *YFP* produced. However, Nielsen et al. observed that if the circuit is in the state (1, 0, 0) and then transitions to state (1, 1, 1), for a short amount of time, *YFP* is produced although this behavior is unwanted. In particular, both input combinations yield a low output, but instead of maintaining a low output throughout the input transition, the output switches from low, to high, to low. More information on the glitching behavior of circuit 0x8E can be found in refs 2 and 3.

The model consists of 15 reactions including 79 reaction rates. The rare event of interest is the quantification of that glitching behavior by determining the probability of *YFP* exceeding 70 molecules before 1000 time units pass. The probability of this event was estimated to be 6.29×10^{-4} with 10^6 runs of the SSA. This value is considered exact for all analyses.

YFP production may be increased and *YFP* degradation may be decreased to increase the probability that *YFP* rises to 70. This gives the following wSSA weighting scheme:

$$b_{13}(\mathbf{x}) = \frac{1}{\delta} a_{13}(\mathbf{x}), \quad b_{14}(\mathbf{x}) = \frac{1}{\delta} a_{14}(\mathbf{x}),$$
$$b_{15}(\mathbf{x}) = \delta a_{15}(\mathbf{x})$$

with all other reactions unbiased. Again, fixing $\rho_D^0 = 0.15$ and $\rho_E^0 = 0.5$ gives the following swSSA weighting scheme:

$$b_{13}(\mathbf{x}) = f(\rho_E^0, \gamma^{max}, \mathbf{x}) a_{13}(\mathbf{x}),$$
$$b_{14}(\mathbf{x}) = f(\rho_E^0, \gamma^{max}, \mathbf{x}) a_{14}(\mathbf{x}),$$
$$b_{15}(\mathbf{x}) = g(\rho_D^0, \gamma^{max}, \mathbf{x}) a_{15}(\mathbf{x})$$

Within this framework, the same analyses performed on the previous three models were performed on the genetic circuit 0x8E model (Table 5, Figure 7). The results of these analyses corroborate those of the analyses performed on the previous three models.

In order to estimate the probability of the event of interest using Weighted Ensemble, this paper implements the following binning procedure for the this model based on coordinate *YFP*: 70 bins for $YFP \in \{0, 1, ..., 69\}$. An absorbing bin for the target states, $YFP = 70$.

Running 1000 SSA runs of 1000 time units each, the average reaction time for this model was estimated to be 1.0052 time units. The time period τ should be some value greater than this to minimize unnecessary polling of the bins. τ is set to 1.5 for this model.

Table 5. Estimated Probability and Run Weight Variance for Each Simulation Method for the Genetic Circuit 0x8E Model at 10^3 Runs ($\mu_{exact} = 6.29 \times 10^{-4}$)[a]

Algorithm	μ_w	σ_w^2	95% CI
SSA	0	0	[0, 0]
wSSA	5.0043×10^{-4}	9.8086×10^{-5}	$[0, 1.11 \times 10^{-3}]$
swSSA	2.7317×10^{-4}	2.8359×10^{-5}	$[0, 6.03 \times 10^{-4}]$
Guided wSSA	2.7908×10^{-6}	7.7807×10^{-9}	$[0, 8.25 \times 10^{-6}]$

[a]Note that a tight confidence interval containing the true probability indicates near-optimal biasing, while zero probability and zero variance indicate poor biasing, as the error state was never reached. Optimal biasing parameters were used ($\delta = 0.6$, $\gamma_{max} = 4.5$).

The results attained by running 15 WE simulations with the described framework is shown in Figure 8. As was the case with the IS-based methods, the results of these analyses corroborate those of the analyses performed on the previous three models.

In order to compare the performance of IS-based and WE methods on this model, the computational gain achieved over running SSA for each method is reported in Table 6. In order to derive the standard error for the WE method, an ensemble of 150 independent WE runs was simulated and the standard error of these 150 samples is reported. For IS-based methods, column N reports the number of simulation runs that produced the reported estimate. For WE method, column N reports the number of SSA runs that is computationally equivalent to the WE simulation. Near-optimal biasing parameters are used for wSSA and swSSA. The same described binning framework is used for WE method with the bin size being set to 100.

As it can be observed from Table 6, Guided wSSA and swSSA achieve a computational gain of less than 1 over SSA with relatively large one-standard-error, indicating that SSA is expected to produce a more accurate estimate if it were to run for the same number of simulations. In the case of swSSA, it indicates the necessity of good biasing. Encouraging reactions 13 and 14 and discouraging reaction 15 seems to be helping the simulation reaching the error state more often, and selecting $\gamma_{max} = 4.5$ seems to be optimal by running 10^3 swSSA simulations (as it can be seen in Figure 7) but the resulting biasing scheme worsens the performance compared to SSA. Weighted Ensemble and wSSA both achieve a computational gain of more than 1 with the same limitation that wSSA performs very poorly in cases where poor biasing parameters are selected (Figure 7B).

Genetic Circuit 0x8E_TI. Circuit 0x8E_TI is a redesign of circuit 0x8E.[19] The circuit variation was designed by Fontanarrosa et al.[2] It has Two added Inverters to add extra delay to a pathway of the circuit to reduce its glitching behavior. The circuit implements the same logic function as circuit 0x8E and has therefore the same inputs (*Ara* (ChEBI = 17535), *IPTG*

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

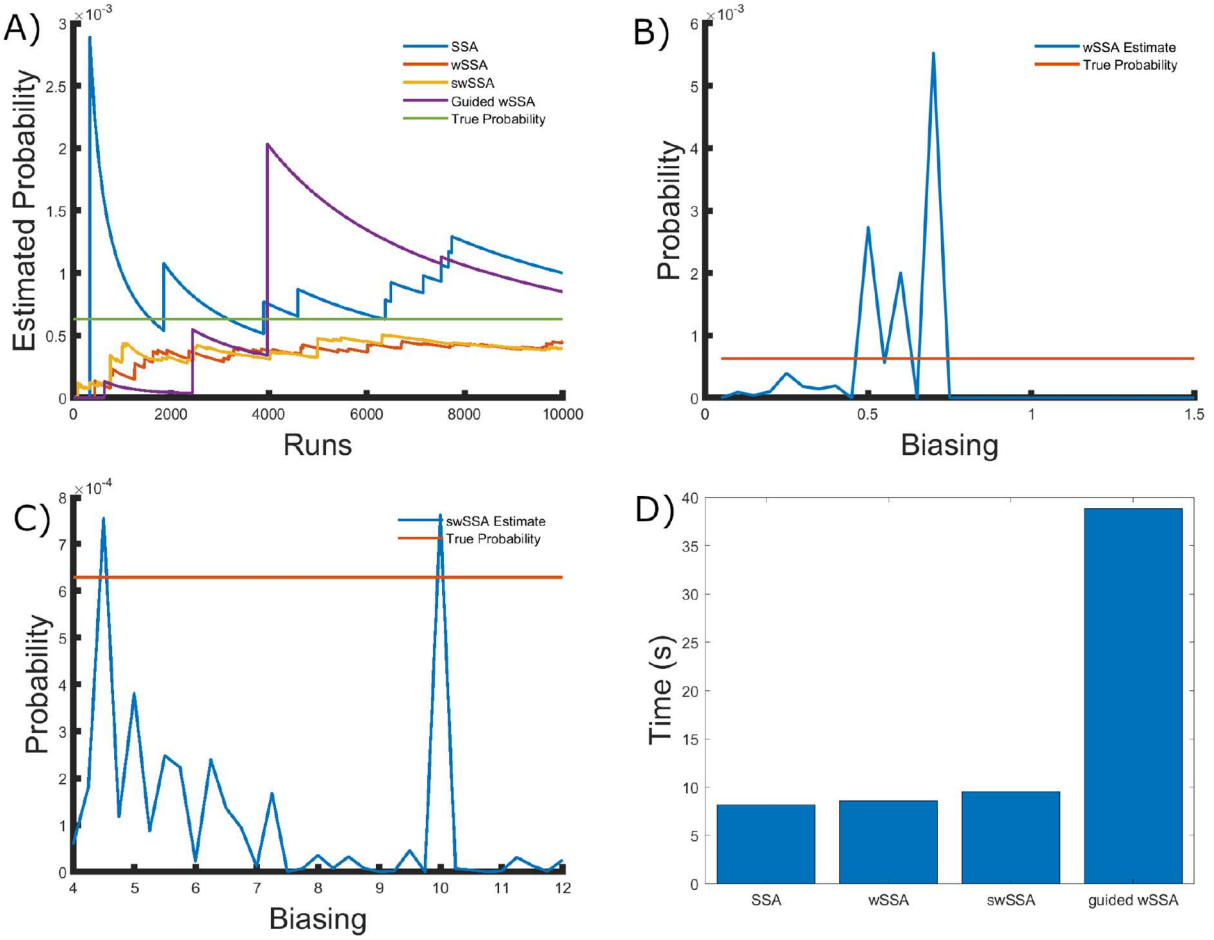


Figure 7. Analysis of the genetic circuit 0x8E model. (A) Comparison of estimated probability to true probability for all four algorithms over 10^4 runs (with optimal biasing parameters). (B) Probability estimated by wSSA after 10^2 runs as δ is varied. Note that δ values in (0,1.5) were used despite $\delta > 1$ corresponding to reciprocal weighting. This is to demonstrate that, for some values of δ, the motivated scheme presented here performs no better than unmotivated biasing. (C) Probability estimated by swSSA after 10^2 runs as γ_{max} is varied. (D) Comparison of time to complete 10^3 runs for each algorithm (with optimal biasing parameters).

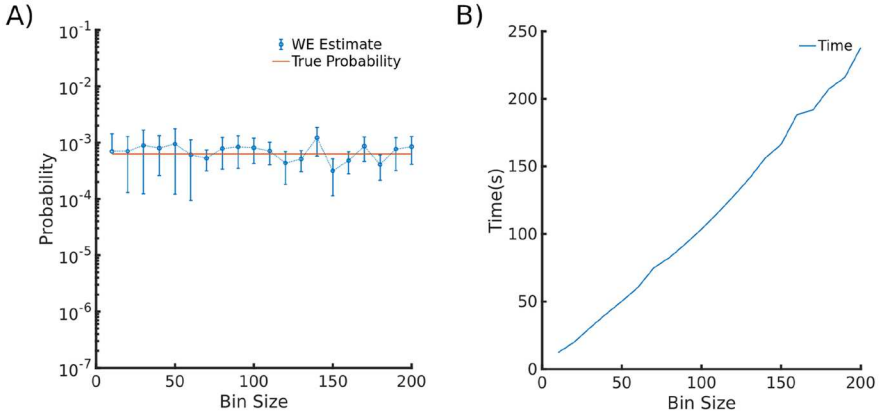


Figure 8. Analysis of the genetic circuit 0 × 8E model. (A) 95% confidence interval constructed by running an ensemble of 15 WE simulations as bin size is varied from 10 to 200 with a step of 10. (B) Comparison of time to complete 15 WE simulations as bin size is varied from 10 to 200 with a step of 10.

(ChEBI = 61448), *aTc* (ChEBI = 15355)), and output (*YFP*). The input combinations and the corresponding high or low output are identical to the circuit 0x8E as well. More information about the circuit, the inverters, and glitching behavior can be found in ref 2.

The model consists of 19 reactions including 109 reaction rates. The rare event of interest is the quantification of the glitching behavior by determining the probability of *YFP* exceeding 100 molecules before 1000 time units pass. The

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

Table 6. Number of Simulations, Estimated Probability, One-Standard-Error, Equivalent Number of SSA Runs Required to Produce the Same Level of Accuracy and the Gain Achieved over Brute-Force SSA for Each Simulation Method for the Circuit 0x8E Model ($p_{exact} = 6.29 \times 10^{-4}$)[a]

Method	N	$\hat{p}$	σn	n^{SSA}	Gain (n^{SSA}/N)
wSSA	5×10^4	5.80×10^{-4}	4.74×10^{-5}	2.58×10^5	5.15
swSSA	5×10^4	6.44×10^{-4}	1.70×10^{-4}	2.22×10^4	4.44×10^{-1}
Guided wSSA	5×10^4	2.56×10^{-4}	1.30×10^{-4}	1.51×10^4	3.01×10^{-1}
WE	5.73×10^4	5.66×10^{-4}	5.44×10^{-5}	1.91×10^5	3.33

[a]Near-optimal biasing parameters were used ($\delta = 0.6$, $\gamma_{max} = 4.5$) for IS-based methods. An ensemble of 150 WE simulations was run to produce the results with the bin size set to 100. Note that the number of simulations, N, for the WE method is the number of SSA simulations computationally equivalent to the WE run.

probability of this event was estimated to be 8.74×10^{-4} with 10^7 runs of the SSA. This value is considered exact for all analyses.

YFP production may be increased and *YFP* degradation may be decreased to increase the probability that *YFP* rises to 100. This gives the following wSSA weighting scheme:

$$b_{17}(\mathbf{x}) = \frac{1}{\delta} a_{17}(\mathbf{x}), \quad b_{18}(\mathbf{x}) = \frac{1}{\delta} a_{18}(\mathbf{x}),$$
$$b_{19}(\mathbf{x}) = \delta a_{19}(\mathbf{x})$$

with all other reactions unbiased. Fixing $\rho_D^0 = 0.15$ and $\rho_E^0 = 0.5$ gives the following swSSA weighting scheme:

$$b_{17}(\mathbf{x}) = f(\rho_E^0, \gamma^{max}, \mathbf{x}) a_{17}(\mathbf{x}),$$
$$b_{18}(\mathbf{x}) = f(\rho_E^0, \gamma^{max}, \mathbf{x}) a_{18}(\mathbf{x}),$$
$$b_{19}(\mathbf{x}) = g(\rho_D^0, \gamma^{max}, \mathbf{x}) a_{19}(\mathbf{x})$$

Within this framework, the same analyses performed on the previous four models were performed on the genetic circuit 0x8E_TI model (Table 7, Figure 9).

Table 7. Estimated Probability and Run Weight Variance for Each Simulation Method for the Genetic Circuit 0x8E_TI model at 10^3 runs ($\mu_{exact} = 8.74 \times 10^{-4}$)[a]

Algorithm	μ_w	σ_w^2	95% CI
SSA	0	0	[0, 0]
wSSA	8.9846×10^{-4}	4.2146×10^{-4}	$[0, 2.17 \times 10^{-3}]$
swSSA	1.2945×10^{-4}	5.9258×10^{-6}	$[0, 2.80 \times 10^{-4}]$
Guided wSSA	8.2357×10^{-5}	6.7493×10^{-6}	$[0, 2.43 \times 10^{-4}]$

[a]Note that a tight confidence interval containing the true probability indicates near-optimal biasing, while zero probability and zero variance indicate poor biasing, as the error state was never reached. Optimal biasing parameters were used ($\delta = 0.85$, $\gamma_{max} = 3.75$).

In order to estimate the probability of the event of interest using Weighted Ensemble, this paper implements the following binning procedure for the this model based on coordinate *YFP*: 100 bins for $YFP \in \{0, 1, ..., 99\}$. An absorbing bin for the target states, $YFP = 100$.

Running 1000 SSA runs of 1000 time units each, the average reaction time for this model was estimated to be 0.9159 time units. The time period τ should be some value greater than this to minimize unnecessary polling of the bins. τ is set to 1.5 for this model.

The results attained by running 15 WE simulations with the described framework is shown in Figure 10. As was the case with the IS-based methods, the results of these analyses corroborate those of the analyses performed on the previous three models.

To compare the performance of IS-based and WE methods on this model, the computational gain achieved over running SSA for each method is reported in Table 8 In order to derive the standard error for the WE method, an ensemble of 60 independent WE runs was simulated and the standard error of these 60 samples is reported. For IS-based methods, column N reports the number of simulation runs that produced the reported estimate. For the WE method, column N reports the number of SSA runs that is computationally equivalent to the WE simulation. Near-optimal biasing parameters are used for wSSA and swSSA. The same described binning framework is used for the WE method with the bin size being set to 150.

The results of this experiment confirm those shown for Genetic Circuit 0x8E. Guided wSSA does not require selection of biasing parameters prior to the simulation but produces an estimate less accurate than the one expected to be produced by SSA running the same number of simulations (Table 8). Weighted Ensemble, wSSA, and swSSA all achieve a computational gain more than 1 with the same limitation that wSSA and swSSA perform poorly in cases where poor biasing parameters were selected (Figure 9B,C). Weighted Ensemble achieves a smaller computational gain than wSSA but is safer with regards to the selection of parameters.

Discussion. The wSSA and swSSA require the user to select the reactions that should be encouraged and the reactions that should be discouraged. Although such a task might seem trivial for very simple models, deep insight into the underlying dynamics of the network is necessary for more complex models. This paper obtains weighting schemes by reasoning that certain reactions take a system toward or away from some state of interest, but it is unclear whether or not these schemes are optimal, nor is there a computationally efficient method of determining an optimal scheme. This pitfall undermines the applicability of both the wSSA and swSSA.

In the wSSA, the biasing factor for those selected reactions must also be specified prior to simulation. This is also nontrivial because these parameters can take on any value greater than zero and it is not obvious which values will result in accurate estimates just by considering the model. Moreover, the accuracy of the estimate is highly sensitive to these values (Figures 3, 5, 7, 9B). Selecting nonoptimal biasing factors can result in an estimate even less accurate than one produced by running the original SSA for the same number of simulations. This paper simplifies biasing schemes by introducing only one biasing factor, which was applied multiplicatively to down-biased reactions and inverse multiplicatively to up-biased reactions. This method simplifies biasing factor optimization to a one-dimensional problem, making parameters simpler to optimize but likely limiting the performance of the wSSA by greatly reducing the size of the biasing space. Even in the simplified one-

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

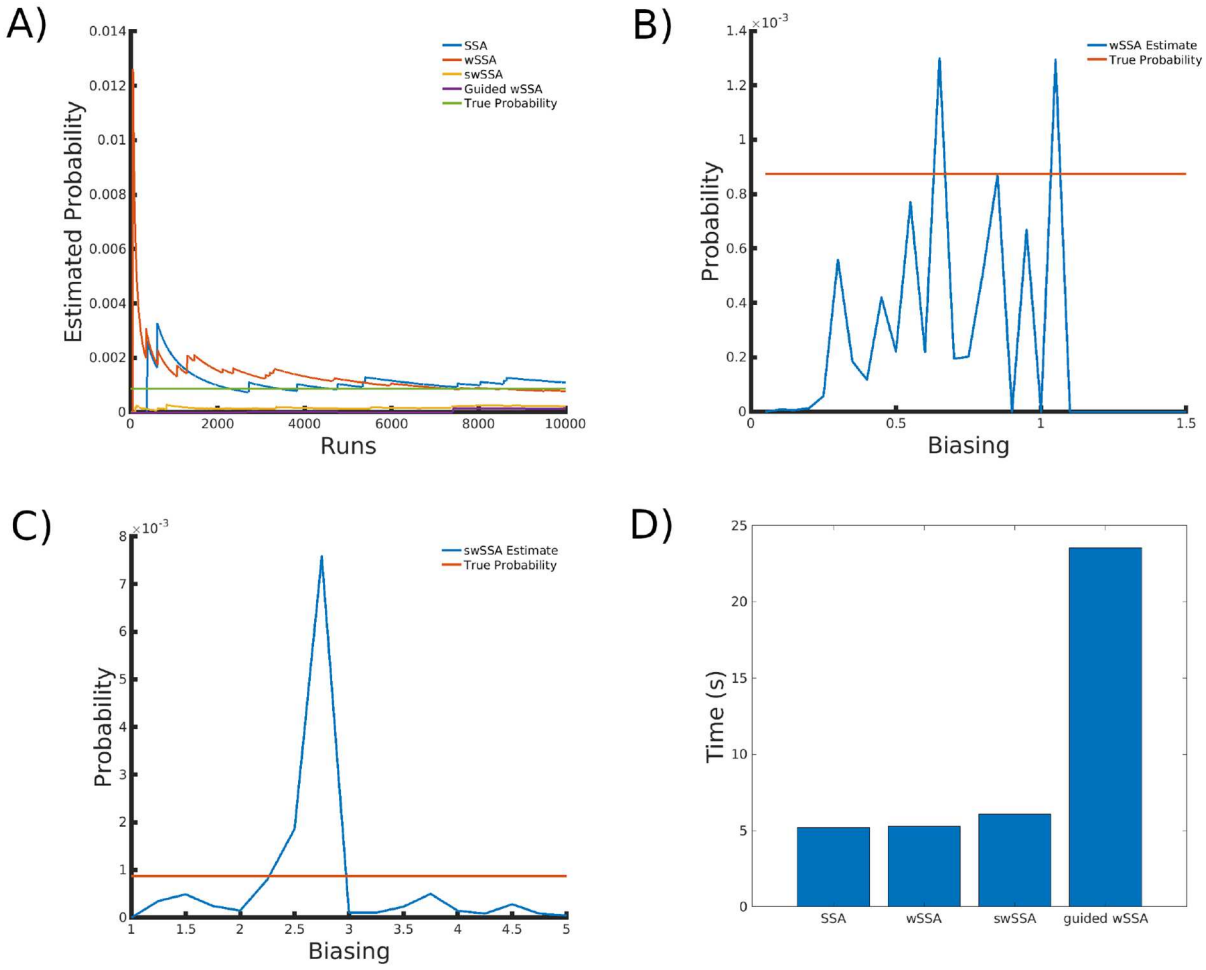


Figure 9. Analysis of the genetic circuit 0x8E_TI model. (A) Comparison of estimated probability to true probability for all four algorithms over 10^4 runs (with optimal biasing parameters). (B) Probability estimated by wSSA after 10^3 runs as δ is varied. Note that δ values in (0,1.5) were used despite $\delta > 1$ corresponding to reciprocal weighting. This is to demonstrate that, for some values of δ, the motivated scheme presented here performs no better than unmotivated biasing. (C) Probability estimated by swSSA after 10^3 runs as γ_{max} is varied. (D) Comparison of time to complete 10^3 runs for each algorithm (with optimal biasing parameters).

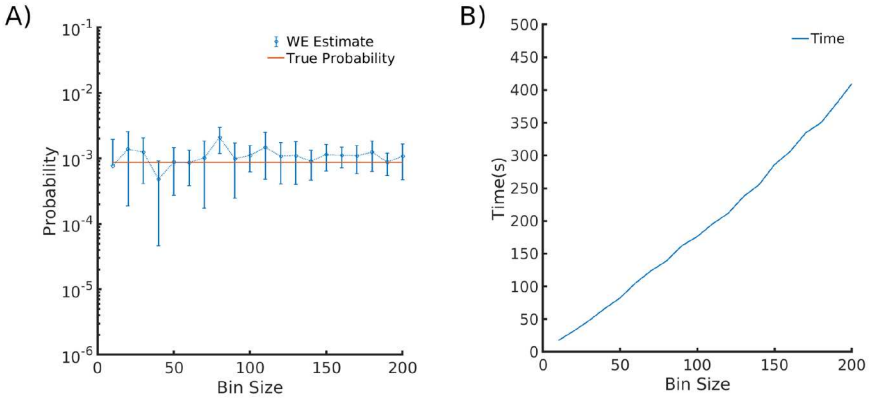


Figure 10. Analysis of the genetic circuit 0x8E_TI model. (A) 95% confidence interval constructed by running an ensemble of 15 WE simulations as bin size is varied from 10 to 200 with a step of 10. (B) Comparison of time to complete 15 WE simulations as bin size is varied from 10 to 200 with a step of 10.

dimensional case, there is no algorithmic way of producing optimal biasing parameters or determining whether a biasing parameter performs well. It is, therefore, not possible to know whether a given biasing setup is working well unless the true probability of the event of interest is known, making the wSSA ineffective in its intended use case.

Traditionally, a low run weight variance has been considered indicative of an optimal biasing scheme.[15] This measure alone is

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

Table 8. Number of Simulations, Estimated Probability, One-Standard-Error, Equivalent Number of SSA Runs Required to Produce the Same Level of Accuracy and the Gain Achieved over Brute-Force SSA for Each Simulation Method for the Circuit 0x8E_TI Model ($p_{exact} = 8.74 \times 10^{-4}$)[a]

Method	N	$\hat{p}$	σn	n^{SSA}	Gain (n^{SSA}/N)
wSSA	5×10^4	9.43×10^{-4}	9.40×10^{-5}	1.07×10^5	2.13
swSSA	5×10^4	2.29×10^{-4}	6.41×10^{-5}	5.56×10^4	1.11
Guided wSSA	5×10^4	5.41×10^{-4}	4.97×10^{-4}	2.18×10^3	4.38×10^{-2}
WE	5.02×10^4	1.32×10^{-3}	1.47×10^{-4}	6.10×10^4	1.21

[a]Near-optimal biasing parameters were used ($\delta = 0.85$, $\gamma_{max} = 3.75$) for IS-based methods. An ensemble of 60 WE simulations was run to produce the results with the bin size set to 150. Note that the number of simulations, N, for the WE method is the number of SSA simulations computationally equivalent to the WE run.

inadequate, however, because IS-based techniques have a tendency to underestimate the true probability with low but nonzero run weight variance (Tables 1, 3, 5, 7), which may yield confidence intervals that do not contain the true probability at all. In these cases, the IS-based techniques certainly still converge to the correct value, but may appear to have converged to an underestimate until (many simulation runs later) a high-weight run occurs which drastically changes the estimated probability and run weight variance. This effect is visible in the IS-based simulations, for example in Figure 7A, where a single sample after 8000 runs causes an almost order-of-magnitude jump in the Guided wSSA estimate.

The swSSA suffers from the same issues. For each reaction designated to be up-biased or down-biased, the maximum amount of biasing allowed, as well as a threshold from which encouragement/discouragement should be applied, must be set prior to simulation, so the user must specify twice as many biasing parameters as in the wSSA. As is the case with the wSSA, the accuracy of the swSSA is sensitive to these parameters (Figures 3, 5, 7, 9C). The swSSA generally exhibits more robustness to variation in biasing parameters than the wSSA, however.

The Guided wSSA eliminates the challenge of choosing a biasing scheme by requiring the user to specify no parameters whatsoever. To achieve this, however, the Guided wSSA relies on multivariate statistical techniques, which require matrix inversion, making the Guided wSSA inherently slower than the SSA, wSSA, and swSSA in simulating trajectories, typically taking around four times as long to complete the same number of runs as the SSA, wSSA, or swSSA (Figures 3, 5, 7, 9D).

The Guided wSSA also exhibits very poor convergence at a fixed number of runs, even relative to the SSA in many cases (Figures 3, 5, 7, 9A). This is likely because of the issue of negative weights (see the Methods section), which is solved using a mathematical technique that changes biasing parameters from their derived optimal value.

The WE method requires manual parameter selection, like the wSSA and swSSA, and suffers from the same issues. However, the WE requires three inputs regardless of model size. These are (1) state-space binning, (2) polling/checking period, and (3) bin size. This makes the WE method more scalable than its IS-based counterparts. As it is the case with the wSSA and swSSA, the WE performance is dependent on the values of these three inputs. However, there are some advantages. First, binning schemes are easier to develop than reaction weightings because the user need only know how to divide the state space into a series of 'steps' which lead to the state of interest, rather than having intuition for system dynamics, and a larger bin size is always more favorable, regardless of the system being studied. Second, the WE method exhibits more robustness to parameter variation than the wSSA (Figures 4A, 6A, 8A) and is more comparable to the swSSA, though a poor choice in bin size may still lead to a very large relative error.

The WE method also becomes much less time-efficient as bin size increases (Figures 4B, 6B, 8B), in contrast to the wSSA and swSSA whose runtime performances are generally unchanged by reaction biasing.

In summary, the original wSSA may achieve rapid convergence and lower variance than competing methods (Tables 1, 3, 5), but only with a narrow set of biasing parameters that cannot be reliably determined in general (Figures 3, 5, 7, 9B). The swSSA demonstrates broader robustness to biasing variation (Figures 3, 5, 7, 9C). The Guided wSSA solves the issue of biasing parameter determination, but has poor run-time performance and does not produce an accurate estimate in most of the experiments presented in this paper.

The WE method is not easily compared to IS-based methods in runtime performance or precision because it does not work by producing many independent simulation runs. The WE method, while suffering the same parameter sensitivity issues as the wSSA and swSSA, is more easily applied to larger models where many wSSA or swSSA reaction biases would need to be individually determined. It is also easier to choose an appropriate bin size, which need only be sufficiently large, rather than finely tuned reaction biasing parameters. This strength is undercut, however, by the WE method's poor runtime performance as bin sizes increase because a sufficient parameter scheme could be prohibitively inefficient. This is in contrast to the wSSA and swSSA, which, when optimally biased, are more efficient than the SSA because more simulations terminate before maximum simulation time by reaching the state of interest.

While the IS-based methods considered here each present unique strengths and challenges, none of them are able to consistently, efficiently, and exactly estimate the probability of rare events in synthetic biology, suggesting that further research and development is necessary for these methods to adequately study rare biological events. Because these methods may produce deceivingly low variances with poor estimates, the user cannot use them to reliably produce an accurate confidence interval, like they could when simulating a higher-probability event using the traditional SSA. At present, results produced by the methods considered here cannot be taken at face value. Instead, they must somehow be validated. In this paper, results were validated through analytical techniques or through SSA simulation. In practice, stochastic simulation of any kind are unnecessary if analytical techniques may be applied. Likewise, validating results with the SSA is impracticable in the case of rare event simulation. Using the SSA would incur a great deal of computational overhead and defeat the purpose of using an accelerated simulation method. It may be possible to mitigate

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

the computational overhead if the obtained parameters can be reused, for example when simulating the effect of small model variations. This could be useful for simulating changes in temperature or other environmental parameters if they do not appreciably alter the optimal IS parameters. This would serve as a useful application of the IS methods, but should still be used with cautious skepticism. Alternatively, multiple accelerated simulation methods could be applied to the same model and trusted only if they agree, reducing risk.

An ideal method would have the favorable efficiency and precision of the wSSA with ideal biasing parameters but would be autonomous like the Guided wSSA or more easily tuned like the WE Dynamics. An ideal method would also maintain good runtime efficiency as models grow in size or as probability of the event of interest decreases, like the wSSA and swSSA, suggesting that stratified sampling methods like the WE Dynamics will not be sufficient. This investigation does not suggest that the dynamic weighting techniques of the swSSA and Guided wSSA are necessary. Taken together, these conclusions point toward a method which autonomously determines a static weighting scheme, either by analyzing the structure of a given system or through an optimization technique, and otherwise simulates similarly to the wSSA or swSSA. The method of determining optimal biasing would need to perform much better than the Guided wSSA, which is shown here to perform far from optimally.

METHODS

Stochastic Simulation Algorithm. The SSA[6] is a Monte Carlo simulation procedure that numerically calculates the temporal behavior of a chemically reacting system. It assumes an input model as a well-stirred system of N chemical species $\{S_1, S_2, ..., S_N\}$ reacting through M irreversible reaction channels $\{R_1, R_2, ..., R_M\}$. The state vector $X(t) \equiv (X_1(t), X_2(t), ..., X_N(t))$ indicates the population of each species at time t and the system is initially at state $\boldsymbol{x}_0$, i.e. $X(0) = \boldsymbol{x}_0$. The state change vector $\boldsymbol{\nu}(j)$ is defined for each reaction R_j as the effect of that reaction on the system's state vector. Thus, given that the system is at state $\boldsymbol{x}$, a single reaction R_j would take the system to the state $\boldsymbol{x} + \boldsymbol{\nu}_j$.

At a state $X(t) = \boldsymbol{x}$, the propensity of reaction R_j, denoted $a_j(\boldsymbol{x})$, is defined such that $a_j(\boldsymbol{x})\,\mathrm{d}t$ is the probability that reaction R_j would fire in the next infinitesimal time interval $[t, t + \mathrm{d}t)$. Given this definition, the propensity of the reaction R_j at any given state would be the constant rate of that reaction, k_j, multiplied by the number of distinct combinations of molecular reactants of R_j at that state.

Given that the system is at state $\boldsymbol{x}$ at time t, i.e. $X(t) = \boldsymbol{x}$, the probability that the system leaves this state between time $t + \tau$ and $t + \tau + \mathrm{d}\tau$ is equal to $a_0(\mathbf{x})\ \mathrm{e}^{-a_0(\mathbf{x})\tau}\ \mathrm{d}\tau$ where $a_0 = \sum_{i=1}^{M} a_j(\mathbf{x})$ and the probability that reaction R_j is the reaction that takes the system to the next state is equal to $a_j(\boldsymbol{x})/a_0(\boldsymbol{x})$.

In order to simulate the temporal behavior of the system, starting from the initial state, the SSA selects the time τ to jump to a next state by sampling an exponential distribution with mean $1/a_0(\boldsymbol{x})$, and the index j of the reaction taking the system to the new state is chosen probablisitically with $P(j) = a_j(\boldsymbol{x})/a_0(\boldsymbol{x})$. To estimate the probability of a given property, SSA performs N simulation runs of the system and reports the estimated probability as the number of successful simulation runs (those satisfying the property) over the total number of simulations.

wSSA. Rare events are difficult to simulate because the number of traditional SSA runs necessary to see a rare event of interest occur even once can be very high. For example, an event with probability 10^{-6} would, on average, be observed once in 10^6 runs and would require many more runs for its probability to be estimated with a high degree of confidence. The wSSA addresses this issue by directing simulation runs toward the state of interest and carefully weighting those runs such that the sum of all run weights divided by the number of runs is an unbiased estimator of the true probability of reaching the state of interest.[8]

Simulation runs are directed toward the state of interest by increasing the likelihood of certain reactions occurring in simulation and decreasing the likelihood of others. Runs are assigned weights specific to the sequence of reactions which occurred during the run in a manner presented in Algorithm 1.

Scheme Algorithm 1. wSSA

Algorithm 1 wSSA

$q \leftarrow 0$
for $k = 1$ to n **do**
  $w \leftarrow 1$
  $t \leftarrow 0$
  $\mathbf{x} \leftarrow \mathbf{x_0}$
  evaluate all $a_j(\mathbf{x})$ and $b_j(\mathbf{x})$, and calculate $a_0(\mathbf{x})$ and $b_0(\mathbf{x})$
  while $t \leq t_{max}$ **do**
    if $x \in \varepsilon$ **then**
      $q \leftarrow q + w$
      Break out of while loop
    end if
    $\tau \leftarrow$ a sample of exponential variable with mean $1/a_0(\mathbf{x})$
    $u \leftarrow$ a sample of unit uniform random variable
    $\mu \leftarrow$ smallest integer satisfying $\sum_{i=1}^{\mu} \geq u b_0(\mathbf{x})$
    $w \leftarrow w \times (a_\mu(\mathbf{x})/b_\mu(\mathbf{x})) \times (b_0(\mathbf{x})/a_0(\mathbf{x}))$
    $t \leftarrow t + \tau$
    $\mathbf{x} \leftarrow \mathbf{x} + \nu_\mu$
    update $a_j(\mathbf{x})$ and $b_j(\mathbf{x})$, and recalculate $a_0(\mathbf{x})$ and $b_0(\mathbf{x})$
  end while
end for
report q/n as the estimated probability

While it is clear that reactions which move the system toward a rare event of interest should be biased upward and reactions which move the system away from a rare event of interest should be biased downward, Kuwahara and Mura[8] propose no method of determining the magnitude of biasing for each reaction.

swSSA. wSSA uses importance sampling in the reaction selection step of the SSA algorithm to sample parts of the sample-space with higher importance more frequently. Kuwahara and Mura[8] propose a fixed, predetermined biasing factor for each reaction in order to encourage reactions that increase the likelihood of reaching the target state and discourage those which decrease the likelihood of reaching the target state.

As the species population changes throughout the course of simulation, relative propensity of each reaction changes too. In order to produce an accurate estimate, the fixed biasing factor used in wSSA should bias relative propensity of reactions correctly for most of the values they take throughout the simulation. This results in a narrow range of values that the biasing factor can take to produce an accurate estimate. Moreover, if a fixed biasing factor is selected for a reaction and the relative propensity of that reaction takes a wide range of values (getting close to both 0 and 1 during the simulation), wSSA is going to inevitably modify some of those relative propensities too much or too little, resulting in a less accurate estimator. If a reaction is assumed to take the simulation toward the target states and the relative propensity of that reaction is already close to 1 in the current state of the simulation, there is no need for further encouragement of that reaction. The same is

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

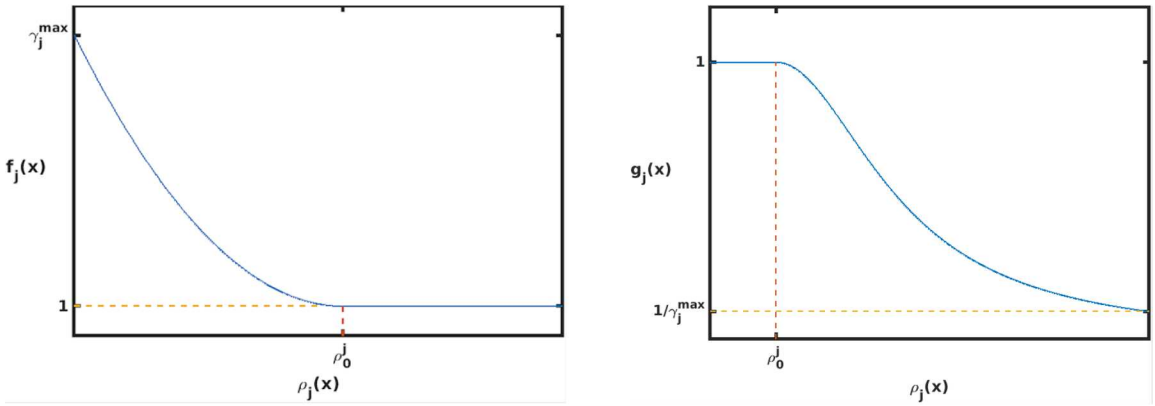


Figure 11. Example of parabolic *f* and *g* functions. γ_j^{max} is the maximum amount of biasing allowed for reaction *j*. ρ_0^j is the threshold beyond which biasing is applied.

true for a reaction assumed to take the simulation away from the target states which already has a relative propensity close to 0. In fact, further modifying relative propensities of reactions in those states would result in a less accurate estimate and decreases the efficiency of the algorithm as it results in oversampling a small part of the sample-space while neglecting other parts that could potentially be of high importance.

In order to address these issues, Roh et al.(2010)[9] propose a state-dependent biasing factor over the fixed biasing factor used in wSSA. The *state-dependent weighted stochastic simulation algorithm*, swSSA, calculates alternative propensities for reactions (b functions in line 6 of Algorithm 1) by multiplying current relative propensities by functions instead of positive constant scalars, as is the case in wSSA. The relative propensity of a reaction j, a_j/a_0, indicates the likelihood of that reaction being selected as the next step. Roh et al. define the parameter ρ_j to be the relative propensity of reaction j at state x, $\rho_j(x) = a_j(x)/a_0(x)$. Like wSSA, prior to simulation, reactions which are to be encouraged or discouraged should be identified. Reactions which are assumed to take the simulation toward target states are added to group G_E, reactions assumed to take the simulation away from target states are added to group G_D, and neutral reactions are added to group G_N. For each reaction j in groups G_E and G_D a threshold, ρ_j^0, and the maximum amount of biasing allowed for that reaction, γ_j^{max} is defined prior to the simulation. For a reaction j in group G_E at state x, if current relative propensity of that reaction, $\rho_j(x)$ is already greater than the predefined threshold for that reaction, ρ_j^0, no further biasing is applied. If $\rho_j(x)$ is less than ρ_j^0 then biasing is applied by multiplying the current propensity of that reaction by a biasing function, $f_j(x)$ which has the two following characteristics: (1) $f_j(x) \to \gamma_j^{max}$ as $\rho_j(x) \to 0$ and (2) $f_j(x) \to 1$ as $\rho_j(x) \to \rho_j^0$. For a reaction j in group G_D at state x, if the current relative propensity of that reaction, $\rho_j(x)$ is already less than the predefined threshold for that reaction, ρ_j^0, no further biasing is applied. If $\rho_j(x)$ is greater than ρ_j^0 then biasing is applied by multiplying the current propensity of that reaction by a biasing function, $g_j(x)$ which has the two following characteristics: (1) $g_j(x) \to 1/\gamma_j^{max}$ as $\rho_j(x) \to 1$ and (2) $g_j(x) \to 1$ as $\rho_j(x) \to \rho_j^0$.

In theory, any function with such characteristics can be used. Roh et al.[9] use parabolic functions, and the results presented in this paper are obtained by running swSSA with the same functions described in Roh et al. An illustration of parabolic *f* and *g* functions and their behavior is given in Figure 11 and the pseudocode of swSSA method can be viewed in Algorithm 2.

Scheme Algorithm 2. swSSA

Algorithm 2 swSSA

1: Partition all reactions into three groups: G_E, G_D, G_N
2: **for** all $R_j \in G_D$ **do**
3: choose $\rho_j^0 \in [0.1, 0.2]$
4: choose the initial value for γ_j^{max}
5: **end for**
6: **for** all $R_j \in G_E$ **do**
7: choose $\rho_j^0 \in [0.5, 0.6]$
8: choose the initial value for γ_j^{max}
9: **end for**
10: **for** all $R_j \in G_N$ **do**
11: $\gamma_j = 1 \quad \forall t,\ b_j(t) = a_j(t)$
12: **end for**
13: $q \leftarrow 0$
14: **for** $k = 1$ to n **do**
15: $w \leftarrow 1$
16: $t \leftarrow 0$
17: $\mathbf{x} \leftarrow \mathbf{x_0}$
18: evaluate all $a_j(\mathbf{x})$ and calculate $a_0(\mathbf{x})$
19: calculate $\rho_j(\mathbf{x})$ for all $R_j \in G_E$ and all $R_j \in G_D$
20: calculate all $\gamma_j(\mathbf{x})$; evaluate $b_j(\mathbf{x})$ and calculate $b_0(\mathbf{x})$
21: **while** $t \le t_{max}$ **do**
22: **if** $x \in \varepsilon$ **then**
23: $q \leftarrow q + w$
24: Break out of while loop
25: **end if**
26: $\tau \leftarrow$ a sample of exponential variable with mean $1/a_0(\mathbf{x})$
27: $u \leftarrow$ a sample of unit uniform random variable
28: $\mu \leftarrow$ smallest integer satisfying $\sum_{i=1}^{\mu} \ge u b_0(\mathbf{x})$
29: $w \leftarrow w \times (a_\mu(\mathbf{x})/b_\mu(\mathbf{x})) \times (b_0(\mathbf{x})/a_0(\mathbf{x}))$
30: $t \leftarrow t + \tau$
31: $\mathbf{x} \leftarrow \mathbf{x} + \nu_\mu$
32: update $a_j(\mathbf{x})$ and $\rho_j(\mathbf{x})$
33: recalculate $\gamma_j(\mathbf{x})$, update $b_j(\mathbf{x})$ and recalculate $b_0(\mathbf{x})$
34: **end while**
35: **end for**
36: report q/n as the estimated probability

Guided wSSA. To avoid reliance on user input and *a priori* knowledge of the system being simulated, the Guided wSSA automatically biases reactions at each point in time during a simulation. Gillespie and Golightly, motivated by results in a simple discrete-time model, argue that reactions are optimally biased when they cause the system to reach the state of interest immediately before the maximum simulation time.

In the Guided wSSA, therefore, biases are calculated such that the system is expected to reach the state of interest at maximum simulation time. This is achieved by assuming a constant reaction hazard for each reaction over the remainder of the simulation such that the number of times each reaction occurs

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

will follow a Poisson distribution. A multivariate normal approximation to the resulting Poisson distribution is made and also the conditioned expectation of reaction count over the remainder of the simulation for each reaction given that the state of interest is attained at maximum simulation time. The exact details of the matrix manipulations necessary to perform these multivariate statistical calculations are included in Algorithm 3.

Scheme Algorithm 3. Guided wSSA

Algorithm 3 Guided wSSA

$q \leftarrow 0$
for $k = 1$ to n **do**
  $w \leftarrow 1$
  $t \leftarrow 0$
  $\mathbf{x} \leftarrow \mathbf{x_0}$
  $\Delta t \leftarrow t_{max}$
  evaluate all $a_j(\mathbf{x})$ and calculate $a_0(\mathbf{x})$
  while $t \leq t_{max}$ **do**
    if $\mathbf{x}' = F^T\mathbf{x}$ **then**
      $q \leftarrow q + w$
      Break out of while loop
    end if
    evaluate all d_i as $d_i = e_i(\mathbf{x_t}) - \mathbf{x_i'}) \times \mathrm{sgn}(\mathbf{x_i'} - [F^T\mathbf{x_0}]_i), \quad i = 1, \ldots, u_0$
    if all $d_i > 0$ **then**
      $\mathbf{b}(\mathbf{x}) \leftarrow \mathbf{a}(\mathbf{x})$ for all j.
    else
      $\mathbf{b}(\mathbf{x_t}) \leftarrow \mathbf{a}(\mathbf{x_t} + H(\mathbf{x_t})S^TF(F^TSH(\mathbf{x_t}S^TF\Delta t)^{-1} \times (\mathbf{x}' - F^T[\mathbf{x_t} + S\mathbf{a}(\mathbf{x_t})\Delta t])$
    end if
    $\tau \leftarrow$ a sample of exponential variable with mean $1/a_0(\mathbf{x})$
    $u \leftarrow$ a sample of unit uniform random variable
    $\mu \leftarrow$ smallest integer satisfying $\sum_{i=1}^{\mu} b_j(\mathbf{x} \geq ub_0(\mathbf{x})$
    $w \leftarrow w \times (a_\mu(\mathbf{x})/b_\mu(\mathbf{x})) \times \exp[b_0(\mathbf{x}) - a_0(\mathbf{x})]\tau$
    $t \leftarrow t + \tau$
    $\Delta t \leftarrow t' - t$
    $\mathbf{x} \leftarrow \mathbf{x} + S^\mu$
    update $\mathbf{a}(\mathbf{x})$ and calculate a_0
  end while
end for
report q/n as the estimated probability

Unfortunately, the support of the multivariate normal distribution includes negative values, which are not meaningful in the context of reaction counts. For this reason, the published version of the Guided wSSA experiences regular errors when used in simulation. Inspection of the R code for the three example cases used by Gillespie and Golightly reveals that a different method of dealing with these negatives is used in each case.

The first of these methods, referred to as "Method A" here, increases each derived optimal biasing factor by the absolute value of the minimum biasing factor plus 0.01, thereby ensuring that no biasing factors will be negative. The second of these methods, referred to as "Method B" here, divides each biasing factor by the minimum biasing factor, which only ensures positive biasing if all biasing factors are negative, and does not work in general. The third and final method, referred to as "Method C" here, replaces each negative biasing factor with one, thereby eliminating negative reaction rates. Method C is used for all analyses included in this paper.

Weighted Ensemble. Weighted Ensemble (WE), originally introduced by Huber and Kim[12] in the context of molecular dynamic simulations and later applied to systems-biology models by Donovon et al.,[14] is a simulation strategy aiming at path sampling of rare events.

Considering a model's state-space, rare-events generally contain paths with states that have lower reachability probability from the initial state(s). WE increases the efficiency of rare-event sampling by limiting the amount of computational resources spent on sampling parts of the state-space with higher reachability probability. The resulting additional computational resources (CPU time) is then spent on further investigating trajectories which have already made it to parts of the state space with less reachability probability. This procedure will increase the number of trajectories reaching the states comprising the rare-event of interest and hence increasing the accuracy of the probability estimate for a fixed amount of computational resources compared to SSA. Donovon et al.[14] propose the following framework for applying WE to chemical reaction networks:

Prior to WE simulation, model's state space is divided into nonoverlapping bins, $\{b_1, b_2, ..., b_j\}$, and a target number of trajectories is then assigned to each of those bins, $\{M_1^{targ}, M_2^{targ}, ..., M_j^{targ}\}$. After every τ units of time, a polling of currently active trajectories takes place and the WE dynamics is applied to all populated bins as described in Algorithm 4. A

Scheme Algorithm 4. WE Dynamics

Algorithm 4 WE Dynamics

for each populated bin b_j **do**
  if size$(b_j) = M_j^{targ}$ **then**
    Do nothing
  end if
  if size$(b_j) < M_j^{targ}$ **then**
    Select one trajectory in b_j
    Split that trajectory into two new trajectories
    if size$(b_j) < M_j^{targ}$ **then**
      Jump to 6
    end if
  end if
  if size$(b_j) > M_j^{targ}$ **then**
    Select two trajectories in b_j
    Merge those two trajectories into one new trajectory
    if size$(b_j) > M_j^{targ}$ **then**
      Jump to 13
    end if
  end if
end for

populated bin is a bin that contains at least one trajectory at current state of the simulation. Selecting a trajectory for splitting is done probabilistically with the probability directly proportional to the trajectory's weight. After a trajectory is split into two, each of the newly spawned trajectories get half the weight of their parent. Selecting trajectories for merging is done probabilistically with the probability inversely proportional to the trajectories' weight. When two trajectories are merged, one of them survives probabilistically based on their relative weight. The weight of the surviving trajectory is updated by adding the weight of the killed trajectory.

τ is selected to be some value greater than the average reaction time of the model. Setting τ to some value less than that will result in excessive polling, as the system is not given enough time to evolve and move within bins. Dividing the state-space into bins should be done in a manner that states with the same reachability probability with respect to the event of interest are grouped together. Failure to do so will hinder the computational gains achieved by applying WE. After a WE run finishes, the aggregated weight of trajectories reaching the target bin is reported as the probability estimate. The confidence intervals reported for WE method in this paper are constructed by simulating an ensemble of WE simulations and using the standard error of the mean target bin.

■ ASSOCIATED CONTENT

Special Issue Paper

Invited contribution from the 13th International Workshop on Bio-Design Automation.

https://doi.org/10.1021/acssynbio.2c00553
ACS Synth. Biol. 2023, 12, 287−304

AUTHOR INFORMATION

Corresponding Author

Hao Zheng − *Department of Computer Science and Engineering, University of South Florida, Tampa, Florida 33620-9951, United States;* orcid.org/0000-0002-8627-0591; Email: haozheng@usf.edu

Authors

Mohammad Ahmadi − *Department of Computer Science and Engineering, University of South Florida, Tampa, Florida 33620-9951, United States*

Payton J. Thomas − *Department of Biomedical Engineering, University of Utah, Salt Lake City, Utah 84112, United States*

Lukas Buecherl − *Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, Boulder, Colorado 80309-0401, United States;* orcid.org/0000-0002-4844-6605

Chris Winstead − *Department of Electrical and Computer Engineering, Utah State University, Logan, Utah 84322-1400, United States*

Chris J. Myers − *Department of Electrical, Computer, and Energy Engineering, University of Colorado Boulder, Boulder, Colorado 80309-0401, United States;* orcid.org/0000-0002-8762-8444

Complete contact information is available at: https://pubs.acs.org/10.1021/acssynbio.2c00553

Author Contributions

[#]M.A. and P.J.T. contributed equally to this work.

Author Contributions

Payton J. Thomas and Mohammad Ahmadi implemented and tested the wSSA, swSSA, Guided wSSA, and WE Dynamics. Lukas Buecherl created and provided information about the model of genetic circuits 0x8E, 0x8E_LHF and 0x8E_TI. Hao Zheng mentored Mohammad Ahmadi throughout the project, and Chris J. Myers mentored Payton J. Thomas and Lukas Buecherl throughout the project. Chris Winstead offered input and valuable insight into the underlying mathematics of the methods analyzed. All authors contributed to the writing of the paper and created all figures in their entirety and all images used in the TOC graphic.

Funding

The authors of this work are supported by the National Science Foundation under Grant Nos. 1856733, 1856740, and 1900542. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

Notes

The authors declare no competing financial interest.

All MATLAB scripts created in this work are accessible at https://github.com/fluentverification/guided_proposals. It includes all simulators, biochemical models, and scripts used to generate results. This repository includes additional example networks which corroborate the findings presented here.

ACKNOWLEDGMENTS

The authors thank Colin Gillespie for providing the original Guided wSSA implementation and the members of the FLUENT Verification Project (https://fluentverification.github.io).

REFERENCES

(1) Esteller, M. Epigenetics in cancer. *New England Journal of Medicine* **2008**, *358*, 1148−1159.

(2) Fontanarrosa, P.; Doosthosseini, H.; Borujeni, A. E.; Dorfan, Y.; Voigt, C. A.; Myers, C. Genetic Circuit Dynamics: Hazard and Glitch Analysis. *ACS Synth. Biol.* **2020**, *9*, 2324−2338. PMID: 32786351.

(3) Buecherl, L.; Roberts, R.; Fontanarrosa, P.; Thomas, P. J.; Mante, J.; Zhang, Z.; Myers, C. J. Stochastic Hazard Analysis of Genetic Circuits in iBioSim and STAMINA. *ACS Synth. Biol.* **2021**, *10*, 2532−2540. PMID: 34606710.

(4) Gillespie, D.*Handbook of Materials Modeling*, Springer, 2005; Chapter 5.11.

(5) Samoilov, M. S.; Arkin, A. P. Deviant effects in molecular reaction pathways. *Nature biotechnology* **2006**, *24*, 1235−1240.

(6) Gillespie, D. T. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.* **1976**, *22*, 403−434.

(7) Gillespie, D. T. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* **1977**, *81*, 2340−2361.

(8) Kuwahara, H.; Mura, I. An efficient and exact stochastic simulation method to analyze rare events in biochemical systems. *J. Chem. Phys.* **2008**, *129*, 165101.

(9) Roh, M. K.; Gillespie, D. T.; Petzold, L. R. State-dependent biasing method for importance sampling in the weighted stochastic simulation algorithm. *J. Chem. Phys.* **2010**, *133*, 174106.

(10) Gillespie, C. S.; Golightly, A. Guided proposals for efficient weighted stochastic simulation. *J. Chem. Phys.* **2019**, *150*, 224103.

(11) Golightly, A.; Wilkinson, D. J. Bayesian inference for Markov jump processes with informative observations. *Statistical applications in genetics and molecular biology* **2015**, *14*, 169−188.

(12) Huber, G. A.; Kim, S. Weighted-ensemble Brownian dynamics simulations for protein association reactions. *Biophysical journal* **1996**, *70*, 97−110.

(13) Tse, M. J.; Chu, B. K.; Gallivan, C. P.; Read, E. L. Rare-event sampling of epigenetic landscapes and phenotype transitions. *PLoS computational biology* **2018**, *14*, No. e1006336.

(14) Donovan, R. M.; Sedgewick, A. J.; Faeder, J. R.; Zuckerman, D. M. Efficient stochastic simulation of chemical kinetics networks using a weighted ensemble of trajectories. *J. Chem. Phys.* **2013**, *139*, 115105.

(15) Gillespie, D. T.; Roh, M.; Petzold, L. R. Refining the weighted stochastic simulation algorithm. *J. Chem. Phys.* **2009**, *130*, 174103.

(16) Flomenbom, O.; Velonia, K.; Loos, D.; Masuo, S.; Cotlet, M.; Engelborghs, Y.; Hofkens, J.; Rowan, A. E.; Nolte, R. J.; Van der Auweraer, M.; et al. Stretched exponential decay and correlations in the catalytic activity of fluctuating single lipase molecules. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 2368−2372.

(17) Kearns, D.; Losick, R. Cell population heterogeneity during growth of Bacillus subtilis. *Genes & development* **2005**, *19*, 3083−94.

(18) Sonenshein, A. L.; Hoch, J. A.; Losick, R.*Bacillus subtilis and Its Closest Relatives*; John Wiley & Sons, Ltd, 2001; Chapter 1, pp 1−5.

(19) Nielsen, A. A. K.; Der, B. S.; Shin, J.; Vaidyanathan, P.; Paralanov, V.; Strychalski, E. A.; Ross, D.; Densmore, D.; Voigt, C. A. Genetic circuit design automation. *Science* **2022**, *352*, aac7341.