Aalto University

OPEN ACCESS

This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Vo, Thanh

# Efficient anticorrelated variance reduction for stochastic simulation of biochemical reactions

*Published in:*
IET SYSTEMS BIOLOGY

*DOI:*
10.1049/iet-syb.2018.5035

Published: 01/09/2018

*Document Version*
Peer reviewed version

*Please cite the original version:*
Vo, T. (2018). Efficient anticorrelated variance reduction for stochastic simulation of biochemical reactions. *IET SYSTEMS BIOLOGY*. https://doi.org/10.1049/iet-syb.2018.5035

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

IET Journals
The Institution of Engineering and Technology

# Efficient Anticorrelated Variance Reduction for Stochastic Simulation of Biochemical Reactions

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

*Vo Hong Thanh*[1]

[1] *Department of Computer Science, Aalto University, Finland and*
*The Microsoft Research - University of Trento Centre for Computational and Systems Biology (COSBI), Italy.*
** E-mail: thanh.vo@aalto.fi*

**Abstract:** We investigate the computational challenge of improving the accuracy of the stochastic simulation estimation by inducing negative correlation through the anticorrelated variance reduction technique. A direct application of the technique to the stochastic simulation algorithm (SSA), employing the inverse transformation, is not efficient for simulating large networks because its computational cost is similar to the sum of independent simulation runs. We propose in this paper a new algorithm that employs the propensity bounds of reactions, introduced recently in our rejection-based SSA (RSSA), to correlate and synchronize the trajectories during the simulation. The selection of reaction firings by our approach is exact due to the rejection-based mechanism. In addition, by applying the anticorrelated variance technique to select reaction firings, our approach can induce substantial correlation between realizations, hence reducing the variance of the estimator. The computational advantage of our rejection-based approach in comparison with the traditional inverse transformation is that it only needs to maintain a single data structure storing propensity bounds of reactions, which is only updated infrequently, hence achieving better performance.

## 1 Introduction

Biochemical reactions at cellular level are inherently stochastic processes. Their occurrences are random events occurring after many random collisions between molecular species. The stochastic effect is further amplified when the involved species have low copy numbers (often refereed to as *population*). It has been repeatedly shown that the stochastic noise plays an important role in the development of biological processes [1–4]. The noise may further propagate between cells and lead to significant biological responses [5–7]. Stochastic chemical kinetics has been adopted as a promising framework to investigate the stochastic behavior of biological reactions by allowing explicit incorporation of stochastic noise into the description of biological systems.

The stochastic chemical kinetics framework keeps track a discrete population of each species, forming the system state, and assigns each possible reaction between species a probability to fire proportional to its *propensity*, depending on the reaction kinetics and its reactants [8]. The temporal dynamics of the system state can be fully expressed by the chemical master equation (CME) and its realizations can be obtained by the stochastic simulation algorithm (SSA) [9]. The basis of SSA is a Monte Carlo sampling where a reaction is randomly chosen with probability proportional to its propensity to update the state. SSA is an exact simulation procedure in the sense that it does not introduce approximation error in the sampling of reaction firings. Many methods have been introduced for implementing the Monte Carlo sampling step including the well-known direct method (DM)[9, 10], the next reaction method (NRM) [11] and their improvements [12–20] as well as others [21–28]. The rejection-based SSA (RSSA), introduced recently by Thanh *et al.* [29, 30], provides an alternative approach for an exact realization of the next reaction firings. Its principle is to abstract exact propensity of each reaction with an interval, denoted by a pair of propensity lower bound and upper bound, encompassing all possible propensity values, and employ these bounds to select reaction firings. The next reaction firing in RSSA is selected in two steps. First, a reaction is randomly chosen according to its propensity upper bound. Second, a rejection-based test is applied to the reaction to verify the selection to ensure it fires with correct probability. Further improvements for the basic RSSA procedure above are discussed in recent work [31–35].

The statistical results obtained by SSA will converge to the solution of CME in the limit of an infinite number of simulation runs; however, due to limited computational resources, often a fixed number of simulation runs will be performed. The convergence rate in the estimation of population of species by simulation is measured by its variance, hence the confidence interval of the prediction. The estimator in the case of independent SSA runs is not efficient because its variance is proportional to the variance of each simulation run. Variance reduction techniques have been introduced (see, e.g., [36–38] for a comprehensive review of the techniques) to reduce the variance of the estimator. The anticorrelated variance reduction, in particular, is an attempt to improve the estimation by inducing negative correlation between the simulation realizations. The correlation between simulations is achieved by transforming the same stream of the random numbers used for the simulation. The theoretical analysis of such technique for the stochastic simulation of biochemical processes with constant rates have been recently developed [39–41]. Even though the proposed strategy [41] is able to apply the anticorrelated variance technique to SSA based on the inverse transformation, it is not efficient. In fact, the sampling strategy performing two simulation runs with the correlated stream of random numbers has the same computational cost of running two independent SSA runs in simulating large models. We investigate in this paper the efficient use of the anticorrelated variance technique to induce dependence between stochastic simulations of biochemical reactions to improve both the accuracy of the estimation and its computational efficiency. We tackle the computational challenge with a new algorithm that employs the propensity bounds and the rejection-based approach to correlate the simulations. We cope with the synchronization of simulations when rejections of reactions occur in applying the anticorrelated variance technique by decomposing the complex rejection simulation procedure. Our algorithm is simpler to implement than the strategies suggested by Schmelser and Kachltvlchyanukul [44] where two random generators are used to synchronize the trajectories during the simulation. Our new algorithm reduces the variance of the estimator, while it is still able to achieve better performance in comparison with direct inverse transformation. We note


© The Institution of Engineering and Technology 2015

our proposed strategy in this paper shares the same spirit with the ones proposed recently in [42, 43], which are used to compute the sensitivities of biochemical reactions.

The paper is organized as follows. Section 2 reviews the basics of the stochastic approach for simulation and analysis of biochemical reactions. Section 3 presents our new algorithm for improving accuracy of simulation estimation by the anticorrelated variance reduction technique relied on the rejection-based stochastic simulation approach. We first recall the background of the rejection-based simulation framework with the concept of propensity bounds. Then, we describe in detail our new approach to couple and to induce correlation between simulated realizations. Section 4 benchmark our algorithm to concrete to demonstrate its efficiency. The concluding remarks are in section 5.

## 2 Stochastic simulation

We consider a well-mixed biochemical reactor volume of $N$ molecular species labelled $S_i$ for $i = 1, \ldots, N$. The state $X(t)$ of the system at a time $t$ is a $N$-vector $X(t) = (X_1(t), ..., X_N(t))$ where $X_i(t)$ denotes the population of species $S_i$ present in the system at that time. Species can interact with other species to produce necessary products for the development of the biological system through $M$ reactions denoted as $R_j$, $j = 1, \ldots, M$, each of which has a general form:

$$R_j : v_{1j}S_1 + ... + v_{nj}S_n \xrightarrow{c_j} v'_{1j}S_1 + ... + v'_{nj}S_n \qquad (1)$$

in which we call the species on the left side: *reactants* and the ones on the right side: *products*. The rate of the reaction is specified by the stochastic constant $c_j$. The number of molecules of a reactant consumed and the number of molecules of a product produced after firing reaction $R_j$ is determined by the non-negative integer $v_{ij}$ and $v'_{ij}$, respectively, which are called *stoichiometric coefficients*. The vector $v_j$ whose $i$-th is equal to $v'_{ij} - v_{ij}$ quantifies the net change in the system state by firing reaction $R_j$. Thus, assume that the current time $t$ and the state $X(t)$, and that a reaction $R_j$ is scheduled to fire later at time $t + \tau$. Then at the scheduled time $t + \tau$, the system moves to a new state as $X(t + \tau) = X(t) + v_j$.

The stochastic chemical kinetics assigns to each reaction $R_j$, $j = 1, \ldots, M$, a *propensity* $a_j$, defined such that, given the system state $X(t)$ at time $t$, the quantity $a_j(X(t))dt$ gives the probability that $R_j$ fires in the next infinitesimal time $t + dt$. In general, the propensity function $a_j$ is dependent on the state $X(t)$ and the reaction kinetics. The existence of the propensity function is called the *fundamental hypothesis* of the stochastic chemical kinetics [9]. For standard mass action kinetics, propensity $a_j$ of a reaction $R_j$ exists and is computed as:

$$a_j(X(t)) = c_j h_j(X(t)) \qquad (2)$$

where $c_j$ is its corresponding rate constant and $h_j(X(t))$ counts the number of distinct combinations of reactants involved in the reaction. For *synthesis reaction*, where species are introduced to the system from an external source, the count of combinations of reactants is set to $h_j(X(t)) = 1$.

The probability distribution of the system state under the stochastic chemical kinetics setting is fully described by CME; however, an analytical analysis of CME, except for trivial cases, is difficult to obtain due to the problem of high dimensional state space. SSA [9] is often the choice to realize sample trajectories of the system state $X(t)$. In each simulation iteration, SSA drives the system to a reachable state by firing one reaction at a time. The reaction firing as well as its firing time of SSA is obtained from sampling the joint probability density function (pdf) $p(\tau, \mu)$, defined such that, given the current state $X(t)$ at time $t$, $p(\tau, \mu)d\tau$ gives the probability that reaction $R_\mu$ fires in the next infinitesimal time $t + \tau + d\tau$. The analytical form of $p(\tau, \mu)$ [9] is given by:

$$p(\tau, \mu) = a_\mu exp(-a_0\tau) \qquad (3)$$

where $a_0 = \sum_{j=1}^{M} a_j$.

In Eq. 3, integrating $p(\tau, \mu)$ over $\tau$ from 0 to $\infty$ shows that the reaction firing reaction $R_\mu$ defines a discrete probability $a_\mu/a_0$; summing $p(\tau, \mu)$ over reaction index $\mu$ from 1 to $M$ gives that the firing time $\tau$ follows an exponential distribution $\texttt{Exp}(a_0)$. These observations are the foundation for the implementation of SSA, For each simulation iteration, propensities $a_j$ for $j = 1 \ldots M$ are computed. Then, the reaction firing time $\tau$ is computed:

$$\tau = \frac{1}{a_0} ln\left(\frac{1}{r_1}\right) \qquad (4)$$

and the reaction firing $R_\mu$ is realized:

$$\mu = \text{smallest reaction index such that: } \sum_{j=1}^{\mu} a_j > r_2 a_0 \qquad (5)$$

where random numbers $r_1, r_2 \sim \texttt{U}(0, 1)$. SSA advances the time to $t + \tau$ and moves the system to a new state $X(t + \tau) = X(t) + v_\mu$ accordingly. The simulation repeats until a predefined ending time is reached.

### 2.1 Simulation analysis

A conventional approach to analyze the dynamics of $X(t)$ at a given time $t$ due to stochastic noise in biochemical reactions is to perform many runs of SSA to generate its realizations. Specifically, let $K$ be the number of simulation runs. Let $X^r$, $r = 1 \ldots K$, be realizations of state $X^*$ obtained by repeatedly performing $K$ runs of SSA under the same simulation conditions. By the law of large numbers, the statistical properties of the ensemble of $K$ realizations $X^r$, $r = 1 \ldots K$, obtained by SSA will converge to the ones by solving CME.

To be concrete, let us concentrate on estimating the statistical mean $\mathbb{E}[X]$ of state $X$ given the ensemble of $K$ simulation realizations. Consider the case when the samples $X^r$ with $r = 1 \ldots K$ are realized by $K$ independent runs of SSA. Let $<X>$ be the sample mean of state $X$ obtained from the ensemble of $K$ simulations. It is calculated as:

$$<X> = \frac{\sum_{r=1}^{K} X^r}{K} \qquad (6)$$

The sample mean $<X>$ is an unbiased estimator of the true mean and it is ensured to approach the true mean $\mathbb{E}[X]$ under the condition that $K$ approaches infinity, i.e., $\lim_{K\to\infty} <X> = \mathbb{E}[X]$. However, in practice, the number of simulation $K$ is often fixed due to limited computational resources. To measure the convergence and accuracy of the estimator $<X>$, the standard deviation, i.e., the square root of the sample variance, is often used. Let $Var[<X>]$ be the variance of the estimator by $K$ independent runs of SSA. We have

$$Var[<X>] = \frac{Var[\sum_{r=1}^{K} X^r]}{K^2} = \frac{\sum_{r=1}^{K} Var[X^r]}{K^2} = \frac{Var[X]}{K} \qquad (7)$$

in which the second equality is obtained using the fact that $X^r$, $r = 1 \ldots K$, are independent and the last derivation is obtained because $X^r$ is an exact realization of $X$. Because the variance $Var[X]$ of $X$ is often unknown, in practice it is approximated by the (unbiased) sample variance $s^2$ as

$$s^2 = \frac{\sum_{r=1}^{K} (X^r - <X>)^2}{K - 1} \qquad (8)$$

*we omit the time $t$ when it clear from the context


© The Institution of Engineering and Technology 2015

### 2.2 Anticorrelated variance technique

The anticorrelated variance technique is an approach to reduce the variance of the estimator by inducing negative correlation between $X^r$, $r = 1 \dots K$. We consider here the case in which the correlation is introduced between two realizations, but it can be extended to multiple realizations [39]. Assume that $K$ trajectories are divided into groups of two (antithetic) realizations denoted $X^r$ and $X^{-r}$ where $r = 1, \dots, K/2$. Assume also that $X^r$ and $X^{-r}$ are negatively correlated, i.e., their covariance $cov(X^r, X^{-r}) < 0$. The estimator based on the anticorrelated variance technique is:

$$<X>^A = \frac{\sum_{r=1}^{K/2}(X^r + X^{-r})/2}{K/2} \tag{9}$$

Note that $<X>^A$ is also an unbiased estimation for the true mean. The variance of the estimator by anticorrelated sampling, however, is smaller than the one of independent sampling. Specifically, we have

$$\begin{aligned} Var[<X>^A] &= \frac{Var[\sum_{r=1}^{K/2}(X^r + X^{-r})]}{K^2} \\ &= \frac{\sum_{r=1}^{K/2}(Var[X^r] + Var[X^{-r}] + 2cov(X^r, X^{-r}))}{K^2} \\ &< \frac{\sum_{r=1}^{K/2}(Var[X^r] + Var[X^{-r}])}{K^2} = Var[<X>] \end{aligned} \tag{10}$$

because $cov(X^r, X^{-r}) < 0$.

The correlation by the anticorrelated variance technique can be applied directly to SSA based on the inverse transformation. Let $F(x)$ be the cumulative distribution function (cdf) of a random variable $Z$. Then the two random numbers generated as $Z^r \sim F^{-1}(v)$ and $Z^{-r} \sim F^{-1}(1 - v)$ in which $F^{-1}(x)$ denotes the inversion of $F(x)$ and $v \sim U(0, 1)$, are negatively correlated. In Algorithm 1, we outline the implementation of the anticorrelated variance technique into SSA, which we call anticorrelated variance SSA (AV-SSA), to sample two realizations $X^r$ and $X^{-r}$ by using the inverse transformation.

The main simulation loop of AV-SSA is in lines 2 - 13. For each iteration, two random numbers $r_1$ and $r_2 \sim U(0, 1)$ are generated. These random numbers are used to sample the next state for $X^r$ by SSA. Then, they are transformed using the anticorrelated variance technique (line 6) to sample the the next state for $X^{-r}$.

Algorithm 1: Anticorrelated Variance SSA

1: initialize state $X^r = X^{-r} = x_0$ and time $t^r = t^{-r} = 0$
2: **repeat**
3: generate two random numbers $r_1, r_2 \sim U(0, 1)$
4: **for** (each trajectory $k \in \{r, -r\}$) **do**
5: **if** $(k == -r)$ **then**
6: set $r_1 = 1 - r_1$ and $r_2 = 1 - r_2$
7: **end if**
8: compute propensity $a_j(X^k)$, for $j = 1 \dots m$, with state $X^k$ and total propensity $a_0^k = \sum_{j=1}^{m} a_j(X^k)$
9: compute $\tau = (1/a_0^k)\ln(1/r_1)$
10: select minimum index $\mu$ such that $\sum_{j=1}^{\mu} a_j(X^k) > r_2 a_0^k$
11: update $X^k = X^k + v_\mu$ and time $t^k = t^k + \tau$
12: **end for**
13: **until** $(t^k \geq T_{max}$ for all trajectories $k \in \{r, -r\})$

The number of random numbers for each simulation step of AV-SSA is two, while in the case of independent SSA runs is four. Thus, the performance of AV-SSA is better than the independent SSA runs in simulating small models because for these models the computational cost for random number generation significantly affects the simulation runtime. A drawback of AV-SSA when simulating large models is that its performance is the same as running two independent SSA runs because the simulation in this setting is largely contributed by the cost of selecting the next reaction firings and propensity updates.

## 3 Anticorrelated variance reduction with rejection-based approach

We present in this section an efficient simulation algorithm that uses the anticorrelated variance technique to reduce the variance of the estimator based on the rejection-based SSA (RSSA) [30]. We rely on propensity bounds of reactions to couple the simulations and then employ rejection-based procedure to verify the selection. To cope with rejections of reactions, we decompose the rejection-based selection procedure in the original RSSA procedure into simple steps. Our algorithm is not only easy to implement, but also has better performance than the strategy described in the previous section.

### 3.1 Rejection-based simulation background

We recall the basics of RSSA [30], which is introduced recently to reduce the number of propensity calculations during the simulation. The mathematical framework for the simulation of RSSA is the rejection-based sampling technique. Employing such a rejection-based mechanism allows RSSA to exactly sample the pdf $p(\tau, j|x, t)$ in Eq. 3.

For its simulation purpose, RSSA abstracts propensity of each reaction $R_j$, $j = 1, \dots, M$, into an interval $[\underline{a_j}, \overline{a_j}]$, enclosing the exact value of the current propensity $a_j(X(t))$. The propensity bound $[\underline{a_j}, \overline{a_j}]$ are derived by constraining the current state $X(t)$ to an arbitrary interval $[\underline{X}, \overline{X}]$. Having these propensity bounds, the selection of the next reaction firing and its time in RSSA is performed as follows. A candidate reaction $R_\mu$ for firing is randomly chosen proportionally to its propensity upper bound $\overline{a_\mu}$. The candidate $R_\mu$ is realized such that it is the smallest reaction index $\mu$ satisfying: $\sum_{j=1}^{\mu} \overline{a_j} > r_1 \cdot \overline{a_0}$ in which $r_1 \sim U(0, 1)$ and $\overline{a_0} = \sum_{j=1}^{m} \overline{a_j}$. The candidate then goes through a rejection-based test with success probability $a_\mu(X(t))/\overline{a_\mu}$ to rectify the selection. The implementation of the rejection-based test requires to compute the exact propensity $a_\mu(X(t))$, but RSSA tries to avoid this by exploiting the propensity lower bound with noting that if $R_\mu$ is accepted with probability $\underline{a_\mu}/\overline{a_\mu}$, then it is also accepted with probability $a_\mu(X(t))/\overline{a_\mu}$. To implement this, let $r_2 \sim \mathtt{U}(0, 1)$ be a random number. RSSA checks whether the condition $r_2 \leq \underline{a_\mu}/\overline{a_\mu}$ holds. If it is the case, then $R_\mu$ is accepted to fire without computing the exact propensity $a_\mu(X(t))$. Otherwise, the exact propensity has to be evaluated and the check $r_2 \leq a_\mu/\overline{a_\mu}$ is performed. If the candidate reaction $R_\mu$ is accepted after the rejection-based test, RSSA generates its firing time. Otherwise, a new candidate is selected and a new rejection test is performed.

Let $k$ be the number of consecutive trials until the candidate reaction $R_\mu$ is accepted. The firing time $\tau$ of the accepted candidate $R_\mu$ is obtained by sampling the Erlang distribution $\mathtt{Erlang}(k, \overline{a_0})$ in which the shape parameter is $k$ and the rate parameter $\overline{a_0}$. In fact, the firing time $\tau$ of an accepted candidate is chosen to be the sum of $k$ independent exponentially distributed numbers with the same rate $\overline{a}_0$. The correctness proof for the chosen distribution of the reaction firing time in RSSA is discussed in detail in Thanh *et al.* [30]. This fact allows us to decompose the rejection-based selection process of RSSA and we will exploit it to synchronize the generation of samples when applying anticorrelated variance technique.


© The Institution of Engineering and Technology 2015

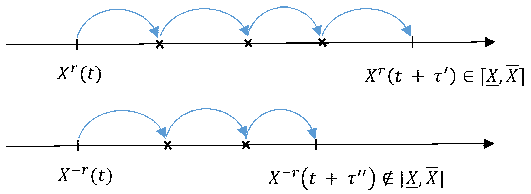


**Fig. 1**: Our of synchronization in the simulation of $X^r$ and $X^{-r}$ due to different number of trials leading to acceptance of reactions.

### *3.2 Anticorrelated realizations with rejection-based simulation*

There are two problems that need to be addressed for using the anticorrelated variance technique with the rejection-based approach: the coupling of realizations and the synchronization of these realizations during the simulation. The coupling requires to efficiently correlate the realizations. The synchronization of realizations is easy to achieve for inverse transformation; rejections make it harder because the number of steps leading to the acceptance of the candidate reaction varies between realizations. We discuss in the following our approach to solve these issues.

We correlate the simulation of states $X^r$ and $X^{-r}$ by defining the global propensity bounds that confine all the values of reaction propensities in the two realizations. Specifically, for each reaction $R_j, j = 1 \dots M$, we compute a lower bound $\underline{a_j}$ and an upper bound $\overline{a_j}$ such that $\underline{a_j} \leq a_j^k \leq \overline{a_j}$ for all $k \in \{r, -r\}$. Although it is possible to derive these global propensity bounds as the maximum and minimum values from each realization, this approach requires to maintain two separated data structures. We choose a simpler solution that only needs to handle a single data structure. More in details, we compute the propensity bounds $\underline{a_j}$ and $\overline{a_j}$ by defining a fluctuation interval $[\underline{X}, \overline{X}]$ that bounds all populations of each species in both the realizations $X^r$ and $X^{-r}$, i.e., $\underline{X}_i \leq X_i^r, X_i^{-r} \leq \overline{X}_i$ for each species $S_i$ with $i = 1, \dots N$. It is done by defining the fluctuation interval as $[\underline{X_i}, \overline{X_i}] = [(1-\delta_i)X_i^{min}, (1+\delta_i)X_i^{max}]$ where $X_i^{min} = \min(X_i^r, X_i^{-r})$ and $X_i^{max} = \max(X_i^r, X_i^{-r})$, respectively, and $0 < \delta_i < 1$ is a predefined parameter. In practice, a value of parameter $\delta_i$ chosen between 10% - 20% for $\delta_i$ often gives the good performance. Repeating this procedure on each species, we are forming a fluctuation interval $[\underline{X}, \overline{X}]$ that bounds all species in the reaction network. The propensity bounds $\underline{a_j}/\overline{a_j}$ are then derived by minimizing/maximizing the propensity function $a_j$ on such fluctuation interval $[\underline{X}, \overline{X}]$.

Knowing the lower bounds $\underline{a_j}$ and upper bounds $\overline{a_j}$ of reactions, we can apply the rejection-based approach to select reaction firings to update the state $X^k$ for $k \in \{r, -r\}$. However, the difference in the number of steps leading to the acceptance of the candidate reaction raises a difficulty for the synchronization of realizations. Let us consider an example described in Figure 1. In the figure, the state $X^r$ is updated after 4 trials while the second state $X^{-r}$ moves to a new one after 3 trials. In addition, the state $X^{-r}$ even moves out of the fluctuation interval, hence a new fluctuation interval has to be defined, while the $X^r$ does not require to define a new one. The complicated situation is due to the complex rejection-based selection for the next reaction in the ordinary RSSA algorithm encompassing many steps. Candidate reactions are selected for the rejection test, but only accepted ones considered, while all rejected candidates are ignored. Once the candidate reaction is accepted its time is generated.

We cope with the synchronization problem by decomposing the multistep rejection-based selection into single steps. We rely on the fact that the time for each trial of a candidate reaction is exponentially distributed with rate $\overline{a}_0$. Specifically, for each trial, we will select a candidate reaction $R_\mu$ and assign to it the time $\tau$, generated from an exponential distribution $Exp(\overline{a}_0)$. The candidate is then inspected through a rejection-based using the exact propensity value evaluating with the corresponding state. In particular, the candidate $R_\mu$ will be accepted to fire and to update the corresponding state $X^k$, $k \in \{-r, r\}$, with success probability $a_\mu(X^k)/\overline{a_\mu}$. After the update, if there existing a new state that is no longer consistent with current fluctuation interval, then a new fluctuation interval is updated and new propensity bounds of reactions are computed. In this way, the simulation of realizations $X^{-r}$ and $X^r$ will be synchronized.

We outline in Algorithm 2 the detailed implementation of our Anticorrelated Variance RSSA. To facilitate the update of species and involving reactions when its population move out of the fluctuation interval, we maintain the set UpdateSpeciesSet (initialized to empty in line 7), which contains species that should update their fluctuation interval due to the reaction firings, and a Species-Reaction (SR) dependency graph [30], which is a directed bipartite graph showing the dependency of reactions on species, to retrieve the reactions affected by species. To build the SR dependency graph in line 2, we inspect through each pair of species $S_i$ and reaction $R_j$ and add a directed edge from $S_i$ to $R_j$ to the graph if a change in the population of species $S_i$ induces a change in the propensity $a_j$ of reaction $R_j$.

The simulation loop for selecting reaction firings is in lines 9 - 31. For each iteration, three random numbers $r_1$, $r_2$ and $r_3 \sim U(0,1)$ are generated. The random number $r_1$ is used to generate the waiting time (line 13) while $r_2$ and $r_3$ are used to select the candidate reaction and validate it through a acceptance-rejection test (lines 15 - 24). The random numbers are then transformed to produce the anticorrelated realization. If the firings of reactions cause the population of species $S_i$ moving out of its fluctuation interval, the species will be added to UpdateSpeciesSet for further processing.

The computation of the new fluctuation interval when there is existing a species moving out of its fluctuation interval is implemented in lines 32 - 38. First, a species $S_i \in$ UpdateSpeciesSet is retrieved and its new fluctuation interval is computed. Then, the reactions affected by species $S_i$, denoted by the set ReactionsAffectedBy$(S_i)$, are extracted from the SR dependency graph. Each reaction $R_j \in$ ReactionsAffectedBy$(S_i)$ will be retrieved to compute for new propensity bounds.

Let us sketch the proof for the exactness of the marginal distribution of states $X^{-r}$ and $X^r$, respectively, by Algorithm 2. We have for each selection loop in lines 9 - 31, a candidate reaction $R_\mu$ is selected with probability $\overline{a}_\mu/\overline{a_0}$ (line 15) and its waiting time is generated following an exponentially distributed $Exp(\overline{a}_0)$ (line 13). During the rejection test in lines 16 - 24, the candidate reaction $R_\mu$ is checked through the rejection test by the exact propensity evaluating with the corresponding state. In particular, let us consider the state $X^{-r}$. The candidate reaction $R_\mu$ is accepted with probability $a_\mu(X^{-r})/\overline{a}_\mu$. So, the probability that a candidate reaction is selected and accepted to update $X^{-r}$ is $a_\mu(X^{-r})/\overline{a}_0$. Let $k_{-r}$ be the number of trials until a candidate reaction is accepted. The firing time of the accepted reaction is the sum of $k_{-r}$ exponential random numbers with the same rate $Exp(\overline{a}_0)$, hence following an Erlang $\texttt{Erlang}(k_{-r}, \overline{a}_0)$ distribution. From this point, the correctness argument of RSSA in Theorem 1 in Thanh *et al.* [30] is adapted to prove $R_\mu$ selected correctly with desired probability. This ensures the exact marginal distribution of $X^{-r}$ and respectively, $X^r$.

## 4 Numerical examples

We report in this section the applicability and efficiency of our proposed algorithm in two biological examples: the gene expression model and the gemcitabine model. All the algorithms were implemented in Java and are made freely available at `https://anticorrelatedvariance-rssa.sourceforge.io`. The benchmark in this section is performed on a Intel i5-540M processor.

### *4.1 Gene expression model*

The gene expression model [45] describes an important regulatory process in which the genetic information is transcribed and then translated into its encoded protein. It composes‘of two steps


© The Institution of Engineering and Technology 2015

Algorithm 2: Anticorrelated Variance RSSA

initialize state $X^r = X^{-r} = x_0$ and time $t^r = t^{-r} = 0$
build species-reaction (SR) dependency graph
for each species $S_i$ with $i = 1 \dots n$ define a bound $[\underline{X_i}, \overline{X_i}]$ such that $\underline{X_i} \leq X_i^r, X_i^{-r} \leq \overline{X_i}$
compute propensity bounds $\underline{a_j}$ and $\overline{a_j}$ for reaction $R_j$ with $j = 1, \dots, m$
set $\overline{a_0} = \sum_{j=1}^{m} \overline{a_j}$
**repeat**
  set UpdateSpeciesSet $= \emptyset$
  generate three random numbers $r_1, r_2$ and $r_3 \sim U(0,1)$
  **for** (each trajectory $k \in \{r, -r\}$) **do**
    **if** $(k == -r)$ **then**
      set $r_1 = 1 - r_1$, $r_2 = 1 - r_2$ and $r_3 = 1 - r_3$
    **end if**
    set $\tau = (1/r_1)\ln(1/\overline{a_0})$
    update time $t^k = t^k + \tau$
    select minimum reaction index $\mu$ s.t. $\sum_{j=1}^{\mu} \overline{a_j} > r_2 \overline{a_0}$
    set accepted = **false**
    **if** $(r_3 \leq \underline{a_\mu}/\overline{a_\mu})$ **then**
      set accepted = **true**
    **else**
      compute $a_\mu(X^k)$
      **if** $(r_3 \leq a_\mu(X^k)/\overline{a_\mu})$ **then**
        set accepted = **true**
      **end if**
    **end if**
    **if** (accepted = **true**) **then**
      update $X^k = X^k + v_\mu$
      **for all** (species $S_i$ where $X_i^k \notin [\underline{X_i}, \overline{X_i}]$) **do**
        set $UpdateSpeciesSet = \overline{UpdateSpeciesSet} \cup \{S_i\}$
      **end for**
    **end if**
  **end for**
  **for all** (species $S_i \in$ UpdateSpeciesSet) **do**
    define a new $[\underline{X_i}, \overline{X_i}]$ such that $\underline{X_i} \leq X_i^r, X_i^{-r} \leq \overline{X_i}$
    **for all** $(R_j \in \overline{\text{ReactionsAffectedBy}}(S_i))$ **do**
      compute propensity bounds $\overline{a_j}$ and $\underline{a_j}$
      update total upper bound sum $\overline{a_0}$
    **end for**
  **end for**
**until** ($t^k \geq T_{max}$ for all trajectories $k \in \{r, -r\}$)

that are transcription and translation. Transcription process starts when the RNA-polymerase (RNAP) binds to the promoter region of the gene and activate it to produce RNA. During the translation, RNA molecules bind to ribosomes and translate to produce the corresponding protein.

Table 1 describes the two steps of the gene expression model. It consists of 5 species and 8 reactions. Protein $P$ is encoded by its gene $G$. The RNA of the transcription is denoted by $M$. The transcription was modeled by reaction $R_1$ where gene $G$ transcribes to $M$, which is then translated into protein $P$ by reaction $R_2$ or degraded by reaction $R_3$. A protein $P$ can interact with other protein to form a reversible dimer $P_2$, represented by reactions $R_5$ and $R_6$. The degradation of the protein is expressed by reaction $R_4$. The dimer $P_2$ can bind to gene $G$ to enhance the activation of the gene which modeled by reactions $R_7$ and $R_8$. The initial condition of the gene expression model is: $\#G = 1,000$, and $\#M = \#P = \#P_2 = \#P_2G = 0$. The stochastic rates for reactions are: $c_1 = 0.09$, $c_2 = 0.05$, $c_3 = 0.001$, $c_4 = 0.0009$, $c_5 = 0.00001$, $c_6 = 0.0005$, $c_7 = 0.005$ and $c_8 = 0.9$.

Table 2 gives the estimated population of three species $P$, $P_2$ and $G$ at time $t = 200$. To compare the estimation of the population of species by algorithms, we fix the total number of simulations

**Table 1** Gene expression model

| Reaction | |
|---|---|
| $R_1$: $G \rightarrow G + M$ | $R_2$: $M \rightarrow M + P$ |
| $R_3$: $M \rightarrow \emptyset$ | $R_4$: $P \rightarrow \emptyset$ |
| $R_5$: $2P \rightarrow P_2$ | $R_6$: $P_2 \rightarrow 2P$ |
| $R_7$: $P_2 + G \rightarrow P_2G$ | $R_8$: $P_2G \rightarrow P_2 + G$ |

to 1000. For SSA, 1000 independent simulation runs are performed. For AV-SSA and AV-RSSA, we performed 500 simulation runs in which each run simulation two correlated realizations. In addition, the fluctuation interval of species in AV-RSSA is defined by $\pm 10\%$ around the minimum and maximum of states. The table shows that the standard deviation of the estimation by anticorrelated variance reduction has been reduced. Specifically, the standard deviation in estimation species $P_2$ by AV-SSA and AV-RSSA is reduced by $35\%$. In Figure 2, we plot the standard deviation in the estimation of the population of species at different time points. The similar reduction in the standard deviation by AV-SSA and AV-RSSA is also observed during time.

Figure 3 shows the performance of algorithms in simulating the gene expression model. For each simulation of this model, the average number of reaction firings is $3.43 \times 10^6$. AV-SSA saves $6.86 \times 10^6$ random numbers in comparison with SSA. By saving these random numbers, AV-SSA is 1.2 times faster than SSA The performance of AV-RSSA is the best. In simulating this model, AV-RSSA only performs 3664 propensity updates (approximately equal to $0.1\%$ in comparison with SSA). The huge update cost reduction make AV-RSSA be 1.8 and 1.5 times faster than SSA and AV-SSA, respectively. In Figure 4, we plot the total simulation time of algorithms with different values of standard deviation of G. We apply the sequential estimation [46] to simulate the model until the estimated standard deviation of species G is less than a prescribed value. We note that in the figure we did not show the CPU time of SSA with prescribed standard deviation of $G$ less than 3 because SSA could not finish in reasonable time in comparison with other algorithms.

### *4.2 Gemcitabine model*

Gemcitabine (2,2-difluorodeoxycytidine, dFdC) is an anti-cancer chemotherapy drug. It has been used to treat different types of cancer including non-small-cell lung cancer, pancreatic cancer, bladder cancer, and breast cancer [47]. Gemcitabine produces clinical effects by incorporating its triphosphate metabolite (dFdC-TP) into DNA leading to the inhibition and blocking of DNA synthesis. It also shows side effects and growing resistance. The gemcitabine model [48] has been developed to understand the mechanisms of resistance to gemcitabine efficacy. The model details mechanisms for the race between gemcitabine and natural nucleoside triphosphate dCTP for DNA incorporation. It also includes the mechanisms of resistance by considering the role of ribonucleotide reductase (RR), deoxycytidine kinase (dCK) and human equilibrative nucleoside transporter1 (hENT1).

We use Gemcitabine model developed by Kahramanoğullari *et al.* [48]. The model consists of 22 species and 29 reactions which are listed in Table 3. The reactions $R_1$ - $R_4$ abstract the transportation of dFdC and 2,2-difluorodeoxyuridine (dFdU) into the cell. The reactions $R_5$ - $R_{10}$ and $R_{11}$ - $R_{16}$, respectively, model the transformation of dFdC and dFdU into these corresponding active metabolites. dFdC is phosphorylated by deoxycytidine kinase (dCK) to monophosphate dFdC-MP and then phosphorylated to form its active metabolites dFdC-DP and dFdC-TP, respectively. Similarly, the corresponding phosphorylated forms of dFdU are monophosphate dFdU-MP, diphosphate dFdU-DP and triphosphate dFdU-TP. Reactions $R_{17}$ - $R_{18}$ model the deamination of dFdC into dFdU by cytidine deaminase (CDA). The incorporation of dFdC-TP and dFdU-TP into DNA to inhibit the DNA synthesis and to block cell


© The Institution of Engineering and Technology 2015

**Table 2** Estimated population of species in Gene expression model at time $t = 200$ by 1000 simulation runs.

| Algorithm | $G$ | $P$ | $P_2$ |
|---|---|---|---|
| SSA | $17.70 \pm 4.15$ | $11281.44 \pm 96.78$ | $100764.99 \pm 583.72$ |
| AV-SSA | $17.88 \pm 3.05$ | $11276.96 \pm 67.81$ | $100772.11 \pm 395.71$ |
| AV-RSSA | $17.73 \pm 3.08$ | $11276.43 \pm 70.79$ | $100737.50 \pm 378.16$ |

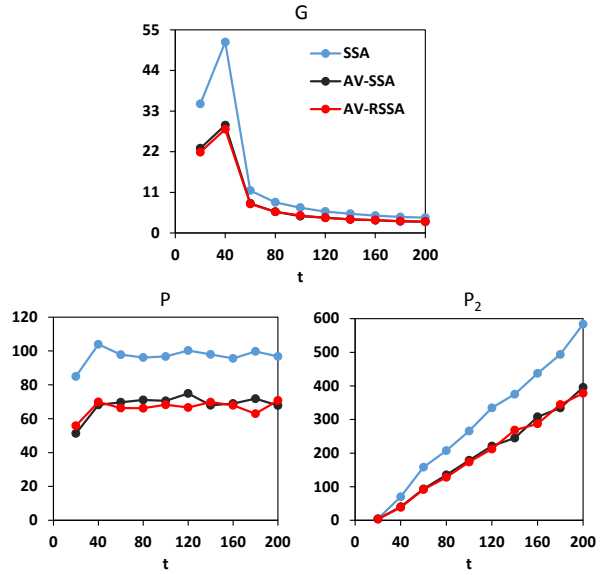


**Fig. 2**: Standard deviation of population of species in Gene expression model by 1000 simulation runs.

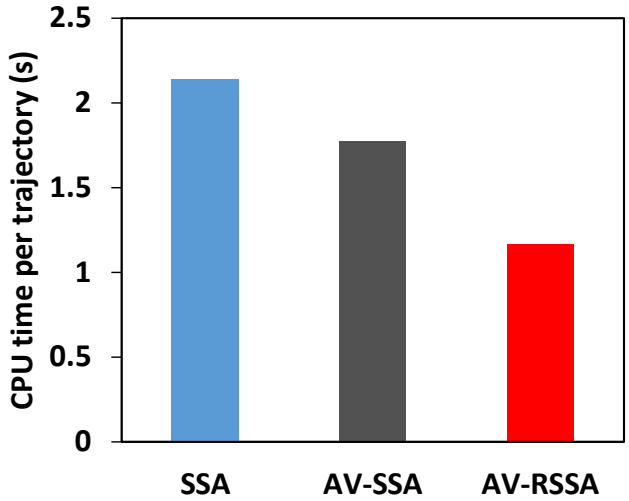


**Fig. 3**: CPU time of algorithms in simulating a trajectory of the Gene expression model.

proliferation in the early DNA synthesis chain is modelled by reactions $R_{19}$ - $R_{20}$. The competition cascade that results in the incorporation of dCTP into DNA are modelled by the reactions $R_{21}$ - $R_{24}$ model. Finally, the inhibitory mechanism of is modeled by reactions $R_{25}$ - $R_{29}$. We refer to Kahramanoğullari *et al.* [48] for the details of the Gemcitabine model and the kinetics parameters. The initial condition used in our simulation is $\#dCK = 1000$, $\#RR = 1000$, $\#dCMPD = 1000$, $\#CDP = 2000$, $\#dFdC$-out$= 100000$.

In Figure 5, we plot the standard deviation in the estimation of the population of species by the independent SSA and the anticorrelated variance approach. For this experiment, the simulation is performed until the time $t = 24$. The total number of simulation runs is fixed

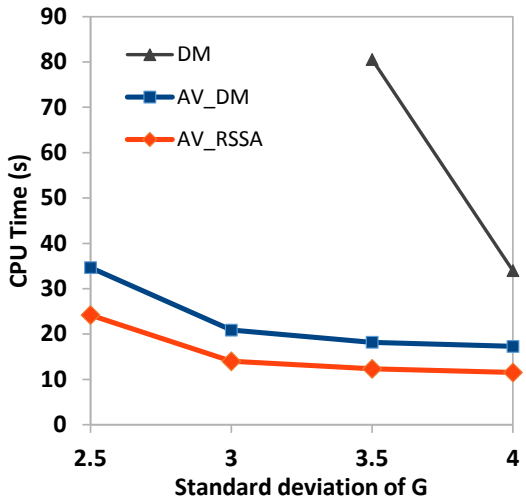


**Fig. 4**: CPU time of algorithms in simulating Gene expression model with prescribed standard deviation of G.

1000. The figure shows that the standard deviation of the estimation by anticorrelated variance reduction is reduced varying from 20% to 50%.

Figure 6 depicts total CPU time of algorithm in simulating the Gemcitabine model. For this experiment, there are, in average, $1.25 \times 10^7$ reaction firings. For this model, the reduction in the number of random numbers by AV-SSA, however, does not have a large effect in its performance in comparison with independent SSA. In contrast, the significant reduction in propensity computations makes AV-RSSA to be the best. Specifically, AV-RSSA performs about $2.28^4$ propensity updates. Thus, it is 2.2 and 1.9 times faster than SSA and AV-SSA, respectively.

### *4.3 FcεRI signaling pathway*

The high-affinity IgE receptor, known as FcεRI, is a high-affinity receptor for the antigen-specific immunoglobulin E (IgE). The receptor is a tetramer consisting of three subunits: an $\alpha$ chain (FcεRI$\alpha$), a $\beta$ chain (FcεI$\beta$), and two disulfide bridge connected $\gamma$ chains (FcεRI$\gamma$). The $\alpha$ chain serves as the binding site for IgE, while the others involve in initiating and amplifying the downstream signaling. The crosslinking of the IgE-antigen complex and the aggregation of the FcεRI leads to degranulation and release of allergic mediators from the immune system [49]. The FcεRI signaling pathway due to its role controlling allergic reactions has been studied extensively [50].

We use the FcεRI model developed by Liu *et al.* [51], which is developed to analyze the mechanisms of Syk phosphorylation in the FcεRI signaling pathway. The model contains 380 species and 3862. We refer to the original work [51] for the details of the model and kinetics parameters.

Table 4 gives the estimated population of some selected species and Figure 7 depicts the performance of algorithms. The results are obtained from 1000 simulating runs of the model with the ending time $t = 0.5$. We have remarks about the results of algorithms in simulating this model. First, the anticorrelated variance technique can help to reduce the variance of the estimation. Second, the


© The Institution of Engineering and Technology 2015

**Table 3** Gemcitabine model

| Reaction | |
|---|---|
| $R_1$: dFdC-out $\rightarrow$ dFdC | $R_2$: dFdC $\rightarrow$ dFdC-out |
| $R_3$: dFdC-out $\rightarrow$ dFdU | $R_4$: dFdU $\rightarrow$ dFdU-out |
| $R_5$: dFdC + dCK $\rightarrow$ dFdC-MP + dCK | $R_6$: dFdC-MP $\rightarrow$ dFdC |
| $R_7$: dFdC-MP $\rightarrow$ dFdC-DP | $R_8$: dFdC-DP $\rightarrow$ dFdC-MP |
| $R_9$: dFdC-DP $\rightarrow$ dFdC-TP | $R_{10}$: dFdC-TP $\rightarrow$ dFdC-DP |
| $R_{11}$: dFdU + dCK $\rightarrow$ dFdU-MP + dCK | $R_{12}$: dFdU-MP $\rightarrow$ dFdU |
| $R_{13}$: dFdU-MP $\rightarrow$ dFdU-DP | $R_{14}$: dFdU-DP $\rightarrow$ dFdU-MP |
| $R_{15}$: dFdU-DP $\rightarrow$ dFdU-TP | $R_{16}$: dFdU-TP $\rightarrow$ dFdU-DP |
| $R_{17}$: dFdC $\rightarrow$ dFdU | $R_{18}$: dFdC-MP + dCMPD $\rightarrow$ dFdU-MP + dCMPD |
| $R_{19}$: dFdC-TP $\rightarrow$ dFdC-TP-DNA | $R_{20}$: dFdU-TP $\rightarrow$ dFdU-TP-DNA |
| $R_{21}$: $\emptyset \rightarrow$ CDP | $R_{22}$: CDP + RR $\rightarrow$ dCDP + RR |
| $R_{23}$: dCDP $\rightarrow$ dCTP | $R_{24}$: dCTP $\rightarrow$ CTP-DNA |
| $R_{25}$: dCTP + dCK $\rightarrow$ dCTP-dCK | $R_{26}$: dCTP-dCK $\rightarrow$ dCTP + dCK |
| $R_{27}$: dFdC-DP + RR $\rightarrow$ dFdC-DP-RR | $R_{28}$: dFdC-TP + dCMPD $\rightarrow$ dFdC-TP-dCMPD |
| | $R_{29}$: dFdC-TP-dCMPD $\rightarrow$ dFdC-TP + dCMPD |

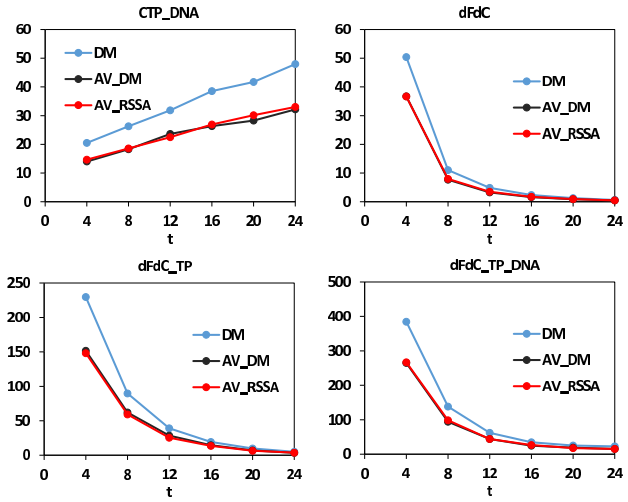


**Fig. 5**: Standard deviation of population of species in Gemcitabine model by 1000 simulation runs.

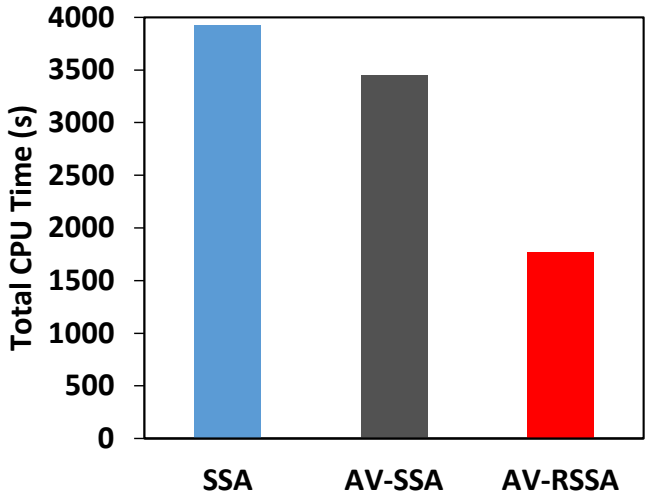


**Fig. 6**: CPU time of algorithms in simulating Gemcitabine model until time $t = 24$ with 1000 simulation runs.

performance of AV-SSA through the direct application of the anti-correlated variance technique by inverse transformation is similar to independent SSA. Third, the performance of AV-RSSA, which uses rejection-based approach to imposes the correlation, is significantly better than AV-SSA. Specifically, AV-RSSA is about 5 times faster than AV-SSA. Such the performance gain achieved by AV-RSSA comes from the reduction in the propensity update cost where in order to form a simulation trajectory AV-RSSA performs only 229 updates, while AV-SSA has to perform $2.8 \times 10^5$ updates.

## 5 Conclusions

We presented in this paper a new algorithm that improves the variance of the stochastic simulation by employing the anticorrelated variance technique. We employ the propensity bounds of reactions, introduced recently in RSSA, to correlate the trajectories during the simulation. We synchronize the realizations by decomposing the complex acceptance-rejection selection in the original RSSA procedure into single steps. Our algorithm can produce exact simulation trajectories with substantial correlation between these trajectories, hence reducing the variance of the estimator, while still achieving better performance in comparing with the standard simulation approaches. In this paper we only investigated the case of two correlated realizations, but our approach is possible to extends for many correlated realizations using hybrid strategy [39] based on the simultaneous rejection-based simulation strategy [31]. Our approach can also be applied to speed up the estimation by the multi-level approach [52], which is promising for further improvements.


## Acknowledgment

This work has been partially done when VHT was at the Microsoft Research - University of Trento Centre for Computational and Systems Biology (COSBI), Italy. The research of VHT supported by Academy of Finland grant 311639, "Algorithmic Designs for Biomolecular Nanostructures (ALBION)".


## 6 References


1 Juan M. Pedraza and Alexander van Oudenaarden. Noise propagation in gene networks. *Science*, 307(5717):1965–1969, 2005.

2 Harley H. McAdams and Adam Arkin. It's a noisy business! genetic regulation at the nanomolar scale. *Trends in Genetics*, 15(2), 1999.

3 Harley H. McAdams and Adam Arkin. Stochastic mechanisms in gene expression. In *PNAS*, 94(3):814–819, 1997.




© The Institution of Engineering and Technology 2015

**Table 4** Estimated population of selected species in FcεRI model at time $t = 0.5$ by $1000$ simulation runs.

| Algorithm | $S10$ | $S11$ | $S12$ |
|---|---|---|---|
| SSA | $132.28 \pm 11.60$ | $15.35 \pm 3.84$ | $18.42 \pm 4.26$ |
| AV-SSA | $132.02 \pm 8.19$ | $15.21 \pm 2.60$ | $18.17 \pm 3.13$ |
| AV-RSSA | $131.76 \pm 7.40$ | $15.43 \pm 2.72$ | $18.26 \pm 2.85$ |

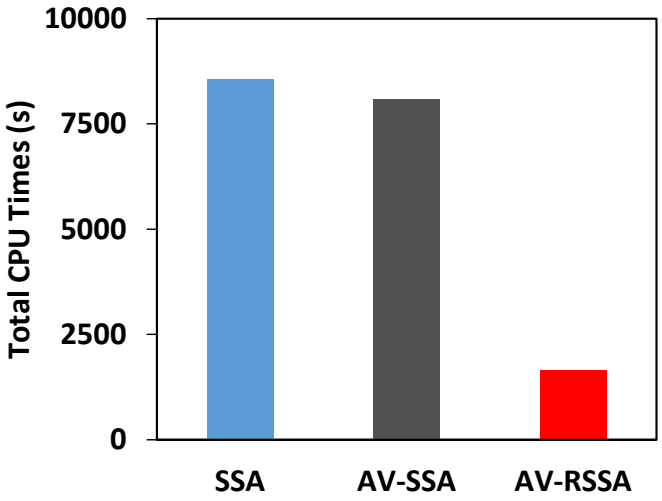


**Fig. 7**: CPU time of algorithms in simulating FcεRI model until time $t = 0.5$ with 1000 simulation runs.


4 Adam Arkin, John Ross, and Harley H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells. *Genetics*, 149(4):1633–1648, 1998.

5 José M. G. Vilar, Hao Y. Kueh, Naama Barkai, and Stanislas Leibler. J. Vilar, H. Kueh, N. Barkai, and S. Leibler. Mechanisms of noise-resistance in genetic oscillators. In *PNAS*, 99(9):5988–5992, 2002.

6 Ertugrul M. Ozbudak, Mukund Thattai, Iren Kurtser, Alan D. Grossman, and Alexander van Oudenaarden. Regulation of noise in the expression of a single gene. *Nature Genetics*, 31:69–73, 2002.

7 Jonathan M. Raser and Erin K. O'Shea. Noise in gene expression: Origins, consequences, and control. *Science*, 309:2010–2013, 2005.

8 Luca Marchetti, Corrado Priami, and Vo H. Thanh. *Simulation Algorithms for Computational Systems Biology*. Springer, 2017.

9 Daniel Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comp. Phys.*, 22(4):403–434, 1976.

10 Daniel Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.

11 Michael Gibson and Jehoshua Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A*, 104(9):1876–1889, 2000.

12 Yang Cao, Hong Li, and Linda Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.*, 121(9):4059, 2004.

13 Vo H. Thanh, Roberto Zunino, and Corrado Priami. Efficient stochastic simulation of biochemical reactions with noise and delays. *J. Chem. Phys.*, 146(8):084107, 2017.

14 Tim Schulze. Efficient kinetic monte carlo simulation. *J. Comp. Phys.*, 227(4):2455–2462, 2008.

15 Alexander Slepoy, Aidan P. Thompson, and Steven J. Plimpton. A constant-time kinetic monte carlo algorithm for simulation of large biochemical reaction networks. *J. Chem. Phys.*, 128(20):205101, 2008.

16 Vo H. Thanh and Roberto Zunino. Tree-based search for stochastic simulation algorithm. In *Proc. of ACM-SAC*, pages 1415–1416, 2012.

17 Vo H. Thanh and Roberto Zunino. Adaptive tree-based search for stochastic simulation algorithm. *Int. J. of Computational Biology and Drug Design*, 7(4):341–57, 2014.

18 James Blue, Isabel Beichl, and Francis Sullivan. Faster monte carlo simulations. *Phys. Rev. E*, 51(2):867–868, 1995.

19 Rajesh Ramaswamy, Nélido González-Segredo, and Ivo F. Sbalzarini. A new class of highly efficient exact stochastic simulation algorithms for chemical reaction networks. *J. Chem. Phys.*, 130(24):244104, 2009.

20 Vo Hong Thanh. A Critical Comparison of Rejection-Based Algorithms for Simulation of Large Biochemical Reaction Networks. *Bulletin of Mathematical Biology*, 1–21, 2018. https://doi.org/10.1007/s11538-018-0462-y.

21 Sean Mauch and Mark Stalzer. Efficient formulations for exact stochastic simulation of chemical systems. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, 8(1):27–35, 2011.

22 Hong Li and Linda Petzold. Efficient parallelization of the stochastic simulation algorithm for chemically reacting systems on the graphics processing unit. *Intl. J. of High Performance Computing Applications*, 24(2):107–116, 2010.

23 Vo H. Thanh and Roberto Zunino. Parallel stochastic simulation of biochemical reaction systems on multi-core processors. In *Proc. of CSSim*, pages 162–170, 2011.

24 Werner Sandmann. Discrete-time stochastic modeling and simulation of biochemical networks. *Comput. Biol. Chem.*, 32(4):292, 2008.

25 Daniel Gillespie. Approximate accelerated stochastic simulation of chemically reacting. *J. Chem. Phys.*, 115:1716–1733, 2001.

26 Yang Cao, Daniel Gillespie, and Linda Petzold. Efficient step size selection for the tau-leaping simulation method. *J. Chem. Phys.*, 124(4):44109, 2006.

27 Anne Auger, Philippe Chatelain, and Petros Koumoutsakos. R-leaping: Accelerating the stochastic simulation algorithm by reaction leaps. *J. Chem. Phys.*, 125(8):84103, 2006.

28 Luca Marchetti, Corrado Priami, and Vo H. Thanh. HRSSA – Efficient hybrid stochastic simulation for spatially homogeneous biochemical reaction networks. *Journal of Computational Physics*, 317:301–317, 2016.

29 Vo H. Thanh. *On Efficient Algorithms for Stochastic Simulation of Biochemical Reaction Systems*. PhD thesis, University of Trento, Italy. urlhttp://eprints-phd.biblio.unitn.it/1070/, 2013.

30 Vo H. Thanh, Corrado Priami, and Roberto Zunino. Efficient rejection-based simulation of biochemical reactions with stochastic noise and delays. *J. Chem. Phys.*, 141(13), 2014.

31 Vo H. Thanh, Roberto Zunino, and Corrado Priami. On the rejection-based algorithm for simulation and analysis of large-scale reaction networks. *The Journal of Chemical Physics*, 142(24):244106, 2015.

32 Vo H. Thanh, Roberto Zunino, and Corrado Priami. Efficient constant-time complexity algorithm for stochastic simulation of large reaction networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(3):657–667, 2017.

33 Vo H. Thanh, Corrado Priami, and Roberto Zunino. Accelerating rejection-based simulation of biochemical reactions with bounded acceptance probability. *The Journal of Chemical Physics*, 144(22):224108, 2016.

34 Vo H. Thanh and Corrado Priami. Simulation of biochemical reactions with time-dependent rates by the rejection-based algorithm. *The Journal of Chemical Physics*, 143(5):054104, 2015.

35 Vo H. Thanh. Stochastic simulation of biochemical reactions with partial-propensity and rejection-based approaches. *Mathematical Biosciences*, 292:67–75, 2017.

36 Søren Asmussen and Peter W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer, 2007.

37 Christian Robert and George Casella. *Monte Carlo Statistical Methods, 2nd edition*. Springer, 2004.

38 James E. Gentle. *Random Number Generation and Monte Carlo Methods, 2nd Edition*. Springer, 2004.

39 Peter A. Maginnis, Matthew West, and Geir E. Dullerud. Application of variance reduction techniques for tau-leaping systems to particle filters. In *Proc. of The 51st IEEE Conference on Decision and Control*, pages 21–37, 2012.

40 Peter A. Maginnis, Matthew West, and Geir E. Dullerud. Anticorrelated discrete-time stochastic simulation. In *Proc. of The 52st IEEE Conference on Decision and Control*, pages 618–623, 2013.

41 Peter A. Maginnis, Matthew West, and Geir E. Dullerud. Exact simulation of continuous time markov jump processes with anticorrelated variance reduced monte carlo estimation. In *Proc. of The 53st IEEE Conference on Decision and Control*, pages 3401–3407, 2014.

42 Vo H. Thanh, Roberto Zunino, and Corrado Priami. Efficient finite difference method for computing sensitivities of biochemical reactions. (available at: `https://arxiv.org/pdf/1707.09193.pdf`), 2017.

43 David F. Anderson and Chaojie Yuan. Low variance couplings for stochastic models of intracellular processes with time-dependent rate functions. (available at: `https://arxiv.org/pdf/1708.01813.pdf`), 2017.

44 Bruce Schmelser and Voratas Kachltvlchyanukul. Correlation induction without the inverse transformation. In *Proc. of the Winter Simulation Conference*, pages 266–274, 1986.

45 Darren J. Wilkinson. *Stochastic Modelling for Systems Biology*. CRC Press, 2nd edition, 2011.

46 Werner Sandmann. Sequential estimation for prescribed statistical accuracy in stochastic simulation of biological systems. *Mathematical biosciences*, 221:43–53, 2009.

47 Stephan A. Veltkamp, Jos H. Beijnen, and Jan H.M. Schellens. Prolonged versus standard gemcitabine infusion: Translation of molecular pharmacology to new treatment strategy. *The Oncologist*, 13(3):261–276, 2008.

48 Ozan Kahramanogullari, Gianluca Fantaccini, Paola Lecca, Daniele Morpurgo, and Corrado Priami. Algorithmic modeling quantifies the complementary contribution of metabolic inhibitions to gemcitabine efficacy. *World Journal of Gastrointestinal Oncology*, 7(12), 2012.




© The Institution of Engineering and Technology 2015

49 James R. Faeder *et al.* Investigation of early events in FcεRI-mediated signaling using a detailed mathematical model. *The Journal of Immunology*,170:3769–3781, 2003.

50 Lily A. Chylek and David A. Holowka and Barbara A. Baird and William S. Hlavacek. An interaction library for the FcεRI signaling network. *Frontiers in Immunology*, 5(172):1664–3224, 2014.

51 Yanli Liu *et al.* Single-Cell Measurements of IgE-Mediated FcεRI Signaling Using an Integrated Microfluidic Platform. *PLoS ONE*, 8(3):60159, 2013.

52 David F. Anderson and Desmond J. Higham. Multi-level Monte Carlo for continuous time Markov chains, with applications to biochemical kinetics. *SIAM Multiscale Modeling and Simulation*, 10(1):146-179, 2012.



© The Institution of Engineering and Technology 2015