A partial-propensity variant of the composition-rejection stochastic simulation algorithm for chemical reaction networks

Rajesh Ramaswamy[a] and Ivo F. Sbalzarini[b]
Institute of Theoretical Computer Science and Swiss Institute of Bioinformatics, ETH Zurich, CH-8092 Zürich, Switzerland

(Received 5 November 2009; accepted 4 January 2010; published online 22 January 2010)

We present the partial-propensity stochastic simulation algorithm with composition-rejection sampling (PSSA-CR). It is an exact formulation of the stochastic simulation algorithm (SSA) for well-stirred systems of coupled chemical reactions. The new formulation is a partial-propensity variant [R. Ramaswamy, N. González-Segredo, and I. F. Sbalzarini, J. Chem. Phys. **130**, 244104 (2009)] of the composition- rejection SSA [A. Slepoy, A. P. Thompson, and S. J. Plimpton, J. Chem. Phys. **128**, 205101 (2008)]. The computational cost of this new formulation is bounded by a constant for weakly coupled reaction networks, and it increases at most linearly with the number of chemical species for strongly coupled reaction networks. PSSA-CR thus combines the advantages of partial-propensity methods and the composition-rejection SSA, providing favorable scaling of the computational cost for all classes of reaction networks. *© 2010 American Institute of Physics*. [doi:10.1063/1.3297948]

I. INTRODUCTION

Stochastic chemical kinetics of a well-mixed system of coupled reactions can be described using the chemical master equation,[1–3] a Markov chain model. Numerical simulation of the chemical master equation is usually done using a kinetic Monte Carlo approach known as the stochastic simulation algorithm (SSA).[1–3] It is governed by the joint probability density function,

$$p(\tau,\mu|\mathbf{n}(t)) = (a\mathrm{e}^{-a\tau})(a_\mu/a), \qquad (1)$$

for two independent random variables: the time to the next reaction (τ) and the index of the next reaction (μ). The vector $\mathbf{n}(t)=(n_1,\dots,n_N)$ is the population of species at time t. Each entry n_i is the number of molecules of the respective species S_i, and N is the total number of species. The propensity of each reaction μ is defined as $a_\mu=c_\mu h_\mu$, where c_μ is the specific probability rate and $h_\mu=h_\mu(\mathbf{n})$ is the reaction degeneracy, which is the number of possible combinations of reactant molecules in reaction μ, given the population $\mathbf{n}$. The reaction propensity is such that $a_\mu \mathrm{d}t$ is the probability that reaction μ happens in the next infinitesimal time interval dt. The total propensity is $a=\Sigma_{\mu=1}^{M} a_\mu$, where M is the total number of reactions. In this paper, we restrict ourselves to exact SSAs that sample τ and μ from Eq. (1). We do not consider approximate methods that sample from an approximation of Eq. (1) in order to improve computational efficiency.

Chemical reaction networks can be represented by their dependency graph. In this graph, each node represents a reaction and an arrow (directed edge) is drawn from node p to node q if the firing of reaction p affects the propensity of reaction q.[4] The out-degree of node p is defined as the number of arrows leaving that node. Using this representation, we distinguish two coupling classes of chemical reaction networks: *weakly coupled* and *strongly coupled*. In weakly coupled reaction networks, the maximum out-degree of the dependency graph (i.e., the degree of coupling of the network) is constant or bounded with increasing network size. Strongly coupled reaction networks have a degree of coupling that increases unboundedly with system size. The scaling of the computational cost (here formalized using the Bachmann–Landau "big-O" notation) of SSAs is determined by the coupling class of the network. For weakly coupled reaction networks, the computational cost (CPU time) of exact SSA formulations has been reduced to $O(\log M)$ in the next reaction method (NRM)[4] and to $O(1)$ in SSA-CR,[5] the latter under the assumption that the ratio of maximum to minimum nonzero reaction propensity is bounded. For strongly coupled networks, however, the computational cost of these SSA formulations remains $O(M)$, as for Gillespie's original direct method.[6] We recently introduced novel exact SSA formulations that are based on partial propensities.[6] These formulations, called PDM and SPDM, have a computational cost of $O(N)$, irrespective of the coupling class of the network.[6] This makes them particularly interesting for strongly coupled networks.

In practice, and especially for networks of fixed size, it is often difficult to determine which coupling class a reaction network belongs to. This is because the coupling class is defined as a function of system size. For fixed-size systems, however, only a single point of that function is known, requiring additional knowledge to determine the coupling class. There is thus a need for an exact SSA that combines the favorable scaling of the computational cost of SSA-CR for weakly coupled networks and of PDM for strongly coupled ones. In this paper, we use the concept of partial propensities[6] to construct a partial-propensity variant of

[a] Electronic mail: rajeshr@ethz.ch.
[b] Electronic mail: ivos@ethz.ch.

0021-9606/2010/132(4)/044102/6/$30.00

SSA-CR, called PSSA-CR. We show that PSSA-CR has a computational cost of $O(1)$ for weakly coupled networks and $O(N)$ for strongly coupled networks, thus combining the advantages of PDM and SSA-CR. To our knowledge, this formulation has the best scaling of the computational cost on any class of reaction networks. As in PDM, we restrict ourselves to chemical reaction networks composed of elementary reactions. Any nonelementary reactions can be equivalently decomposed into elementary ones at the expense of a larger network size.[3,7] Also, as in SSA-CR, the $O(1)$ scaling for weakly coupled networks is achieved under the assumption that the ratio of maximum to minimum nonzero reaction propensity is bounded.

II. THE PARTIAL-PROPENSITY SSA WITH COMPOSITION-REJECTION SAMPLING (PSSA-CR)

The partial-propensity SSA with composition-rejection sampling (PSSA-CR) is based on the idea of factorizing the reaction propensities as described below,[6] grouping and binning them, and using composition- rejection sampling[8,5] to determine the index of the next reaction.

A. Prerequisites for PSSA-CR

In this section, we recall the basic ingredients for PSSA-CR: partial propensities, partial-propensity SSAs, and composition-rejection sampling. For a more detailed treatment of these concepts, the reader is referred to the corresponding original publications.[6,8,5]

1. Partial propensities

The partial propensity of a reaction with respect to one of its reactants is defined as the propensity per number of molecules of that reactant.[6] For example, the partial propensity $\pi_\mu^{(i)}$ of reaction μ with respect to (perhaps the only) reactant S_i is a_μ/n_i, where a_μ is the propensity of reaction μ and n_i is the number of molecules of S_i. The partial propensities of the three elementary reaction types are:

- Bimolecular reactions ($S_i+S_j \rightarrow$ products): $a_\mu = n_i n_j c_\mu$ and $\pi_\mu^{(i)} = n_j c_\mu$, $\pi_\mu^{(j)} = n_i c_\mu$. If both reactants are of the same species, i.e., $S_i = S_j$, only one partial propensity exists, $\pi_\mu^{(i)} = (1/2)(n_i - 1)c_\mu$, because the reaction degeneracy is $(1/2)n_i(n_i - 1)$.
- Unimolecular reactions ($S_i \rightarrow$ products): $a_\mu = n_i c_\mu$ and $\pi_\mu^{(i)} = c_\mu$.
- Source reactions ($\varnothing \rightarrow$ products): $a_\mu = c_\mu$ and $\pi_\mu^{(0)} = c_\mu$.

We consider only these elementary reaction types, since any reaction with three or more reactants can be treated by decomposing it into a combination of elementary reactions.[3,9,7]

2. Partial-propensity SSAs

Partial propensity methods group the partial propensities of all reactions according to the index of the factored-out reactant.[6] This results in at most $N+1$ groups of size $O(N)$. Every reaction and its corresponding partial propensity are then identifiable by two indices: a *group index* and an *element index*. The group index identifies the partial-propensity group to which a reaction belongs and the element index identifies the position of the reaction inside that group. Determining the index of the next reaction is done by first sampling its group index and then its element index.

TABLE I. Overview of PSSA-CR.

```
1. Initialize the data structures (see Appendix B). Set time $t \leftarrow 0$.
2. While $t < t_f$, where $t_f$ is the final simulation time, repeat:
   2.1. Sample the group index $I$ using composition-rejection sampling.
   2.2. Sample the element index $J$ using composition-rejection sampling.
   2.3. Read the index of the reaction identified by the group index $I$ and
        the element index $J$ from the look-up table.
   2.4. Compute the time to the next reaction $\tau \leftarrow a^{-1} \ln(r^{-1})$, where $a$ is
        the total propensity of all reactions and $r$ a uniformly distributed
        random number in $[0,1)$.
   2.5. Update the population of species and the partial propensity
        structure. Update the bin assignments of changed partial
        propensities.
   2.6. Increment time: $t \leftarrow t + \tau$.
3. Stop.
```

After the selected reaction has fired and the populations of the involved species have changed, the partial propensities are updated using a dependency graph over species. This dependency graph points to all partial propensities that need to be updated due to the change in population. Since any partial propensity is a function of the population of at most one species, the number of updates is at most $O(N)$. In weakly coupled reaction networks, the number of updates is $O(1)$, since the degree of coupling is bounded by a constant (by definition of a weakly coupled network).

3. Composition-rejection sampling

Composition-rejection sampling[8] is a way of sampling realizations of a random variable x according to a given probability density function. In SSAs, sampling the index of the next reaction involves a discrete probability density function $p(x_i)$, $i = 1, \ldots, k$. The sampling process starts by binning the $p(x_i)$'s according to their value and then proceeds in two steps. The composition step is used to identify the bin by linear search, and the rejection step is used to identify the $p(x_i)$ inside that bin.

In SSA-CR, composition-rejection sampling over propensities is used to sample the index of the next reaction as governed by the probability density function in Eq. (1).

B. Detailed description of the PSSA-CR algorithm

PSSA-CR uses a composition-rejection sampling strategy over partial propensities in order to sample the index of the next reaction. Since every reaction in a partial propensity method is identified by its group index and its element index, we apply two composition-rejection steps: one to sample the group index and one to sample the element index. Table I gives an overview of PSSA-CR. The individual steps are described in detail below.

The partial propensities are stored in a "partial propensity structure" $\mathbf{\Pi} = \{\mathbf{\Pi}_i\}_{i=0}^{N}$ as a one-dimensional array of one-dimensional arrays.[6] Each array $\mathbf{\Pi}_i$ contains the partial propensities of reactions belonging to group i, i.e., the partial propensities where n_i has been factored out.[6] The partial-propensity structure only needs to be constructed once, at the beginning of a simulation. This is done automatically as outlined in Appendix B. The reaction indices μ corresponding to a certain entry in $\mathbf{\Pi}$ are stored in a look-up table $\mathbf{L}=\{\mathbf{L}_i\}_{i=0}^{N}$. Each reaction μ is identified by its group index I and its element index J as $\mu = \mathrm{L}_{I,J}$. The "group-sum array" $\mathbf{\Lambda}$ stores the sums of the partial propensities in each group $\mathbf{\Pi}_i$, i.e., $\Lambda_i = \Sigma_j \Pi_{i,j}$. We also store the total propensity of each group in an array $\mathbf{\Sigma}$, computed as $\Sigma_i = n_i \Lambda_i$, $i=1,\ldots,N$, and $\Sigma_0 = \Lambda_0$.[6]

In PSSA-CR, the entries of $\mathbf{\Sigma}$ are then sorted into $G_\Sigma = \log_2(\Sigma_{\max}/\Sigma_{\min})+1$ bins such that bin b contains all Σ_i's with $2^{b-1}\Sigma_{\min} \le \Sigma_i < 2^b \Sigma_{\min}$. $\Sigma_{\min}$ and $\Sigma_{\max}$ are the smallest and largest values in $\mathbf{\Sigma}$ that can possibly occur during a simulation. They are determined as outlined below. The total propensity of each bin b, $\sigma_b^{(\Sigma)}$, is computed by summing up the Σ_i's in that bin. Similarly, the entries of each $\mathbf{\Pi}_i$ are sorted into $G_{\Pi_i} = \log_2(\Pi_{i,\max}/\Pi_{i,\min})+1$ bins with bin b containing all elements in $\mathbf{\Pi}_i$ such that $2^{b-1}\Pi_{i,\min} \le \Pi_{i,j} < 2^b \Pi_{i,\min}$. $\Pi_{i,\min}$ and $\Pi_{i,\max}$ are the smallest and largest values in $\mathbf{\Pi}_i$ that can possibly occur during a simulation. The total partial propensity of each bin b is stored in $\sigma_b^{(\Pi_i)}$. The $\Pi_{i,\min}$'s and $\Sigma_{\min}$ can always be computed *a priori*. $\Pi_{i,\min}$ is the minimum nonzero value in $\mathbf{\Pi}_i$ when all partial propensities are calculated with one molecule of each reactant. $\Sigma_{\min}$ is the minimum among all $n_i \Pi_{i,\min}$'s, where n_i is the population of species S_i used to calculate $\Pi_{i,\min}$. Estimating the $\Pi_{i,\max}$'s and $\Sigma_{\max}$ *a priori* may be possible by using prior knowledge about the chemical reaction network, such as physical constraints. In cases where the $\Pi_{i,\max}$'s and $\Sigma_{\max}$ cannot be estimated *a priori*, PSSA-CR dynamically updates the $\Pi_{i,\max}$'s and $\Sigma_{\max}$ over the course of the simulation. If this increases any G_{Π_i} or G_Σ, the corresponding data structures are dynamically enlarged.

We apply the composition-rejection sampling strategy[8,5] to obtain the group index I and the element index J of the next reaction μ. The group index I is sampled in two steps: (1) the composition step to find the bin b_I and (2) the rejection step to find Σ_I inside that bin. The composition step is done by linear search, thus

$$b_I = \min\left[b : r_1 a < \sum_{i=0}^{b} \sigma_i^{(\Sigma)} \right], \tag{2}$$

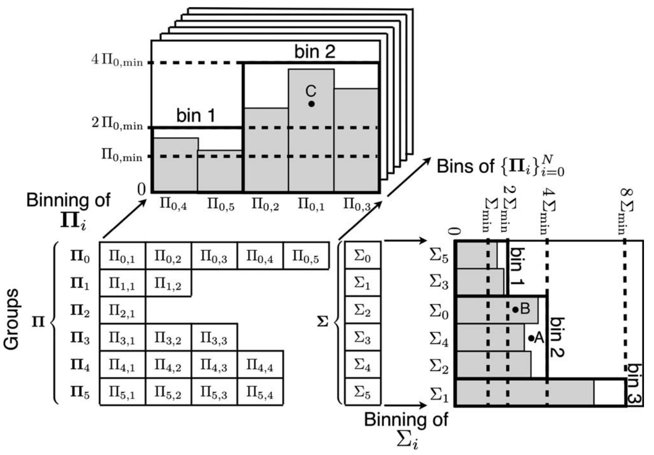

FIG. 1. Illustration of the two composition-rejection sampling steps used in PSSA-CR. The example shown is for a network with $N=5$ species and $M=19$ reactions. The partial propensities are grouped into six $(=N+1)$ groups in the "partial-propensity structure" $\{\mathbf{\Pi}\}_{i=0}^{N}$. The sum of propensities in group $\mathbf{\Pi}_i$ is stored in Σ_i. The elements of $\mathbf{\Sigma}$ and of each $\mathbf{\Pi}_i$ are sorted into dyadic bins. The shaded bars represent the values of the corresponding entries. The extent of each bin is shown by a bold rectangle. Due to the dyadic binning, the bars always cover at least 50% of the area of any bin's rectangle. In order to sample the index of the next reaction, two composition-rejection sampling steps are used: one for the group index I and another one for the element in index J in $\mathbf{\Pi}$. Points A, B, and C refer to the example given in the main text.

where a is the total propensity of all reactions in the network and r_1 is a uniform random number in [0, 1). The rejection step samples the group index I from the elements in bin b_I. For this step, we generate a uniformly distributed random number r_2 in $[0, 2^{b_I}\Sigma_{\min})$ and a uniformly distributed random integer r_3 between 1 and the number of elements in bin b_I. If the r_3th Σ_i in bin b_I is less than r_2, the index of that Σ_i is chosen as the group index I. If this inequality is not satisfied, the rejection step is repeated. This is illustrated in Fig. 1 for an example with six partial-propensity groups. Assume that, in this example, the composition step has selected bin $b_I=2$ as the one containing Σ_I. The rejection step then samples uniformly random points inside the rectangle defining this bin (bold rectangle). A sample is accepted if it falls inside one of the bars representing the Σ_i's. If the first sample (point A in Fig. 1 with $r_3=2$ and $r_2 > \Sigma_4$) is rejected, sampling is repeated until the point falls inside one of the bars (point B in Fig. 1 with $r_3=1$ and $r_2 < \Sigma_0$). This determines the group index of the next reaction ($I=0$ in the example in Fig. 1). By binning the Σ_i's as described, we ensure that the area covered by the Σ_i bars in any bin is at least 50% of the total area of the bin's bounding rectangle. The expected number of iterations of the rejection sampling is hence less than or equal to two.

In order to sample the element index J, the same composition-rejection procedure is also applied within the identified group I. The composition step again involves a linear search for the bin b_J containing the partial propensity of the next reaction, as

$$b_J = \min\left[b : r_4 \Lambda_I < \sum_{i=0}^{b} \sigma_i^{(\Pi_I)} \right], \tag{3}$$

where r_4 is a uniform random number in $[0,1)$. The rejection step as described above is subsequently used to find the element index J from a uniformly distributed random number r_5 in $[0, 2^{b_J}\Pi_{I,\min})$ and a uniformly distributed random integer r_6 between 1 and the number of elements in bin b_J. In the example in Fig. 1, the group index $I=0$ has been selected. Assume that the composition step for the element index J has selected bin $b_J=2$ in the group $\mathbf{\Pi}_0$. Rejection sampling in this bin is then repeated until a point inside any of the bars representing the partial propensities $\Pi_{0,j}$ is selected (point C in Fig. 1 with $r_6=2$ and $r_5 < \Pi_{0,1}$). This determines the element index of the next reaction ($J=1$ in the example in Fig. 1). The indices I and J of the next reaction are then translated to the reaction index μ using the look-up table, hence $\mu=\mathbf{L}_{I,J}$.

Once a reaction has been executed, $\mathbf{n}$, $\mathbf{\Pi}$, $\mathbf{\Lambda}$, and $\mathbf{\Sigma}$ are updated using the same update algorithm and data structures as in PDM:[6]

$\mathbf{U}^{(1)}$ is an array of M arrays, where the ith array contains the indices of all species involved in the ith reaction.

$\mathbf{U}^{(2)}$ is an array of M arrays containing the corresponding stoichiometry (the change in population of each species upon reaction) of the species stored in $\mathbf{U}^{(1)}$.

$\mathbf{U}^{(3)}$ is an array of N arrays, where the ith array contains the indices of all entries in $\mathbf{\Pi}$ that depend on n_i.

After each reaction, we use $\mathbf{U}^{(1)}$ to determine the indices of all species involved in this reaction. The stoichiometry is then looked up in $\mathbf{U}^{(2)}$ and the population $\mathbf{n}$ is updated accordingly. Subsequently, $\mathbf{U}^{(3)}$ is used to locate the affected entries in $\mathbf{\Pi}$ and recompute them. Since the partial propensities of unimolecular and source reactions are constant and need never be updated, $\mathbf{U}^{(3)}$ only contains the indices of the partial propensities of bimolecular reactions.

After updating the partial propensities, the bin memberships of all modified $\Pi_{i,j}$'s and Σ_i's need to be updated. This requires locating the bin assignment of any $\Pi_{i,j}$ and Σ_i in a one-step operation. We implement this by having every $\Pi_{i,j}$ and Σ_i store two additional integers: the bin membership and the location inside that bin. Depending on their new value, the changed $\Pi_{i,j}$'s and Σ_i's are kept inside the same bin or moved to a different bin. Then, the corresponding bin sums are updated by adding the total change. This can be done in $O(1)$ operations since the ordering of elements in a bin does not matter. Elements that are removed from a bin are simply replaced by the last element in that bin, which is then removed.

The computational cost of PSSA-CR is $O(G_\Sigma+\max\{G_{\Pi_0},\ldots,G_{\Pi_N}\}+N)$ for strongly coupled reaction networks and $O(G_\Sigma+\max\{G_{\Pi_0},\ldots,G_{\Pi_N}\})$ for weakly coupled ones (see Appendix A for proof). If the dynamic range of propensities is bounded over the time of a simulation, the computational cost on weakly coupled networks reduces to $O(1)$ (see Appendix A).

III. BENCHMARKS

We benchmark the computational performance of PSSA-CR on both a weakly coupled and a strongly coupled reaction network. We choose the cyclic chain model[10] and the colloidal aggregation model[11] as representative networks, respectively. We compare the performance of PSSA-CR with that of SDM, the sorting direct method,[12] and SPDM, the analogous sorting variant of PDM.[6]

All tested SSA formulations are implemented in C++ using the random number generator of the GSL library and compiled using the Intel C++ compiler version 11.1 with the O3 optimization flag. All timings are measured on a Linux 2.6 workstation with a 2.8 GHz quad-core Intel Xeon E5462 processor and 8 Gbyte of memory. For all test cases, we simulate the reaction network until 10^7 reactions have been executed and report the average CPU time Θ per reaction. All simulations are run without any *a priori* estimate of the $\Pi_{i,\max}$'s and $\Sigma_{\max}$. Instead, the $\Pi_{i,\max}$'s and $\Sigma_{\max}$ are constantly updated over the course of a simulation and the number of bins is dynamically increased when necessary.

A. A weakly coupled reaction network: Cyclic chain model

The cyclic chain model is given by the reaction network

$$\begin{aligned} &\mathrm{S}_i \xrightarrow{c_i} \mathrm{S}_{i+1} \quad i=1,\ldots,N-1, \\ &\mathrm{S}_N \xrightarrow{c_N} \mathrm{S}_1. \end{aligned} \tag{4}$$

For N chemical species, this network has $M=N$ reactions. The degree of coupling (maximum out-degree of the dependency graph) of this reaction network is 2, independent of system size.

At time $t=0$, we set all $n_i=1$ and all specific probability rates $c_i=1$. Figure 2(a) shows $\Theta(N)$ for PSSA-CR, SPDM, and SDM. As expected from the theoretical cost analysis, Θ is $O(1)$ for PSSA-CR and $O(N)$ for SPDM and SDM. PSSA-CR outperforms SPDM for N above a certain break-even point [$N>700$ here; Fig. 2(a)] and is faster than SDM for all N tested. Below the break-even point, the overhead of the additional data structures and the binning involved in PSSA-CR is not amortized by the better scaling of the computational cost. The $O(1)$ scaling for PSSA-CR in this case is realized because the reaction network is weakly coupled (degree of coupling is independent of N) and all G_{Π_i}'s and G_Σ are constant with system size.

In order to test the efficiency of PSSA-CR for a weakly coupled reaction network with *increasing* number of bins, we simulate this test case with specific probability rates c_i randomly chosen between 1 and 10^6 from an exponential distribution. All other simulation parameters are unchanged. Figure 2(b) shows the scaling of Θ for PSSA-CR, SPDM, and SDM. In this multiscale case, G_Σ increases slowly with system size (by 2% over a 16-fold increase in N), leading to a very slow increase in Θ (proportional to $N^{0.028}$ in this case) of PSSA-CR, as predicted by the theoretical cost analysis. Nevertheless, PSSA-CR is more efficient than SPDM for N above a certain break-even point [$N>500$ here; Fig. 2(b)] and more efficient than SDM for all N tested.

In summary, the measured computational cost of PSSA-CR is $O(1)$ for the cyclic chain model if the number of bins is bounded. If G_Σ or G_{Π_i} increase with system size, the computational cost is $O(G_\Sigma+\max\{G_{\Pi_0},\ldots,G_{\Pi_N}\})$, as derived in Appendix A.

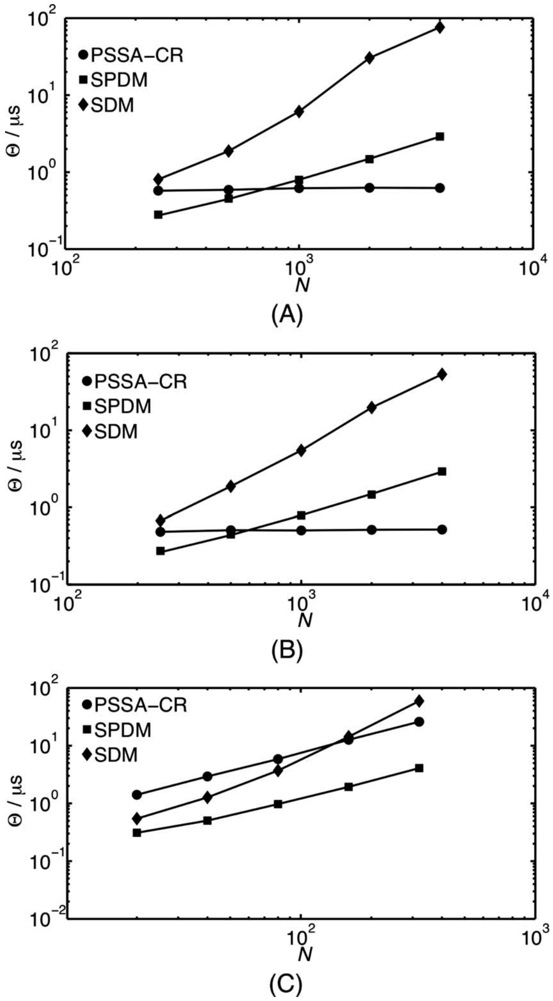

FIG. 2. Computational cost of PSSA-CR (circles), SPDM (squares), and SDM (diamonds). The average CPU time Θ per reaction, averaged over 100 independent runs, is shown as a function of the number of species N. (a) $\Theta(N)$ for the weakly coupled cyclic chain model with bounded dynamic range of nonzero reaction propensities. Θ is $O(1)$ for PSSA-CR and $O(N)$ for SPDM and SDM. (b) $\Theta(N)$ for the weakly coupled cyclic chain model with increasing dynamic range of nonzero reaction propensities. $\Theta \propto N^{0.028}$ for PSSA-CR and $\Theta \propto N^{1}$ for SPDM and SDM. (c) $\Theta(N)$ for the strongly coupled colloidal aggregation model. Θ is $O(N)$ for both PSSA-CR and SPDM, whereas it is $O(N^2)$ for SDM.

B. A strongly coupled reaction network: Colloidal aggregation model

The colloidal aggregation model is given by

$$\begin{aligned} &S_n + S_m \xrightarrow{c_{n,m}} S_{n+m} \quad n=1,\ldots,\left\lfloor\frac{N}{2}\right\rfloor; \quad m=n,\ldots,N-n,\\ &S_p \xrightarrow{\bar{c}_{p,q}} S_q + S_{p-q} \quad p=1,\ldots,N; \quad q=1,\ldots,\left\lfloor\frac{p}{2}\right\rfloor. \end{aligned} \tag{5}$$

For N chemical species, the number reactions is $M=\lfloor\frac{N^2}{2}\rfloor$. The degree of coupling of this reaction network is $3N-7$ and hence scales with system size.

At time $t=0$, we set all $n_i=1$ and all specific probability rates $c_i=1$. Figure 2(c) shows $\Theta(N)$ for PSSA-CR, SPDM, and SDM. Θ is $O(N)$ for PSSA-CR and SPDM, and it is $O(N^2)$ for SDM. The Θ of PSSA-CR is always larger than that for SPDM. This constant offset is caused by the additional overhead of binning and bin reassignments in PSSA-CR, which is not necessary in SPDM. The break-even point of PSSA-CR with SDM is around $N>160$. For systems larger than this, the extra overhead in PSSA-CR is amortized.

IV. CONCLUSIONS AND DISCUSSION

We introduced PSSA-CR, a partial propensity[6] variant of the stochastic simulation algorithm with composition-rejection sampling (SSA-CR).[5] PSSA-CR uses two composition-rejection sampling steps over partial propensities in order to determine the index of the next reaction. Computational efficiency is achieved by grouping the partial propensities and using dyadic binning in the sampling.

PSSA-CR is an exact SSA formulation whose computational cost is $O(N)$ on strongly coupled reaction networks and $O(1)$ on weakly coupled networks with a bounded range of propensities. We presented a theoretical cost analysis of PSSA-CR and benchmarked it on three prototypical test cases: (1) a nonstiff weakly coupled reaction network, (2) a multiscale (stiff) weakly coupled reaction network, and (3) a strongly coupled reaction network. All benchmarks confirmed the theoretically predicted scaling of the computational cost. To our knowledge, PSSA-CR has the best scaling of the computational cost on any type of reaction network.

PSSA-CR, however, inherits the limitations of partial-propensity methods[6] and of SSA-CR.[5] It is limited to chemical reaction networks composed of elementary reactions involving at most two reactants. Nonelementary reactions can be treated by decomposing them into elementary reactions.[3,7] This, however, increases the network size and hence the computational cost of PSSA-CR. For small networks, PSSA-CR is outperformed by other methods due to the additional overhead involved in the composition-rejection sampling. SSA formulations such as SDM,[12] NRM,[4] SSA-CR,[5] PDM, or SPDM[6] might be more efficient here. In addition, PSSA-CR only achieves the $O(1)$ scaling for weakly coupled networks for which ratio of maximum to minimum nonzero reaction propensity is bounded by a constant throughout a simulation.

To our knowledge, PSSA-CR has the best scaling of the computational cost on any class of reaction networks. This, however, does not imply that the actual computational cost of PSSA-CR is lowest in all cases, since the prefactor depends on the data structures involved. If the coupling class of a particular network is not known in practice, PSSA-CR seems a reasonable choice for exact stochastic simulations of large reaction networks. Compared to other partial propensity methods, such as SPDM, the better computational scaling of PSSA-CR for weakly coupled networks is paid for by a larger prefactor in the computational cost for strongly coupled networks.

ACKNOWLEGDMENTS

R.R. was financed by a grant from the Swiss SystemsX.ch initiative, evaluated by the Swiss National Science Foundation. This project was also supported with a grant from the Swiss SystemsX.ch initiative, Grant No. LipidX-2008/011 to I.F.S.

APPENDIX A: COMPUTATIONAL COST OF PSSA-CR

The computational cost of PSSA-CR is determined by the sampling and update steps of the algorithm. Composition-rejection sampling of the group-index I has a cost that is $O(G_\Sigma)$. This is because (a) the composition step involves a linear search over at most G_Σ elements and (b) the computational cost of the rejection step is $O(1)$, since the maximum number of iterations needed is bounded by a constant. Similarly, the computational cost of the composition-rejection sampling of the element index J is $O(\max\{G_{\Pi_0}, \ldots, G_{\Pi_N}\})$.

The computational cost of the update step is $O(N)$. Assuming that the number of species involved in any chemical reaction is $O(1)$ (i.e., does not increase beyond a constant bound as the number of species in the network increases), the number of entries in $\mathbf{\Pi}$ that need to be updated after any reaction has fired scales at most linearly with N.[6] In summary, the total computational cost of PSSA-CR thus is $O(G_\Sigma + \max\{G_{\Pi_0}, \ldots, G_{\Pi_N}\} + N)$.

For weakly coupled reaction networks, the update step becomes $O(1)$, since the number of entries in $\mathbf{\Pi}$ that need to be updated is independent of system size. This reduces the computational cost of PSSA-CR for weakly coupled networks to $O(G_\Sigma + \max\{G_{\Pi_0}, \ldots, G_{\Pi_N}\})$. In addition, if $\Sigma_{\max}$ and $\Pi_{i,\max}$ are bounded for all i, the number of bins $G_\Sigma = \log_2(\Sigma_{\max}/\Sigma_{\min}) + 1$ and $G_{\Pi_i} = \log_2(\Pi_{i,\max}/\Pi_{i,\min}) + 1$ are also bounded. This renders the computation cost of PSSA-CR $O(1)$ for weakly coupled networks that have a bounded dynamic range of propensities.

APPENDIX B: AUTOMATIC INITIALIZATION OF THE PSSA-CR DATA STRUCTURES

Given the stoichiometry matrix, the initial population of the species, and the specific probability rates of all reactions, all data structures used in PSSA-CR are generated automatically (i.e., without user interaction). This needs to be done only once, at the beginning of a simulation.

TABLE II. Algorithm for generating the partial-propensity structure $\mathbf{\Pi}$ using the definitions given in Sec. II B.

```
1. 1. Initialize $\boldsymbol{\nu}$, $\mathbf{n}(0)$, and $\mathbf{c}$.
2. Using $\boldsymbol{\nu}$, build a list of all reactants in each reaction. The reactants
   have a negative stoichiometry, except in the case of a source reaction,
   where is it zero. If no species in a reaction has a negative
   stoichiometry, then the reactant index is 0 and the reaction is a source
   reaction.
3. Go through the reactant lists of all reactions:
   3.1. If the number of distinct reactants in a reaction is 2, compute the
        partial propensity of this reaction by factoring out the population of
        the species with the *smaller* index $i$ from the full reaction
        propensity. Append this partial propensity to $\mathbf{\Pi}_i$.
   3.2. If the number of reactants in a reaction is 1, then check
        3.2.1. If it is a bimolecular reaction between the same species $S_i$,
               store the corresponding partial propensity in $\mathbf{\Pi}_i$.
        3.2.2. If it is a unimolecular reaction with species $S_i$ as a reactant,
               store the partial propensity in $\mathbf{\Pi}_i$.
        3.2.3. If it is a source reaction $(i=0)$, store the partial propensity in
               $\mathbf{\Pi}_0$.
4. Stop.
```

The stoichiometry matrix $\boldsymbol{\nu}$ is an $(N+1)\times M$ matrix, where $\nu_{i,j}$ stores the stoichiometry of the ith species in reaction j. The 0th species is the source reservoir. N and M are the number of species and reactions, respectively. The vector $\mathbf{n}(t)$ contains the population of all species at time t, and $\mathbf{c}$ contains the specific probability rates all reactions. The partial propensity structure $\mathbf{\Pi}$ is constructed from $\boldsymbol{\nu}$, $\mathbf{n}(0)$, and $\mathbf{c}$ using the algorithm given in Table II and the definitions from Sec. II B. From this $\mathbf{\Pi}$, the group-sum array $\mathbf{\Lambda}$ is computed as $\Lambda_i = \Sigma_j \Pi_{i,j}$ and the total propensity in each group is $\Sigma_i = n_i \Lambda_i, \quad i = 1, \ldots, N$ $(\Sigma_0 = \Lambda_0)$, as outlined in Sec. II B.

[1] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976).
[2] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[3] D. T. Gillespie, Physica A **188**, 404 (1992).
[4] M. A. Gibson and J. Bruck, J. Phys. Chem. A **104**, 1876 (2000).
[5] A. Slepoy, A. P. Thompson, and S. J. Plimpton, J. Chem. Phys. **128**, 205101 (2008).
[6] R. Ramaswamy, N. González-Segredo, and I. F. Sbalzarini, J. Chem. Phys. **130**, 244104 (2009).
[7] K. R. Schneider and T. Wilhelm, J. Math. Biol. **40**, 443 (2000).
[8] L. Devroye, *Non-Uniform Random Variate Generation* (Springer, New York, 1986).
[9] T. Wilhelm, J. Math. Chem. **27**, 71 (2000).
[10] Y. Cao, H. Li, and L. Petzold, J. Chem. Phys. **121**, 4059 (2004).
[11] P. Meakin, Annu. Rev. Phys. Chem. **39**, 237 (1988).
[12] J. M. McCollum, G. D. Peterson, C. D. Cox, M. L. Simpson, and N. F. Samatova, Comput. Biol. Chem. **30**, 39 (2006).