Dynamic Partitioning of Large Discrete Event Biological Systems for Hybrid Simulation and Analysis

Kaustubh R. Joshi, Natasha Neogi and William H. Sanders

Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana IL 61801

Abstract. Biological systems involving genetic reactions are large discrete event systems, and often contain certain species that occur in small quantities, and others that occur in large quantities, leading to a difficulty in modeling and simulation. Small populations inhibit the usefulness of utilizing differential equations to represent the system, while the large populations cause stochastic discrete event simulation to become computationally intensive. This paper presents an algorithmic approach for the dynamic partitioning and stochastic hybrid simulation of biological systems. The algorithm uses a Poisson approximation for discrete event generation and ordinary differential equations to model continuous behavior. The populations are dynamically partitioned so that some populations are simulated in a discrete stochastic fashion, while others are simulated by continuous differential equations, and this partition between discrete and continuous behavior is updated. A hybrid model of a biological toggle switch is used to illustrate our approach, and yields promising results.

1 Introduction

The advancement of experimental techniques and rapid accumulation of genetic information have opened a new frontier in biomedical engineering. The ability to engineer artificial gene regulatory networks with sophisticated computational and functional abilities grows ever closer due to the availability of well-characterized components from natural gene networks. Hence, the construction, analysis, and interpretation of qualitative and quantitative models acts to drive the field forward [14]. There are several complementary approaches involving discrete and stochastic simulation that can be used to model gene networks; they are surveyed in this paper. The goal is to use such descriptions to accurately predict the properties and function of modules connected into networks, and to make *in silico* suggestions for optimal design strategies prior to implementation *in vivo*. However, as the models become progressively more complex, the algorithms become computationally expensive [15].

For systems in which species fluctuate by varying orders of magnitude, the largest fluctuating species require the most time to simulate stochastically because exact stochastic simulation techniques scale with the number of reaction events [11]. The use of stochastic partitioning, whereby the state of the reaction is modeled using extents of reactions rather than molecules of species, thereby leading to the ability to separate the state into subsets of fast and slow reactions, allows for the reduction in the computational burden for simulation. The fast reactions can then be approximated deterministically as ordinary differential equations (or stochastically by the Langevin Equations), while the slow reactions can be treated as stochastic events with time-varying reaction rates. However, the partition must be evaluated periodically, as reactions can change their behavior as time progresses.

In the next section, an informal discussion of the hybrid nature of large, discrete event biological systems is conducted. A survey of simulation and analysis techniques to date is performed, and the potential limitations of these methods are discussed. Section 3 focuses on the various modeling techniques used to encapsulate biological systems. Deterministic differential equations, the Langevin approximation, and discrete stochastic simulation models are developed. A Poisson process is used to define the discrete reaction rate for continuous mass action models in Section 4. Section 5 discusses several strategies to partition reactions in subsets that can be modeled continuously or discretely. A method for sampling from a non-homogeneous Poisson process in order to determine the next reaction and time to next reaction is outlined in Section 6. Section 7 is divided into two parts. The first part describes our implementation of a hybrid simulator that performs dynamic population-based partitioning to separate fast and slow reactions. The fast reactions are simulated using ordinary continuous differential equations, and the slow reactions are simulated using a non-homogeneous Poisson process. The simulation algorithm is outlined, and implementation issues are discussed. The second part of Section 7 contains a discussion of the model of a toggle switch that is simulated using the hybrid simulator.

2 Problems with Large Discrete Event Biological Systems

In many systems, the probability of occurrence of any event in an infinitesimal time interval dt will depend only on the number of individuals at time t and on parameters that might depend on t, that is, the future evolution of the system depends only on its present state, and not on the system's history. Such processes can be represented as Markov processes. These systems are memoryless, and do not need to keep a state history to determine their evolution.

Systems that are Markov in nature and have exponential event distributions can be modeled using Monte Carlo simulation. The notion of Monte Carlo simulation comes from the solution of the inverse problem, in which differential equations are approximated by stochastic jumps obtained by the Monte Carlo method. This simulation is a simple realization of the Markov process, and is often called the *Feller process* [3].

If the population size is large, then in the limiting case, the fractions of the total population represented by each species are the relevant quantities to be considered. This notion is called the *Mass-Action Law*, and is commonly accepted in physical chemistry [2].

2.1 Current Simulation and Analysis Techniques

Solutions to the stochastic formulation of coupled chemical reactions can be computed using the Monte Carlo procedure specified by Gillespie [7]. The algorithm calculates a stochastic description of the temporal behavior of the coupled reactions. This description can be shown to have a more rigorous physical basis than conventional kinetic equation formulations based on the assumption that changes in the chemical reaction system over time are both continuous and deterministic. This assumption of continuous and deterministic behavior is invalid for low concentrations of reactant species or sufficiently slow reactant rates. The Direct and First Reaction methods [7], outlined by Gillespie, are simulation algorithms that calculate the probabilistic outcome of each discrete chemical event and the resulting changes in the number of each molecular species. It should be noted that, in the limiting case of a large number of reactant molecules, these methods are entirely equivalent to the solution of the traditional kinetic differential equations derived from the mass action law [8].

As models become progressively more complex, however, these algorithms often become expensive computationally. Several techniques have been employed to reduce the computational burden. Employment of a deterministic equilibrium assumption on polymerization reaction kinetics has yielded a decrease in computational complexity, as shown by He, Zhang, Chen, and Yang [12]. Gibson and Bruck have refined Gillespie's First Reaction algorithm to reduce the required number of pseudo-random numbers that must be generated, a technique that works best for systems in which some reactions occur much more frequently than others [5]. Rao and Arkin have demonstrated how to numerically simulate systems reduced by the quasi-steady-state assumption [21]. This work expands upon ideas by Janssen [13] and Vlad and Pop [26], who examined the adiabatic elimination of fast relaxing variables in stochastic chemical kinetics. Resat, Wiley, and Dixon applied a probability-weighted Monte Carlo approach to address systems with reaction rates that vary by several orders of magnitude, but this method increases error in species fluctuations [10]. Haseltine and Rawlings [11] expanded upon the idea of a partitioned system and simulation via Gillespie's direct method to construct approximations that reduce the computational burden for simulation. Their work sets up the framework for partitioning a system into discrete and continuous reactions, but addresses only static partitioning.

Recently, Solari et al. developed an approximation to stochastic population dynamics based on almost independent Poisson processes whose parameters obey a set of coupled ordinary differential equations [22, 23, 1]. Error bounds for the moment-generating function have been developed. For large populations, the Poisson approximation becomes a discrete integration of the Langevin approximation [16]. A implementation of this simulation method has been outlined, and several improvements to improve its efficiency are discussed in the relevant literature [22].

Unfortunately, for large-scale biological systems that have complex macromolecules (i.e., proteins), long and short time scales, and interactions of several different molecule types, no one approach has yet to yield satisfactory results in all the arenas of accuracy, computational complexity, scalability, and coverage.

3 Modeling

A coupled system of p biochemical reactions can be modeled by equations of the form:

$$\sum_i R_i^{\alpha_i} \longrightarrow \sum_j P_j^{\beta_j} \tag{1}$$

where α_i molecules each of n different reactants R_i react to form β_j molecules each of m different products $P_j, i, j = 1 \ldots p$. If it is assumed that enough molecules of each species are present such that the number of molecules can be approximated as a continuous quantity that varies deterministically over time, the concentration of each species can be written in terms of the concentrations $[R_i], [P_i]$ of all other species via a set of differential equations. However, when the concentrations of certain species drop below a given level, the assumption that the number of molecules can be approximated as a continuously varying deterministic function is no longer valid.

Alternatively, it can be assumed that non-reactive collisions occur far more often than reactive collisions, and thus that the fast dynamics of motion can be neglected [5]. Thus, the system may be represented best using the number of each kind of molecule. Using this approach, it can be concluded that the probability that a certain reaction μ will take place in the next instant of time dt is given by $a_\mu dt + o(dt)$, where a_μ is independent of dt and $o(dt)$ represents terms that are negligible for small dt. However, a_μ may depend on the current number of molecules of each kind, and the current time. Furthermore, it may depend on quantities such as temperature and volume, which may change with time. The state of the system in the stochastic framework is defined by the number of molecules of each species and changes discretely whenever one of the reactions occurs. Formally, we can define the probability that the state of a system S changes to S' via the occurrence of the reaction as [5]:

$$P(S', t + dt | S, t) = a_\mu dt + o(dt) \tag{2}$$

Note that since a_μ, and thus the transition probability, are dependent only on the current state and not on previous states, the underlying process is Markov in nature. If an individual probability variable is created for each possible state of the system, then Equation 2 can be used to write out the system of coupled differential equations that defines the system, such as:

$$\frac{d}{dt} P(S_j, t + dt | S_i, t) = f(a_{\mu_{ij}}, P(S_i, t)) \tag{3}$$

for all reactions μ_{ij} that bring the system state from S_i to S_j. This set of differential equations has probabilities as variables and is referred to as a *Master*

Equation. It is a direct consequence of the satisfaction of the forward Kolmogorov equation by the probabilities [6]. For systems with very few states, the entire set of Master Equations may be written out and explicitly solved. For large systems, however, this approach quickly becomes intractable [5].

The Master Equation, can, however, be approximated by the stochastic Langevin equations, which are differential equations, and therefore scale more easily with respect to population size. If the characteristic size of the system is defined by Ω, the Master Equation is recast in terms of intensity variables (concentrations), and a Kramers-Moyal expansion is performed, the Master equation results in a system size expansion in Ω. In the limiting case as Ω becomes large, the discrete Master Equation can be approximated by its first two differential moments with the continuous Fokker-Planck equation, and has an Ito solution of the form [5]:

$$d[\mu_i] = b_i(\mu)dt + \sqrt{b_i(\mu)}dW_i \tag{4}$$

where:

$$b_i(\mu) = \sum_{i=0}^{all\mu} a_i P(S,t) \tag{5}$$

and dW_i is a Weiner process. Equation (5) is a stochastic differential equation that specifies the evolution of the trajectories of the system state [7]. The error induced by this approximation is directly related to the size of the system Ω. Even if the system size Ω is large, the Langevin approximation will most likely be valid for only a subset of the reactions [6].

We wish to create a process model for the large, discrete event system that, in the limiting case of large population sizes, captures the Langevin equations, and in the case of small population sizes, can be represented using stochastic discrete interactions.

4 Poisson Processes for Mass-Action Models

Intuitively, the mass action law formalizes the notion that if the size of a system (both in populations and the environment) increases, then the rate of actions also increases. Mathematically, if the rate of the i^{th} reaction is $\lambda_i(X)$, which is a function of the population size X, then the mass action law enforces the limiting property ([8]) $\lim_{X\to\infty} \lambda_i(X) = k\lambda_i(X/k)$. More specifically, in the context of chemical reactions, the mass action law states that the rate of actions is proportional to the product of the concentration of the dependent populations. Consider a system with M substances (populations), in which r reactions occur. Let $X_i, i = 1, 2, \ldots, M$ be the quantities of each substance. Each of the r reactions is represented as follows:

$$R_r = \sum_{i=1}^{M} \alpha_{i,r} X_i \xrightarrow{k_r} \sum_{i=1}^{M} \beta_{i,r} X_i \tag{6}$$

where k_r is the kinetic reaction constant associated with each reaction.

The system specified by Equation (7) can then be modeled using a Continuous Time Markov Chain (CTMC), where each state $\overrightarrow{X} = (X_i = v | i = 1 \dots M, v \in \mathbb{R})$ corresponds to a distinct valuation for each X_i. In a state, each reaction fires after an exponentially distributed random delay whose rate is given by:

$$\lambda_r(X) = \frac{k_r}{V^{(\sum_{i=1}^{M} \alpha_{i,r})-1}} \prod_{i=1}^{M} \frac{X_i!}{(X_i - \alpha_{i,r})!} \tag{7}$$

As the population size increases, that is, $X_i \to \infty, i = 1 \dots M$, the limiting behavior of the above CTMC can be described by the following differential equations:

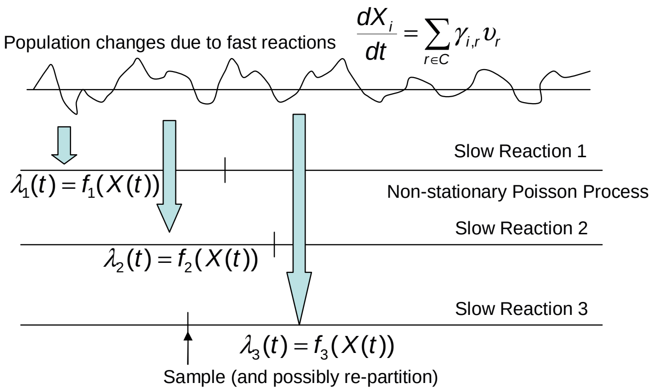
$$\lim_{X_1,\dots,X_M \to \infty} \frac{dX_i}{dt} = \sum_r (\beta_{i,r} - \alpha_{i,r}) k_r \prod_{i=1}^{M} [X_i]^{\alpha_{i,r}} \tag{8}$$

where $[X_i] = X_i / V$ is the concentration of the substance X_i in the limit. A first order correction for stochastic departure can be introduced to obtain the corresponding Langevin Equations. For the purposes of this paper, we shall instead approximate the continuous behavior using deterministic ordinary differential equations. In future work, we shall investigate the implementation of stochastic differential equations.

We wish to create an approach to simulating large, discrete event systems that, in the limiting case of large population sizes, can be modeled by continuous (and possibly stochastic) differential equations, and in the case of small population sizes, can be represented using event-by-event realizations of the Monte-Carlo simulation method. We hope to leverage advantages inherent to meshing the Monte-Carlo and continuous approaches in order to achieve either computational savings or an increase accuracy over an approach that employes a single simulation method.

5 Partitioning Techniques: Continuous versus Discrete Behavior

Our basic approach is to partition the set R of reactions into a set of *continuous* reactions and a set of *discrete* reactions. The continuous reactions evolve using the differential equations, while the discrete reactions evolve using standard CTMC simulation. During the occurrence of two consecutive discrete reactions, the system state changes due to the continuous reactions. Since these continuous reactions could change the quantities of substances on which the rates of the discrete reactions depend, the discrete reactions have to be represented by non-homogeneous Poisson processes (rate is time varying). Between two consecutive discrete events, progress of the continuous reactions is simulated using standard differential equations simulation.



$\frac{dX_i}{dt} = \sum_{r \in C} \gamma_{i,r} \upsilon_r$
Population changes due to fast reactions
Slow Reaction 1
Non-stationary Poisson Process
$\lambda_1(t) = f_1(X(t))$
Slow Reaction 2
$\lambda_2(t) = f_2(X(t))$
Slow Reaction 3
$\lambda_3(t) = f_3(X(t))$
Sample (and possibly re-partition)

Fig. 1. Partitioning of State Variables into Continuous and Discrete Classes

5.1 Dynamic Partitioning

Although the determination of which reactions exhibit deterministic behavior versus which exhibit stochastic behavior could be done statically based on prior knowledge about the system, the classification into stochastic or deterministic is intrinsically dependent on the system state, which evolves over time. Therefore, more flexibility (and accuracy) can be obtained by deciding the partition as the system evolves (Figure 1). The partition is based on the rates of reaction (as given by Equation 8 and 9, and re-partitioning can be performed either only after a discrete transition, or as and when needed (i.e. even between discrete events). Although partitioning only after discrete transitions is computationally less expensive (since one does not need to monitor the differential equation evolution), it has the drawback that it is possible for the state of the system to change such that the assumptions made by the continuous regime are no longer valid. Therefore, we follow the approach of allowing re-partitioning at any time during the evolution of the system. The approaches to perform such a partitioning are given below. They each have their advantages and their drawbacks.

Population Partitioning In the population partitioning approach, the basic idea is to partition the various populations X_i in the system into a set of continuous populations, and a set of discrete populations. Let $P = X_i, \text{ for } i = 1, \dots, M$ be the set of populations in the system. For some state $\overrightarrow{X}$, $C_s(\overrightarrow{X}) \subset P$ is the set of continuous populations, and $D_s(\overrightarrow{X}) = P - C_s(\overrightarrow{X})$ is the set of discrete populations. The set of reactions R is also partitioned into two sets; those which modify any discrete population $D_r(\overrightarrow{X})$, and those that do not $C_r(\overrightarrow{X})$.

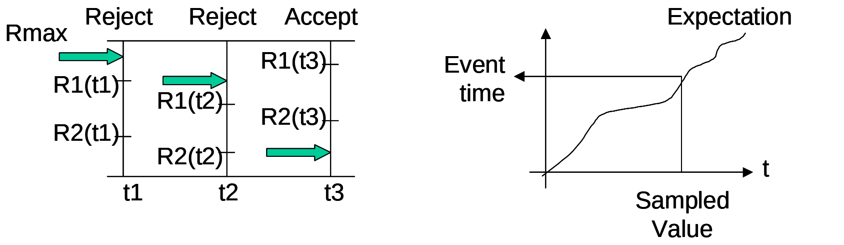
After the occurrence of each event (or if the evolution of the continuous part of the system crosses some switching threshold k), we reconstruct $C_s(\overrightarrow{X})$ and $D_s(\overrightarrow{X})$ if needed using the rule: if $X_i > k$ then $X_i \in C_s(\overrightarrow{X})$ for the threshold k. Initially, the threshold k would be a user definable parameter based on experimental determination of the magnitude at which the discrete behavior of the species in question is reasonably approximated by continuous equations for the given environmental conditions. If the performance of the simulation algorithm is sensitive to disturbances in k, it becomes necessary to define an *error bound* on k, and to reformulate the algorithm so that it adaptively selects k to be the lowest value that gives a bounded error for any propagation of the simulation within one discrete time step. This technique is similar to the technique used by adaptive meshing algorithms that are used to pick the step size in the numerical solution of differential equations [6]. The error bounds derived on the simulation of non-homogeneous Poisson processes [23] would be useful in this process to limit the highest value selected for k.

Partitioning based on population size is intuitive and reactions may change from discrete to continuous only at discrete events, while reactions may change from continuous to discrete at any point the population size drops below the threshold k. However, only those reactions that do not modify any discrete populations can be simulated using differential equations. The remaining reactions must be simulated using CTMC simulation.

Action Partitioning In this potential approach, the reactions are partitioned directly without partitioning the populations as well. This partitioning is done strictly based on the rate of reaction. Hence, some limiting condition that provides an estimate of the error between the stochastic model and the deterministic model for a reaction with a given rate must be derived to act as an upper bound. It is unclear at present what exact metric will best serve this purpose. All fast reactions can then be simulated using differential equations, resulting in a great computational savings, thereby greatly increasing the size of system which can be modeled. However, reactions may change from discrete to continuous at any time, and hence the partitions may have to be updated at every differential equation time step. Stochastically simulating reactions which depend on both continuous and discrete variables is more computationally intensive, due to the increase in Poisson parameters λ_i in the stochastic process. All the populations can change between two discrete events. Hence the number of continuous parameters to the continuous Poisson process increases, thus decreasing its efficiency. At this point, it is unclear how to exactly to best implement this approach, and it will be investigated more thoroughly in future work.

6 Sampling from a Non-Homogeneous Poisson Process

Using the partitioning approaches described in Section 5, the discrete event simulation problem reduces to one of simulating a non-homogeneous Poisson process, i.e. one in which the rate changes continuously with time. Such a process can be simulated using the technique of *thinning* [17]. Thinning is essentially a uniformization technique, whereby the variable rate of the Poisson process is uniformized using a constant rate λ_{max} that upper bounds the varying rate of the process between updates. An exponentially distributed random number with rate λ_{max} is then generated to represent a tentative next arrival time for the non



(a) Accept/Reject Process for Next Event (b) Expected Value of Next Event Time

Fig. 2. (a) Next event generation: Arrow represents a sampled uniform random number at every step (b) "Time to Next Event" Process Generation

homogeneous Poisson process. Let the current time be t and the tentative next arrival time be Δt. The actual rate $\lambda(t + \Delta t)$ of the non-homogeneous Poisson process is computed at time $t+\Delta t$. Then, the tentative arrival time is accepted as the next arrival time of the non-homogeneous Poisson process with probability (Figure 2): $\lambda(t + \Delta t)/\lambda_{max}$ and rejected with probability $1 - \lambda(t + \Delta t)/\lambda_{max}$.

If the arrival time is rejected, the process of generating tentative arrival times can be repeated with the current time set to $t+\Delta t$, due to the memoryless nature of the Poisson process, until an arrival time is finally accepted. Let the random variable N represent the number of arrival events (all but the last one being tentative) that need to be generated to simulate one discrete event. If Δt_i is the i^{th} tentative arrival time, then the probability mass function for N is given by:

$$P[N = n] = \prod_{i=1}^{n-1} \left(1 - \frac{\lambda(t + \sum_{j=1}^{i} \Delta t_j)}{\lambda_{max}}\right) \frac{\lambda(t + \sum_{j=1}^{n} \Delta t_j)}{\lambda_{max}} \tag{9}$$

Let: $0 \leqslant r(i) = \lambda(t+\sum_{j=1} i\Delta t_j)/\lambda_{max} \leqslant 1$ be the acceptance probability for the i^{th} sample, and let $r_{max} = \max_i\{r(i)\}$ be the maximum possible value for $r(n)$. Then, the average value of N can be upper bounded by the average value of a geometric random variable with success probability r_{max}, i.e. $E[N] \leqslant 1/r_{max}$.

Clearly, bringing this ratio as close to 1 as possible will result in the generation of fewer event arrival time and improve the running time of the algorithm. Two techniques can be used to do this task. First, using the memoryless nature of the Poisson process, we can make more informed estimates of λ_{max} at the time of each tentative event rejection. Such a technique is similar to adaptive uniformization [25]. The second technique is to use the law of conservation of mass to compute an upper limit. However, this approach is clearly conservative because the law of conservation of mass provides an upper limit on the amount of substance that can ever be generated in the given system. A better approach might be to optimistically assume a smaller upper bound $\lambda_{max}(optimistic)$, relying on the fact that most of the generated arrival times will not be too far away. Then, while integrating, if the actual rates ever become higher than $\lambda_{max}(optimistic)$, then we redo the computation from the last event with a larger λ_{max}(*optimistic*). However, such a process can introduce bias into the system, and care must be taken to correct for this bias every time a computation is redone. The efficiency of this computation can be calculated based on the exponential distribution function. Clearly, the more number of times we have to recompute, the less gains we obtain.

An alternative to the thinning technique that we plan to explore in the future is the technique of time-scale transformation of a homogeneous (rate one) Poisson process via the inverse of the integrated rate function [17] as shown in Figure 2 (b). This technique seems promising because the required integration can be easily incorporated into the ODE system representing the continuous reactions, and thus be carried out with little overhead.

7 Simulator Implementation and Example

Parts of the hybrid simulation techniques described earlier were implemented in an extensible simulation framework we are developing. The framework was written in ISO C++, and is designed in a modular way to allow the plugging in of different simulation algorithms (thus providing the means to compare techniques). The implementation of the hybrid simulation algorithm and an example of a genetic regulatory system that was simulated using the algorithm are described in this section.

7.1 Simulator Implementation

The hybrid simulation algorithm that was implemented in the simulation framework is outlined as shown in Algorithm 1. It accepts a biochemical system consisting of a set of substances X, and a set of reactions R defined by α, β (as in Equation 6). The algorithm employs true dynamic switching (potentially repartitioning when either the discrete or continuous populations change). The repartitioning scheme is based on population sizes, whereby a species is labelled as discrete (belonging to set D_s) if its population falls below a certain *switching threshold*, and continuous (belonging to set D_c) otherwise. Reactions are labelled as discrete (belonging to set D_r) if any of their dependant populations are discrete, and continuous (belonging to set C_r) otherwise. This criteria ensures that the validity conditions for the ODE representation (i.e. $X \rightarrow \infty$) are closely approximated when that representation is used.

The algorithm uses the thinning technique (described in Section 6) to simulate the resulting non-homogenous Poisson process. In our implementation, the maximum rates required for this technique are manually specified based on the population limits imposed by the law of mass conservation. However, we are currently investigating the possibility of automatic computation of tighter rate limits (which are state-dependent) using symbolic state-space exploration techniques.

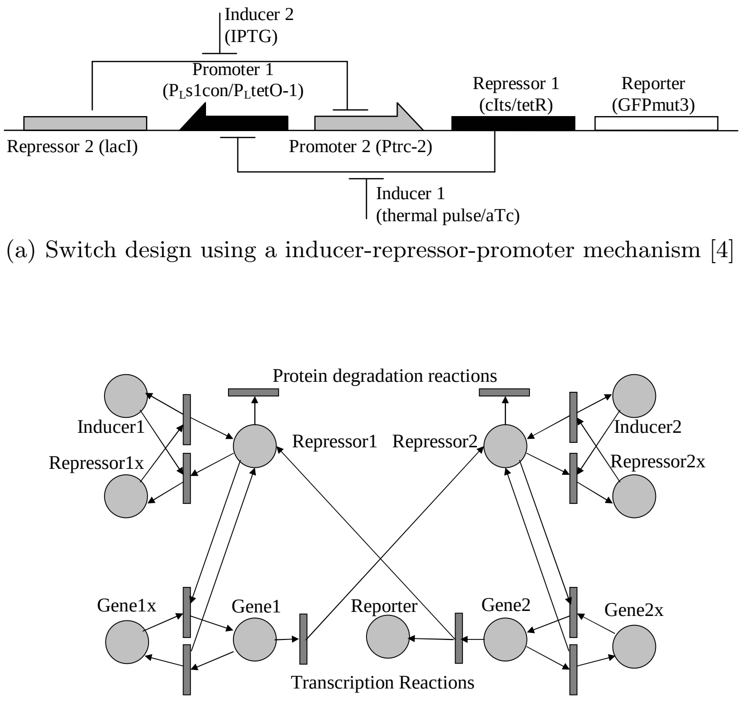
The main loop of the algorithm consists of generating a potential firing time (by sampling a negative exponential random number generator through

Algorithm 1: Hybrid Simulation Algorithm

```
HybridSim($t_1$,$t_2$,$X$,$R$)
$t \leftarrow t_1$
while $t < t_2$
    if repartition
        $C_s \leftarrow D_s \leftarrow C_r \leftarrow D_r \leftarrow \phi$
        foreach $x_i \in X$, if $|x_i| < threshold_i$ then $D_s \leftarrow D_s \cup x_i$
                                                  else $C_s \leftarrow C_s \cup x_i$
        foreach $r \in R$, if REACTANTS(r)$\cap D_s \neq \phi$ then $D_r \leftarrow D_r \cup r$
                                                     else $C_r \leftarrow C_r \cup r$
        Stoichiometric matrix $\gamma_{i,r} \leftarrow \begin{cases} \beta_{i,r} - \alpha_{i,r} & \text{if } r \in C_r \\ 0 & \text{otherwise} \end{cases}$
        $\forall r \in D_r$ compute $\lambda_{max}(i)$
        $\lambda_{max} \leftarrow \sum_{r \in D_r} \lambda_{max}(i)$
    $\Delta t \leftarrow$RANDOM(Exp($\lambda_{max}$)), $\Delta \leftarrow \min(\Delta t, t_2)$
    $\Delta t' \leftarrow$ODESOLVE($dx_i/dt = \sum_r \gamma_{i,r} k_r \prod_{i=1}^{M} [|x_i|]^{\alpha_{i,r}}$, over$[t, t+\Delta]$)
    if $\Delta t' < \Delta t$ (threshold crossed in OdeSolve) then repartition$\leftarrow$true
    else
        foreach $r \in D_r$ $\lambda(r) = k_r / V^{\sum_{i=1}^{M} \alpha_{i,r} - 1} \prod_{i=1}^{M} |x_i|!/(|x_i| - \alpha_{i,r})!$
        if $(u \leftarrow$ RANDOM$(U[0,1])) < \sum_{r \in D_r} \lambda(r)/\lambda_{max}$
            $m \leftarrow \min r \in D_r$, s.t. $\left\{ u < \sum_{j=1}^{r} \lambda(j)/\lambda_{max} \right\}$
            FIRE(Reaction m)
            if threshold crossed during firing then repartition$\leftarrow$true
    $t \leftarrow t + \Delta t'$
```

the `Random` function), simulating the ODEs up to that time, and then determining whether the firing time was real or virtual using the conditions specified in Section 6. If the firing time was real, then a reaction is chosen (using a Uniform random number generator) and fired. If the firing causes the population of a substance to cross the switching threshold, then the system is marked for repartitioning.

The function `OdeSolve` uses a standard 5^{th} order 6-point Cash-Karp Runge-Kutta solver [20] to simulate the ODEs. It uses adaptive step-size computation, and is designed stop if the population of any continuous substance falls below the switching threshold. The function returns the time at which ODE simulation was stopped. Thus, a stopping time that is earlier than the next firing time indicates a switch from continuous to discrete, and invokes repartitioning. A technical issue arises if the threshold is crossed in the middle of the solver step. Currently, we resolve this issue by performing a linear interpolation of the values of the continuous populations at the end-points of the step to determine the crossover point (i.e. $t_{switch} = t_1 + h(threshold - x_1)/(x_2 - x_1)$, where t_1 is the lower endpoint, h the step-size, and x_1, x_2 are the population values at the endpoints of the step). However, if needed, an higher order interpolation (up to the order of the ODE solution method) can be used instead. In future work, we plan to implement and compare the other approaches for equation partitioning (i.e. reaction partitioning) and non-homogenous Poisson process simulation (i.e. the time-scale transformation technique) described in this paper.



(a) Switch design using a inducer-repressor-promoter mechanism [4]

(b) Stochastic Petri net model for molecular interactions

Fig. 3. A genetic toggle switch implemented in Escherichia coli

7.2 A Genetic Toggle Switch

The hybrid simulation algorithm described earlier was validated using the model of a genetic toggle switch described by Gardner et. al. [4], and shown in 3(a). The switch toggles between stable expression of either of two genes in response to external signals. It is constructed from two promoters and corresponding repressors, designed so that each promoter is be inhibited by the repressor transcribed by the other promoter. The circuit has two stable states: the “high” state with Promoter 2 on, and the “low” state with Promoter 1 on, given comparable promoter kinetics. Gardner et al. [4] demonstrate how this inherently bi-stable relation can be engineered by using the *lac operon* with the $IPTG$ inducable $Ptrc$–2 promoter and $LacI$ repressor as one promotor-repressor pair (Promoter 2, Repressor 2). For the other promoter-repressor pair (Promoter 1, Repressor 1), either the aTc inducible P_LtetO–1 promoter and $TetR$ repressor pair, or the thermally inducible P_Ls1con promoter and $cIts$ repressor pair is used. The state of the switch is sensed using a reporter gene coding for green fluorescent protein (GFP) directly downstream from the *Repressor 1* transcription site. The “high” state occurs with the introduction of Inducer 2, $IPTG$, which binds to Repressor 2 ($LacI$) to turn Promoter 2 ($Ptrc$–2) on, and results in production of Repressor 1 and the reporter GFP. The “low” state occurs when a pulse of Inducer 1 (aTc

or a temperature pulse) is introduced, causing Promoter 1 to turn on, and Repressor 2 to be produced. Gardner et al. found that all cells grown in colonies for six hours in the presence of *IPTG* under conditions for *cIts* stability remained in the Promoter 2 on state after the removal of *IPTG*. Transiently increasing the temperature so that *cIts* was unstable for 7 hours resulted in the toggling of the system state to Promoter 1 on.

Goss and Peccoud [9] demonstrate how genetic phenomena (such as the toggle switch) can be expressed in the form of stochastic Petri nets (SPNs) that model molecular populations and their interactions. Such a SPN for the genetic toggle switch is shown in Figure 3(b). Based on the SPN, the molecular reactions that describe the behavior of the switch are:

$$\begin{aligned} Inducer2 + Repressor2 &\leftrightarrow Repressor2x \\ 4Repressor2 + Gene2 &\leftrightarrow Gene2x \\ Gene2 &\rightarrow Gene2 + Reporter + Repressor1 \\ Inducer1 + Repressor1 &\leftrightarrow Repressor1x \\ 4Repressor1 + Gene1 &\leftrightarrow Gene1x \\ Gene1 &\rightarrow Gene1 + Repressor2 \end{aligned} \tag{10}$$

For the purpose of simulation, the reporter distribution was simulated for an initial setting of the toggle switch as being in the "high" state, that is, with the *IPTG* Inducer 2 signal present, and Gene 2 expressed. Then, the signal was removed at 500 seconds, and the *aTc* Inducer 2 signal was applied at 3000 seconds. Since the toggle switch represents a relatively small system, the switching threshold for our simulation runs was set to 75% of the highest populations achieved in the system. This resulted in the active repressor being classified as continuous. Whenever an inducer signal was applied, it was also large enough to be classified as continuous, thus making the inducer-repressor reaction continuous, and allowing several fast reactions to be skipped over.

The system was simulated with various values for the kinetic rate constants and various population sizes for the inducer signals, and the evolution of the reporter protein transcription was observed. The results compared favorably with those found in experimental literature [4, 19] in that the behavior was found to be correct, and switching behavior was observed for values of the kinetic rate constants predicted by Gardner et. al. [4]. Although extensive numerical comparisons were not performed, the results were similar to a golden run performed using a discrete event simulator (implementing Gillespie's technique [7]) implemented in the simulation framework. Presently, we are working on improving the efficiency of the algorithm with the aim to conduct a comprehensive quantitative comparison between the accuracy and simulation time of the hybrid approach and a completely discrete simulator implementing Gillespie's algorithm. Additionally, are attempting to simulate a more complex genetic regulatory system (the Glucose-Galactose switch [24, 18]) containing both Protein-Protein interactions and regulated gene expression (and thus large fluctuations in populations) where it is hoped that hybrid simulation will make a large impact.

8 Conclusions and Future Work

Quantitative models and simulation techniques for the analysis of complex biological and genetic processes are indispensable in the construction and evaluation of these systems. In this paper we have described techniques that can be used for dynamically partitioning a system of biochemical equations into continuous and discrete parts, and for subsequently simulating the partitioned system via a combination of discrete event simulation and ODE solution. This meshing of techniques can allow the solver to retain the best features of both techniques. The simulation algorithm was validated on a gene regulatory network model implementing the functionality of a toggle switch, and was found to provide comparable results with exact stochastic simulation. Our future efforts will be directed towards performing algorithmic error analysis of the different approaches described in this paper, and quantitatively comparing these approaches with both discrete event and ODE simulations in order to assess their speed and accuracy properties. We will also work towards further increasing the efficiency and level of automation of the hybrid simulation algorithm using symbolic state-space exploration techniques.

9 Acknowledgements

This work was supported, in part, by a grant from the National Science Foundation under the Embedded and Hybrid System Project Grant Number: CCR-0311616, and by a gift from the Pioneer Hi-Bred International Corp. The authors thank Helen Gill of the NSF, and Jean Peccoud of Pioneer for their encouragement, help, and support. They also thank Jenny Applequist for improving the readability of the paper.

References

1. Juan P. Aparicio and H. G. Solari. Population dynamics: Poisson approximation and its relation to the langevin process. *Phy. Rev.*, 86(18):4183–4186, April 2001.
2. T.C. Gard. *Introduction to Stochastic Differential Equations.* Marcel Dekker, New York, 1998.
3. C.W. Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences.* Springer-Verlag, Berlin, 2nd edition, 1990.
4. T.J. Gardner, C.R Cantor, and J.J. Collins. Construction of a genetic toggle switch in escherichia coli. *Nature*, 403:339–342, 2000.
5. M.A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A*, 104:1876–1889, 2000.
6. M.S. Gibson and E. Mjolsness. *Computational Methods for Modelling Biochemical Networks.* MIT Press, Cambridge, MA, in press.
7. D.T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. of Physical Chemistry*, 81:2340–2361, 1977.
8. D.T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188:404–445, 1992.

9. Peter J.E. Goss and Jean Peccoud. Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets. *Proc. Natl. Acad. Sci.*, 95(12):6750–6755, June 1998.
10. Resat H., Wiley H.S., and Dixon D.A. Probability weighted dynamic monte carlo method for reaction kinetics simulations. *J. of Physical Chemistry B*, 105(44):11026–11034, 2001.
11. E. L. Haseltine and J.B. Rawlings. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *J. of Chemical Physics*, 117(15):6959–6969, October 2002.
12. J.P. He, H.D. Zhang, J.M. Chen, and Y.L. Yang. Monte carlo simulation of kinetics and chain length distributions in living free-radical polymerization. *Macromolecules*, 30(25):8010–8018, 1997.
13. J.A.M. Janssen. The elimination of fast variables in complex chemical reactions: Mesoscopic level reducible case. *J. of Statistical Physics*, 57:187–198, 1989.
14. Mads Kaern, William J. Blake, and J.J. Collins. The engineering of gene regulatory networks. *Annu. Rev. Biomed. Eng.*, 5:179–206, 2003.
15. Jason Kastner, Jerry Solomon, and Scott Fraser. Modeling a hox gene network in silico using a stochastic simulation algorithm. *Developmental Biology, 246, Elsevier Sci.*, 246:122–131, 2002.
16. G.T. Kurtz. Approximation of discontinuous processes by continuous processes. In *Stochastic Nonlinear Systems in Physics, Chemistry and Biology*, Berlin, 1981. Springer-Verlag.
17. P.A.W. Lewis and G.S. Shedler. Simulation methods for poisson processes in nonstationary systems. In *Proc. of the 10th conference on Winter simulation*, volume 1, pages 155–163, Miami Beach, FL, 1978.
18. S. Ostergaard, Olsson L., M. Johnston, and J. Nielsen. Increasing galactose consumption by saccharomyces cerevisiae through metabolic engineering of the gal gene regulatory network. *Natl. Biotechnical J.*, 18(12):1283–1286, Dec 2000.
19. J. Peccoud and Kent Vander Velden. Molecular networks: Structure, dynamics and applications. In *Proc. of the 10th Intl. Workshop on Petri Nets and Performance Models*, pages 2–8, Urbana, IL, September 2003.
20. William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd edition, October 1992.
21. C. Rao and A. Arkin. Stochastic chemical kinetics and the quasi-steady state assumption: Applications to the gillespie algorithm. *J. Of Chemical Physics*, 118(4999–5010), March 2003.
22. H.G. Solari and M.A. Natiello. Poisson approximation to density dependent stochastic processes: A numerical implementation and test. In *Proc. of the Workshop: Dynamical Systems from Number Theory to Probability-II*, volume 6, pages 79–94. Växjö Univ. Press. Ser. Math. Modelling, 2003.
23. H.G. Solari and M.A. Natiello. Stochastic population dynamics: the poisson approximation. *Phy. Rev. E*, 67:031918, 2003.
24. Ideker T., Thorsson V. Ranish J.A., Christmas R., Buhler J., Eng J.K., Bumgarner R., Goodlett D.R., Aebersold R., and Hood L. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Sci.*, 292:929–934, 2001.
25. A. P. A. van Moorsel and W. H. Sanders. Adaptive uniformization. *ORSA Communications in Statistics: Stochastic Models*, 10(3):619–648, August 1994.
26. M.O. Vlad and Pop A. A physical interpretation of age-dependent master equations. *Physica A*, 155:276–310, 1989.