Proposal: Collaboration between Univ. of Minnesota and the IBM-Rochester
Area of Interest: **Stochastic Simulation of Biochemistry** on Blue Gene.
P.I.: Marc D. Riedel, *Assistant Professor*, *ECE*, *Univ. of Minnesota*

Tackling the Stochastic Simulation of Biochemical Networks with Real Computing Power

Cell biology traditionally has been a descriptive science, focusing on detailed, yet *qualitative*, studies of the molecular machinery of cells. Increasingly, it is striving to become an exact science, modeling cellular processes *quantitatively.* In principle, such models should be amenable to computer analysis and simulation. Indeed, given a complete model of a cell (in the form of a set of biochemical reactions) and its current state (defined by the quantities of the constituent molecules present), as well as the conditions of the external environment, one should be able to predict precisely how it develops over time.

Such a comprehensive simulation of a whole cell remains a distant and elusive goal [51]. Cell models are far from complete; much work remains for biologists in elucidating all the complex molecular mechanisms. However, for some sub-systems – for instance, for specific signaling pathways – detailed models exist. Here the baton is handed off to the computer scientists: analyzing and simulating such models presents a formidable computational challenge.

The models often consist of hundreds of reactions operating on hundreds of types of molecules. Consider the mating response of Baker's yeast (*Saccharomyces cerevisiae*), among the most widely-studied and best-understood biochemical pathways [24]. It describes the intricate chain of biochemical events triggered in a yeast cell when it detects the presence of pheromone molecules produced by other yeast cells in its vicinity. The outcome is a fateful decision for the yeast cell: to mate or not to mate. A state-of-the-art model for this pathway – the culmination of decades of experimental investigations by dozens of labs – consists of 97 reactions operating on approximately 276 types of molecules [49]. Other complex models include those for viral infection mechanisms [3] and the reproductive life cycle of bacteria [47].

Existing approaches to analyzing such models rely upon detailed Monte Carlo simulations that track the minutiae of the reactions that are executing [19]. Such simulations can be very lengthy since there are a multitude of reactions happening nearly in *parallel* and these must all be executed *serially.* Each trial might consist of millions of reaction events; thousands of such trials must be performed in order to get an accurate estimate. (In order to reduce the uncertainty of a given Monte Carlo run by a factor of $1/f$, one has to run roughly f^2 as many trials.) Not surprisingly, biologists are clamoring for more powerful computers for their simulations [32].

And yet, Monte Carlo is generally a method of last resort for physical simulations. It it used to compute an *implicit* estimate of the solution when an *explicit* calculation is intractable. A recurring theme in this project is the application of **analytical** techniques – as opposed to blind simulation – in order to characterize the behavior of biochemical reactions. Instead of seeking answers in raw simulation data, we tailor the analysis to the questions at hand.

Although designed by evolution, cellular processes have remarkable parallels to computing systems engineered by humans. They exhibit modular designs; the modules have well-defined biochemical inputs and outputs; and there are elaborate signaling protocols, with synchronization and error-correction mechanisms. Clearly, the knowledge accumulated over the decades in computer engineering may provide valuable insights into the workings of such systems.

What sort of specific expertise can circuit designers bring to the table? One of the great successes of the integrated circuit design community has been in *abstracting* and *scaling* the design problem. The physical behavior of transistors is understood in terms of differential equations – say, with models found in tools such as SPICE [37]. However, the design of integrated circuits proceeds at a more abstract level – in terms of gates, modules, and components.

In this project, we will explore the concepts of modularization and abstraction in the context of biochemical systems. Specific algorithmic expertise from circuit design can be brought to bear on the problem: state-space exploration through probabilistic analysis; reachability analysis with symbolic techniques and decision diagrams; and characterizing reaction dynamics with information-theoretic measures. This informed approach to analysis, in turn, opens up the possibility of **synthesis**: engineering biochemical reactions to produce specific outputs in response to different combinations of inputs.

1 Analysis of Biochemical Reactions

A variety of approaches have been proposed for the analysis of cellular processes. The simplest of these, from a computational standpoint, is *macroscopic*-level modeling that treats molecular quantities in terms of simple "on/off" activation levels [25][28][31][48]. Research in this vein includes the application of inference techniques such as Bayesian Networks [12][38]. Unfortunately, such modeling lacks a quantitative foundation; furthermore, it often fails to capture important aspects of the system behavior.

Another approach has been to model cellular systems with ordinary, coupled differential equations. This approach assumes that molecular concentrations are *continuous* quantities that vary *deterministically* over time [22][52]. Typical cellular systems, viewed at a molecular level, break this assumption. The molecules involved are generally large and complex (e.g., proteins and enzymes). The quantities that are involved are surprisingly small: on the order of tens, hundreds, or thousands of molecules of each type. At this scale, individual reactions matter [20].

1.1 Discrete Events

In this project, we advocate analysis in a setting that is familiar to the electrical and computer engineering community: **discrete events** operating on a set of **finite states**. We describe the state of the system in terms of the molecular quantities measured in *whole* (i.e., non-negative integer) amounts. State transitions occur as discrete events when biochemical reactions fire.

Example 1 Consider a system with three types of molecules X_1, X_2, and X_3. The **state** of the system is described by

$$[x_1, x_2, x_3],$$

where x_1, x_2, and x_3 are *integer* variables, assuming non-negative values corresponding to the number of molecules of types X_1, X_2, and X_3, respectively. For instance, the system might be in the state $[3, 3, 3]$ with three molecules of each type.

Consider the three reactions:

$$\begin{aligned} R_1: &\quad X_1 + X_2 \rightarrow 3X_3 \\ R_2: &\quad X_1 + 2X_3 \rightarrow 3X_2 \\ R_3: &\quad 2X_2 + X_3 \rightarrow 2X_1 \end{aligned}$$

The types that are consumed are referred to as the *reactants*, whereas those that are created are referred to as the *products*. Note that these reactions are *coupled*: the types appear both as reactants and products in different reactions.

Suppose that the system is in the state $[3, 3, 3]$ and reaction R_1 fires. One molecule of type X_1 and one of type X_2 are consumed; three of type X_3 are produced. This results in the state transition:

$$[3, 3, 3] \xrightarrow{R_1} [2, 2, 6].$$

As reactions fire, a cellular process follows a sequence of such transitions. Figure 1 illustrates the trajectory taken from the state $[3, 3, 3]$ by the sequence R_1, R_2, and R_3. (Note that the system *returns* to the state $[3, 3, 3]$ in this case.) □

1.2 Markov Chains

Fixing environmental variables, such as temperature and external chemical gradients, we can assume that a cellular system behaves as a **Markov process**: The probability of future events depends only on the present state, not on the past sequence of events. Indeed, at each point in time, the probability of a given reaction occurring is a function of the current state only. It is proportional to the quantity of the reactants present as well as a *rate constant.* The latter is a real-valued parameter that is either deduced from biochemical principles or measured experimentally [22].

More precisely, consider a system consisting of M types of molecules $X_1, \ldots, X_M$, interacting through N reactions $R_1, \ldots, R_N$. For a reaction R_j let Q_j be the set of indices of the reactant types. With coefficients $q_1, \ldots, q_P$ for the reactant types, let

$$r_j = k_j \sum_{i \in Q_j} \binom{x_i}{q_i},$$

where x_i is the number of molecules of type X_i, k_j is the rate constant, and $\binom{x_i}{q_i}$ denotes the binomial coefficient. If any $x_i < q_i$, (i.e., there are insufficient molecules of a reactant for the reaction to proceed), then set $r_j = 0$. Now

$$p_j = \frac{r_j}{\displaystyle\sum_{k=1}^{N} r_k}$$

gives the probability that reaction R_j is the next one to fire, $j = 1, \ldots, N$.

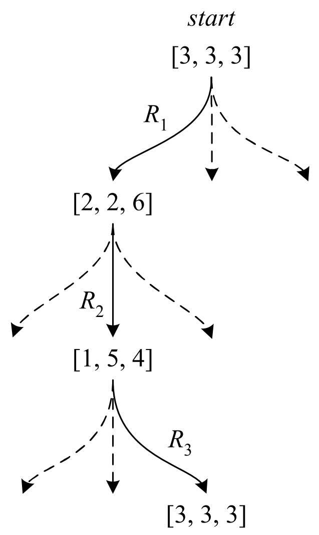


Figure 1: Biochemical reactions as discrete events. Beginning from the state $[3, 3, 3]$, R_1 fires, followed by R_2, followed by R_3.

Example 2 For the reactions in Example 1, let the state be $\mathbf{S} = [x_1, x_2, x_3]$. The firing probabilities for R_1, R_2, and R_3 are computed as follows:

$$\begin{aligned}
p_1(x_1, x_2, x_3) &\equiv \frac{\frac{1}{2}x_1(x_1-1)x_2}{\frac{1}{2}x_1(x_1-1)x_2 + x_1x_3(x_3-1) + 3x_2x_3}, \\
p_2(x_1, x_2, x_3) &\equiv \frac{x_1x_3(x_3-1)}{\frac{1}{2}x_1(x_1-1)x_2 + x_1x_3(x_3-1) + 3x_2x_3}, \\
p_3(x_1, x_2, x_3) &\equiv \frac{3x_2x_3}{\frac{1}{2}x_1(x_1-1)x_2 + x_1x_3(x_3-1) + 3x_2x_3},
\end{aligned}$$

where x_1, x_2 and x_3 denote the numbers of molecules of types $X_1, X_2,$, and X_3, respectively. Suppose that $\mathbf{S} = [3, 3, 3]$. Then the firing probabilities for R_1, R_2, and R_3 are

$$p_1(3,3,3) \equiv \frac{9}{9+18+27} = \frac{1}{6}, \qquad p_2(3,3,3) \equiv \frac{18}{9+18+27} = \frac{1}{3}, \qquad p_3(3,3,3) \equiv \frac{27}{9+18+27} = \frac{1}{2},$$

respectively. □

1.3 Biological Outcomes

Randomness is inherent to all forms of biochemical computation: at any given instant, the choice of which reaction fires next is a matter of chance. Certain biochemical systems appear to exploit this randomness for evolutionary

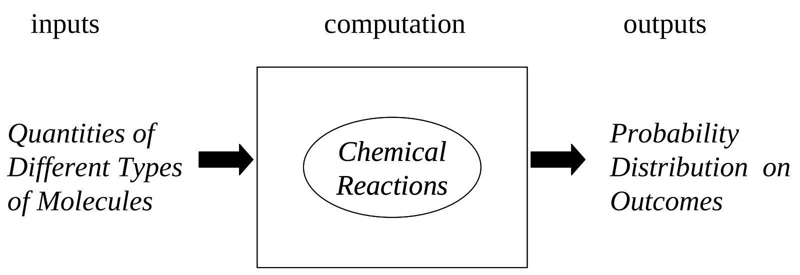

inputs
computation
outputs
Quantities of Different Types of Molecules
Chemical Reactions
Probability Distribution on Outcomes

Figure 2: Framework for biochemical computation.

advantage, choosing between different outcomes with a probability distribution – in effect, hedging their bets with a portfolio of responses that is carefully tuned to the environmental conditions [39].

For instance, the *lambda* bacteriophage, a virus that infects the *E. coli* bacteria, chooses one of two survival strategies: either it integrates its genetic material with that of its host and then replicates when the bacterium divides (call this "stealth" mode); or else it manipulates the molecular machinery of its host to make many copies of itself, killing the bacterium in the process, and thereby releasing its progeny into the environment (call this the "hijack" strategy). The choice of which strategy to pursue, while based on environmental inputs, is probabilistic: in some cases, the virus chooses the first strategy, say with probability 0.33, and the second with probability 0.67, while in other cases the probabilities are reversed [35]. Clearly the virus is hedging its bets, an approach that provides significant advantages in an evolutionary context. Other examples include the *pap pili* epigenetic response of bacteria [23] and the lentiviral positive-feedback loop in the HIV virus [53].

Most existing methods that model cellular processes probabilistically focus on time-ordered sequences and equilibrium conditions [15]. In this project, we advocate a framework for analysis that characterizes the **probability distribution** of biological outcomes. Such outcomes are indicated by thresholds in certain molecular quantities. For instance, the decision of the lambda virus is indicated by thresholds on two of its constituent types, Cro and cII [3]: the decision to go into "stealth" mode is indicated by $Cro > 55$, while the decision to go into "hijack" mode by $cII > 145$. These two conditions are mutually exclusive; however, this need not be the case in general.

Our framework for biochemical computation is summarized in Figure 2. We note that the outcomes need not be simple threshold conditions; they can be arbitrarily complex *logical* functions defined on the state space.

2 Stochastic Simulation

A stochastic treatment of chemical reactions was first discussed by Gillespie in 1977 [19]. He proposed *Monte Carlo* simulation: Beginning from an initial state, reactions are chosen at random, based on propensity calculations. As reactions fire, the quantities of the different species change by integer amounts. Repeated trials are performed and the probability distribution of different outcomes is estimated by averaging the results.

The drawback of Gillespie's stochastic simulation algorithm (SSA), as it has become known, is the amount of computation required. Although the simulation does not track the spatial location of individual molecules, it executes each and every reaction that occurs, updating the quantities of species present. At each step, the choice of which reaction occurs next entails a probability calculation as well as the generation of a random number. Each trial consists of a long sequence of reactions; many such trials must be performed in order to obtain an accurate estimate [9]. This adds up to significant computation time [32] [33].

Gibson and Bruck proposed algorithmic improvements to Gillespie's SSA [13][14]. Their method achieves significant speedups by structuring the computation through prioritized data structures and by using random numbers parsimoniously. Also, several methods have been proposed to expedite stochastic simulation through approximations. Gillespie proposed a technique called "tau-leaping" [16][17]; the technique was analyzed and refined in numerous follow-up papers [6][7][8][41][50]. Other approximate techniques include the *partial equilibrium* assumption [9] [40] and *quasi steady-state* analysis [40]. Unfortunately, such approximations are not always applicable, particularly for sensitive segments of the simulation where single-molecule events can affect the outcome; furthermore, the resulting errors are generally difficult to quantify.

3 Algorithmic Optimizations

In preliminary work, we proposed a new framework for stochastic simulation based on localized probabilistic analysis of the state space as well as aggressive caching of probability calculations [42]. The method is *exact*, sampling trajectories with the same probability distribution as Gillespie's SSA. The improvement in running time is substantial – sometimes by as much as an order of magnitude.

3.1 Cycle Leaping

Most biochemical systems of interest contain coupled reactions – that is to say, some of the species appear both as reactants and as products in different reactions. Also, most contain reversible reactions. Indeed, systems with these properties exhibit the most interesting dynamics. However, as a result, stochastic simulation gets mired in loops, visiting the same sequence of states repeatedly. (Such a cycle was illustrated in the trajectory shown in Figure 1.) Cycles are not surprising, as the simulation is tracking the minutiae of the physical behavior. And yet, computing time is frittered away as the calculations of the propensities and random numbers are replayed.

In our algorithm, whenever a cycle is encountered, the exit probabilities are computed and the simulation leaps directly to one of the exit states, a technique that we call **cycle leaping**. As trajectories are formed, a history of the states is recorded. When a cycle is encountered, the exit probabilities are computed. Then, based upon a single random number, the simulation leaps directly to one of the exit states, as illustrated in Figure 3.

It should be noted that there is overhead in applying cycle leaping:

- A history of the states must be maintained. For each state that is visited, a check must be performed to see if it is in the history.
- When a cycle is detected, the localized probability calculations must be performed.

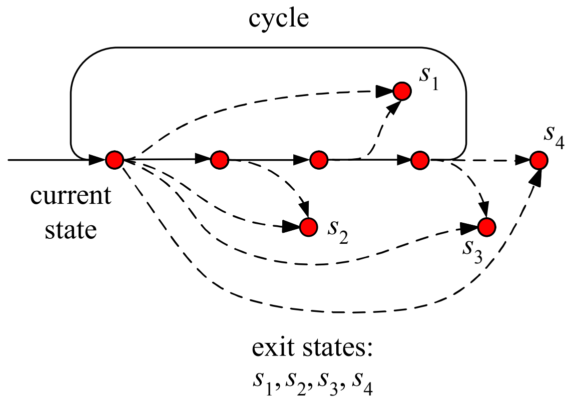


Figure 3: Cycle Leaping

In practice, this overhead is minimal. One only targets small cycles, say 100 states in length. Accordingly, the history is a small sliding window. All probability calculations are linear in the length of the cycle and linear in the number of transitions per state (to the exit states). The trajectories in many models spend 99% or more of their time in loops. With cycle leaping, the trajectories are shortened to 1% or less of their original length, and so the overhead is easily justified.

3.2 Event Leaping

For randomized analysis, *pseudo-random* numbers must be generated algorithmically. This is not a trivial task from a computational standpoint. To generate a high-quality pseudo-random number requires between 5 and 20 integer or floating-point operations [27]. This cost often dominates the computation time of stochastic simulation. And yet, in existing algorithms, much of the effort in generating randomness is wasted.

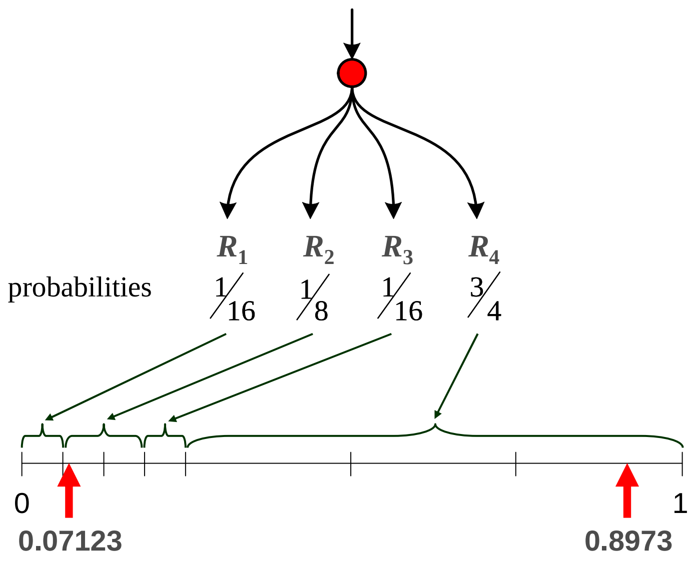


Figure 4: Sampling a probability distribution.

Consider the example in Figure 4. In a given state, there are four possible reactions with the probabilities shown. To sample this distribution, we generate a random number between 0 and 1. Say we generate 0.07123; accordingly we would choose reaction R_2 (shown on the left-hand side). If we generate 0.8973, then we would choose R_4 (shown on the right-hand side). In both cases, but particularly the latter, we can argue that the random numbers generated are overly precise. Any number between 0.25 and 1 would have resulted in the choice of R_4. Indeed, unless the probability distribution is perfectly even – with all reactions equally likely – then the precision of the random numbers cannot be matched perfectly to the distribution. Implementations generally used standard library calls for pseudo-random generation. Most of these do not offer flexibility in the precision of the numbers that are generated.

In our approach, we tune the distribution to the randomness. Note that for sequences of several reactions, the probabilities are *multiplicative*. For instance, to compute the probability of a sequence of reactions R_1, R_2 through states S_1, S_2, we would simply multiply the probability of R_1 occurring from S_1 by the probability of R_2 occurring from S_2. If two different sequences merge to the same state, then the probabilities are *additive*.

By analyzing the probabilities of reaction sequences, the distribution can be leveled. The sequences are extended – most probable first – until their probabilities drop below a threshold that matches the precision of the pseudo-random numbers. Then, based on a single pseudo-random number, the simulation leaps directly to one of the next states.

With event leaping, multiple reactions are, in effect, executed in a single step, shortening the trajectories and resulting in more efficient utilization of random numbers. Of course, there is a penalty to be paid: the analysis likely will explore reaction sequences that will not occur in the trajectory. This penalty is offset by the savings in the computation of random numbers. Furthermore, it is minimized if caching is exploited.

4 Probabilistic Analysis and Caching

In this project, we will develop an *analytic framework* for the probabilistic exploration of the state spaces of biochemical systems. Instead of generating trajectories at random, our approach will be to analyze the state space through a branch-and-bound search. This is illustrated in Figure 5: the state space for the reactions in Example 1 is explored, beginning at the state $[3, 3, 3]$. For the initial state and each subsequent state that is visited, separate branches are formed for each possible reaction. (Again, note that along each branch, the probabilities are multiplicative; when two branches merge, the probabilities are additive.)

The search is neither strictly *depth*-wise, nor strictly *breadth*-wise. Rather, at each step, the highest probability branch is explored next. The search terminates when the sum of the probabilities of the unexplored branches is less than the desired amount of uncertainty. We expect this approach to expedite the analysis significantly. More important, since each branch is labeled with a probability value, the analysis provides highly structured information about the state space. Through informed queries on the analysis results, one can study aspects of the system such as its boundary states and bifurcation points.

From this simple example, it is apparent that even short reaction sequences can sprawl over large regions of the state space. If the analysis is likely to revisit the same portion of the state space, then it is judicious to record the probability calculations and later retrieve them, if needed. We perform this caching not only for each trajectory, but also across successive trajectories. As larger and larger swathes of the state space become known, longer and longer leaps are made. Of course, the overhead of such caching can be considerable; accordingly, the data structures must be finely tuned. If too much information is cached, the burden of indexing it and retrieving it can outweigh the cost of recalculating it. Nevertheless, we have found that this approach, combined with the cycle and event leaping techniques discussed earlier, provides significant improvements in the running time.

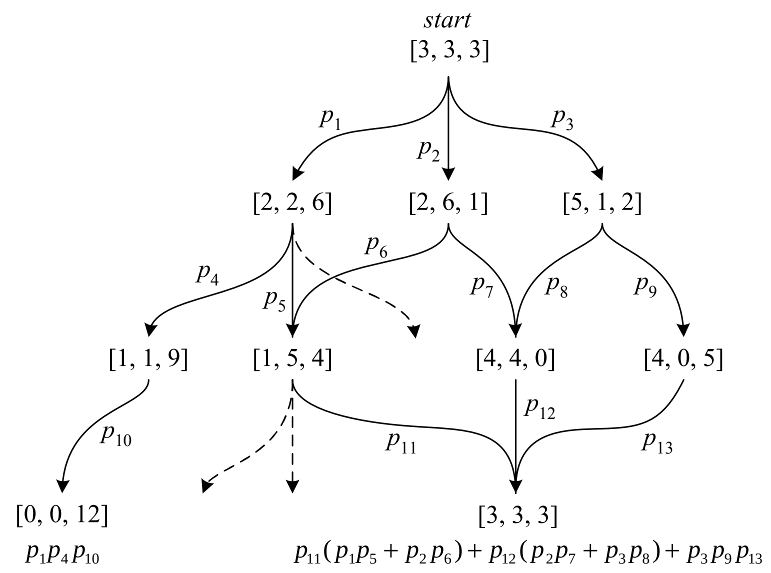


Figure 5: Probabilistic analysis of the state space for the reactions in Example 1.

4.1 Reachability Analysis

Symbolic techniques have enjoyed huge successes for the problem of reachability analysis in sequential circuit verification. Using BDDs, it is routinely possible to characterize a space with quadrillions of states [5]. For cellular processes, we will perform reachability analysis with *multi-valued* decision diagrams as our data structure.

Example 3 Consider a system with types A, B, and C and biochemical reactions

$$\begin{aligned} R_1: &\quad A + 2B \rightarrow 2C, \\ R_2: &\quad B + \ \ C \rightarrow 2A, \\ R_3: &\quad A + \ \ C \rightarrow 2B. \end{aligned}$$

Suppose that, initially, this system can be in any of three states

$$S_1 = [4, 7, 5], \quad S_2 = [3, 5, 7], \quad \text{and} \quad S_3 = [3, 4, 5].$$

We represent this collection of *reachable* states with a multi-terminal decision diagram, shown on the left in Figure 6. Here the nodes are labeled with the types and the edges with the corresponding numbers of molecules. A path from the source to the "Yes" sink specifies a reachable state.

After one of the three reactions fires, we can be in any of the states depicted in the diagram on the right in Figure 6. □

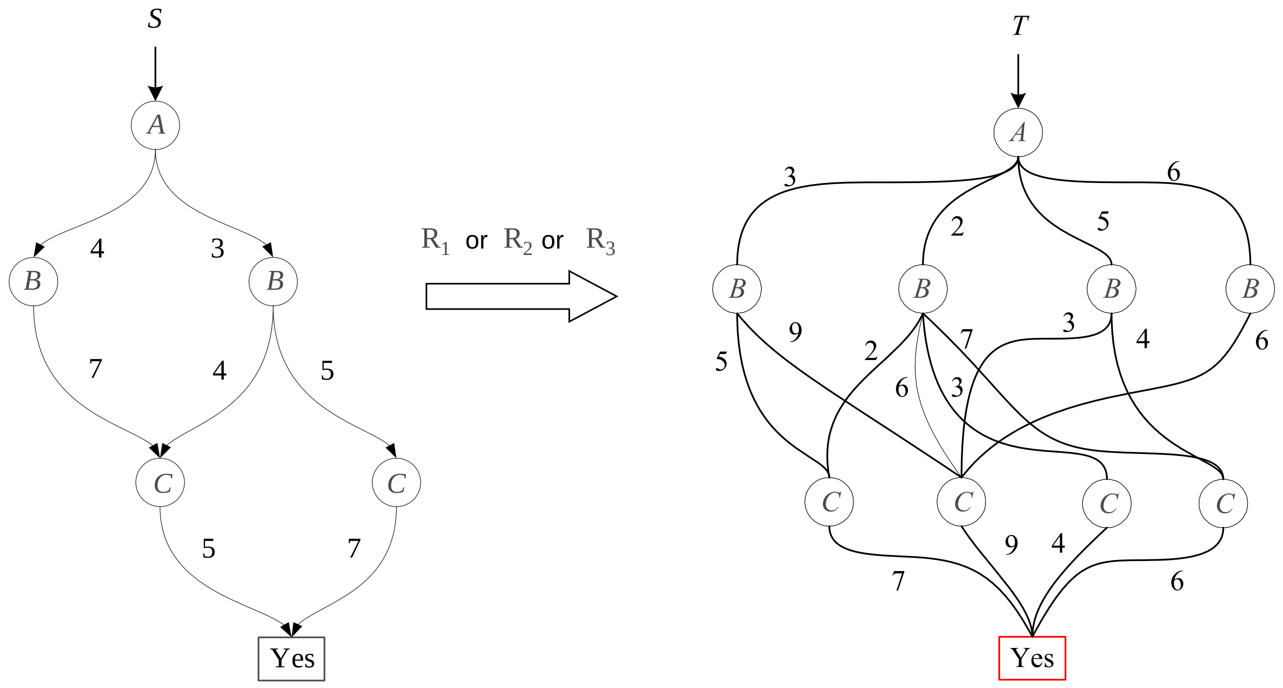


Figure 6: On the left, the set of initial reachable states. On the right, the set of reachable states after one of the reactions R_1, R_2, or R_3 fires.

As with binary decision diagrams, multi-terminal decision diagrams can be manipulated efficiently: logical operations such as the "OR" performed in Example 3 can be performed in an amount of time that is linear in the size of the diagrams. Of course, the diagrams can grow to be very large. Nevertheless, the approach offers the possibility of answering "yes/no" without performing exhaustive simulation. In essence, the algorithm tracks the evolution of all reachable states *in parallel* through the data structures.

5 Information-Theoretic Measures

The narrow task of computational analysis is to *validate* a model: Does it accurately (or even approximately) implement the behavior that is observed experimentally? The broader task is to extract higher-level insights regarding the dynamics: What chain of events accounts for a specific outcome? How sensitive is the model to variations in the parameters? How robust is it to perturbations in the inputs or from the environment?

While Monte Carlo simulations faithfully reproduce the physical behavior of a system, they are computationally prohibitive. Furthermore, by averaging the results, such simulations produce only coarse-grained statistics that provide little insight into the dynamics of the stochastic behavior. For instance, in a system with a bimodal response (say "stealth" vs. "hijack"), we can expect each trajectory eventually to veer in one direction or the other. Is this decision gradual or abrupt? If it is gradual, then the behavior might be due to straight-forward stochastic competition; if it is abrupt, then there may be interesting dynamics at play.

More precisely, from a given starting state, how does the uncertainty in the outcome change as a function of time (or the number of steps)? For instance, if the model predicts a probability distribution of 0.20/0.80 for a "yes/no" outcome, when is this "decision" made? In one extreme, it is made right at the outset: from the initial state, the probability of one reaction is 0.20, and that of another is 0.80. Depending on which reaction is chosen, the fate is sealed; from that point onward, the probability of the corresponding outcome is 1. At the opposite extreme, nothing is decided until the very end: the probability distribution remains 0.20/0.80 until the final step when one outcome or the other is chosen. Of course, real systems will fall somewhere between these two extremes.

In this project, we will explore metrics for characterizing the state space. In preliminary work, we have described a conceptually simple, yet effective, metric: the *probability gradient* [11]. For each point in the state space, the probability gradient is a vector corresponding to the *expected value* of transitions. More precisely, consider a system consisting of N types of molecules $\{X_1, \ldots, X_N\}$. The state of the system is

$$[x_1, \ldots, x_N] \in \mathbb{N}^N,$$

where x_i is the number of molecules of type X_i. Suppose that, from a given state $\vec{S}$, there are M possible transitions. Suppose that the j-th transition occurs with probability p_j and is characterized by the vector

$$\vec{V_j} = [v_{1,j}, \ldots, v_{N,j}] \in \mathbb{Z}^N,$$

where $v_{i,j}$ is the change in the number of molecules of type X_i that this transition produces. The gradient is computed as

$$\vec{G} = \sum_{j=1}^{M} p_j \vec{V_j}.$$

For the *first-order* gradient, we consider the transitions resulting from a single reaction firing; for the *m-th order* gradient, we consider the transitions resulting from all possible sequences of m reactions firing.

The gradient allows us to characterize the topology of the state space. If the gradient is small in magnitude, then we are likely in a "flat" region with considerable uncertainty in the outcome; if it is large in magnitude, then we are likely in a "canyon" progressing rapidly toward a decision. If the outcome is nearly certain, then a trajectory can be terminated early, reducing the computation time. Furthermore, profiling the probability gradient allows us to detect abrupt changes in the topology – indicating the decision points of the system.

6 Synthetic Biology

Increasingly, biology is becoming a quantitative and exact science, as computer modeling and analysis are applied to cellular chemistry [10]. Further, with the advent of techniques for synthesizing and manipulating the genetic code of living cells, it is striving to become an *engineering* discipline. In the nascent field of *synthetic biology*, researchers aim for "logical" control over biological processes by designing pathways that produce specific outputs in response to different combinations of inputs [4]. This approach has far-reaching implications for domains such as biochemical sensing, drug production, and disease treatment. Already, researchers are boasting of impressive feats of synthetic bio-engineering: cellulosic ethanol [46], anti-malarial drugs [43], and tumor detection [1].

By custom-designing the genetic material of organisms such as yeast or *E. coli* bacteria, it is possible to directly synthesize cellular biochemistry. In principle, genetic engineering can produce biochemical reactions of nearly any form. And yet, designing a set of reactions to implement a desired functionality – efficiently and robustly – is a formidable challenge. Current approaches rely on *ad hoc* designs, validated through stochastic simulation (often performed with massive computing power [32]).

What sort of expertise can the EDA community bring to the table? One of the great successes of the circuit design community has been in *abstracting* and *scaling* the design problem. We suggest that specific algorithmic expertise from circuit design can be brought to bear on the problem of synthesizing biochemical reactions.

6.0.1 Probabilistic Analysis

Certain biochemical systems appear to exploit this randomness for evolutionary advantage, choosing between different outcomes with a probability distribution – in effect, hedging their bets with a portfolio of responses. For instance, the *lambda* bacteriophage, a virus that infects the *E. coli* bacteria, chooses one of two survival strategies: either it integrates its genetic material with that of its host and then replicates when the bacterium divides (call this *stealth* mode); or it manipulates the molecular machinery of its host to make many copies of itself, killing the bacterium in the process, and thereby releasing its progeny into the environment (call this *hijack* mode). The choice of which strategy to pursue is probabilistic, with the probability distribution tuned to the environmental conditions [3].

6.0.2 Engineering a Programmable Portfolio of Responses

While nature readily exploits randomness, bio-engineers have eschewed it – reasoning that synthesizing probabilistic behavior is simply too complex [36]. And yet, a probabilistic response could be crucial for some applications.

Consider the following hypothetical design problem. Suppose that a given pharmaceutical compound functions by eliciting cells in an organ to produce a drug. The compound cannot be injected directly into the cells; the only option is to flood the entire organ with it. Unfortunately, if every cell in the organ responds, then the patient receives too high of a dose of the drug. What is required is that only a predetermined *fraction* of the cells respond. In other words, the compound should elicit a probabilistic response in each cell, tuned according to the requirements. Now consider a generalization of this scenario. Suppose that the goal is to have the mutually exclusive production of different substances according to a probability distribution.

Example 4 For a system with molecular types W_1, W_2, and W_3, suppose that we wish to program the following response,

$$p_1 = 0.3, \qquad p_2 = 0.4, \quad p_3 = 0.3,$$

for W_1, W_2, and W_3 exceeding threshold values, respectively. For each response, we set up *trigger* reactions:

$$T_1 \xrightarrow{1} W_1, \;\; T_2 \xrightarrow{1} W_2, \;\; T_3 \xrightarrow{1} W_3.$$

We initialize the system with quantities of T_1, T_2, and T_3 in the desired ratio of $30 : 40 : 30$. (Should we want a different probability distribution, we simply change the ratio of these initial quantities.)

Given these ratios, the first *trigger* reaction fires first with probability 0.3, the second fires first with probability 0.4, and the third fires first with probability 0.3. We set up *amplification* reactions:

$$T_1 + W_1 \xrightarrow{10^3} 2W_1, \quad T_2 + W_2 \xrightarrow{10^3} 2W_2, \quad T_3 + W_3 \xrightarrow{10^3} 2W_3.$$

Also, we set up *weak annihilation* reactions:

$$\begin{array}{lll} W_1 + T_2 \xrightarrow{10^3} W_1, & W_1 + T_3 \xrightarrow{10^3} W_1, & W_2 + T_1 \xrightarrow{10^3} W_2, \\ W_2 + T_3 \xrightarrow{10^3} W_2, & W_3 + T_1 \xrightarrow{10^3} W_3, & W_3 + T_2 \xrightarrow{10^3} W_3. \end{array}$$

Note that these reactions have much higher rate constants than the *trigger* reactions. Finally, we set up *strong annihilation* reactions:

$$W_1 + W_2 \xrightarrow{10^6} \phi, \quad W_1 + W_3 \xrightarrow{10^6} \phi, \quad W_2 + W_3 \xrightarrow{10^6} \phi.$$

(ϕ indicates that there are no products that we care about for this reaction.) Note that these reactions have still higher rate constants.

It is apparent that, as soon as a *trigger* reaction fires, this choice immediately gets amplified and all other choices inhibited. So, the initial firing probabilities determine the final outcome. □

6.1 General Methodology

We propose a general methodology for synthesizing a probabilistic response, consisting of five categories of reactions. For the mutually exclusive production of molecular types A_i, $i = 1, \ldots, n$, select:

Trigger Reactions

$$\forall\, i : \qquad T_i \xrightarrow{k_i} W_i,$$

Amplification Reactions

$$\forall\, i : \qquad T_i + W_i \xrightarrow{k'_i} 2W_i,$$

Weak Annihilation Reactions

$$\forall\, j \neq i : \qquad W_i + T_j \xrightarrow{k''_{ij}} W_i,$$

Strong Annihilation Reactions

$$\forall\, j \neq i : \qquad W_i + W_j \xrightarrow{k'''_{ij}} \phi,$$

Working Reactions

$$\forall\, i : \qquad F_i + W_i \xrightarrow{k''''_i} W_i + A_i.$$

The rate constants should be selected so that the *trigger* and *working* reactions are the slowest; the *amplification* and *weak annihilation* reactions comparatively much faster; and the *strong annihilation* reactions fastest of all:

$$k_i \approx k_i'''' \quad \ll \quad k_i' \approx k_{ij}'' \quad \ll \quad k_{ij}'''.$$

With this scheme, the first *trigger* reaction to fire determines the outcome. Accordingly, for all i, to obtain type A_i with probability p_i, select the quantities of the trigger molecules $T_1, \ldots, T_n$ according to

$$p_i = \frac{k_i T_i}{\sum_{\forall j} k_j T_j}.$$

Once a *trigger* reaction fires, producing a molecule of W_i, the corresponding *amplification* reaction kicks in, producing more copies of W_i. Also, the corresponding *weak annihilation* reactions kick in, consuming all the competing trigger molecules T_j, $j \neq i$. In the unlikely event that a competing *trigger* reaction happens to fire before this occurs, then the stray competing molecule W_j, $j \neq i$, is quickly consumed by the corresponding *strong annihilation* reaction. Finally, the *working reaction* produces the desired output A_i – at its leisure – according to the amount of food F_i that is supplied.

6.2 Results and Discussion

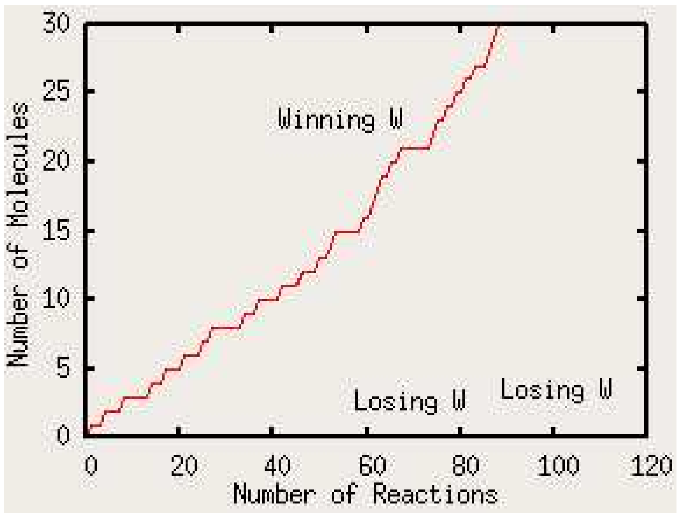
Our scheme can be generalized to implement an arbitrary probability distribution on *logical combinations* of qualitatively different outcomes. It is fully programmable, with the probability distribution specified by the initial quantities of the reactants. It executes nearly independently of the rate constants; only a coarse separation in the magnitudes is required. Although it is beyond the scope of this abstract, an analysis of our scheme shows that it produces the expected probability distribution with vanishingly small error, given such a separation.

While applications in synthetic biology remain (just) over the horizon, the impetus for this work is more immediate: It provides a framework for analyzing and characterizing the stochastic behavior of (natural) biological systems. Consider the *lambda* phage, described by an elaborate set of 75 reactions in 57 molecular types [3]. The figure below compares the behavior of reactions that we synthesized in Example 4, on the left-hand side, to that of the *lambda* phage, on the right-hand side. With our scheme, the winning response is selected unequivocally, nearly from the outset. For the *lambda* phage, the winning response is selected, but with some hesitation. Our analysis indicates why this is so: the *lambda* phage is lacking the *strong annihilation* reactions; as a result, the stochastic competition persists longer.

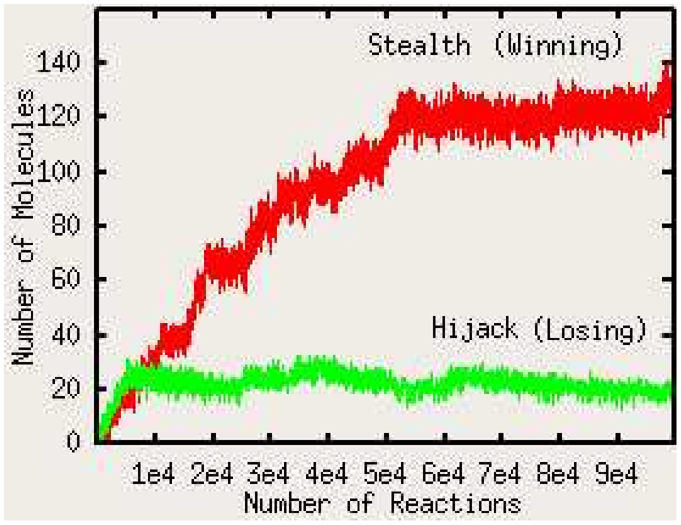
7 Applying Computing Power

At present, computation time severely limits the application of stochastic simulation. For a well-known model of the *lambda phage* virus, consisting of only 75 reactions in 57 molecular types, a full simulation takes several minutes on a high-end workstation. A somewhat larger model for the pheromone-response pathway in diploid yeast, consisting of 96 reactions in 273 types, takes several hours [34]. Tackling significantly larger models, such as those for various metabolic pathways, is well beyond the grasp of current implementations. The computational burden stems from the physical nature of the problem: in a cell, vast numbers of chemical reactions happen nearly simultaneously. Stochastic simulation tracks each and every one of these reactions. If implemented sequentially, each trajectory becomes impossibly long.

In addition to reducing computation time, increased accuracy is often called for. Cellular processes are not always independent of spatial constraints; they sometimes occur on the surface of membranes or in highly confined



Trajectory for Example 4.



Trajectory for the *lambda* phage.

areas, in the presence of chemical gradients. For such models, the assumption of spatial homogeneity is questionable. We are exploring a mesoscopic-level simulation technique that incorporates *reaction-diffusion* gradients. With this approach, parallelization could be applied *spatially*, with the cell volume divided into compartments, each assigned to a different processor. Gradients could be modeled without the computational blow-up that is incurred in tracking individual molecules.

7.1 Parallelization

The problem is clearly ripe for parallelization. Previous work has focused on specific models [45] and specialized hardware for the task [44]. We envisage the development of a general-purpose toolkit for stochastic simulation on a massively-parallel platform such as BlueGene. Parallelization could be accomplished by running random trajectories on individual processors; only loose synchronization would be needed to collect and average the results of the independent trajectories.

We are considering two approaches:

1. Run separate trajectories on each processor. Communicate and aggregate the statistics on a central node.
2. Partition the state space dynamically into regions and assign these to different processors. Hand off trajectories from processor to processor as the simulation moves through the state space.

Due to its simplicity, the first approach may win out in terms of efficiency. However, in running trajectories separately on different processors, we cannot implement caching. While clearly more complicated to implement, the second strategy could leverage caching effectively. (From a research perspective, it presents a number of enticing challenges.)

7.2 Entropy Calculations

We are also considering the application of analytical techniques. For instance, we are exploring information theoretic measures such as *entropy*: from a given starting state, how does the uncertainty in the outcome of a biological process change as a function of time (or the number of steps)? In order to characterize systems, we can perform *multi-phase* stochastic simulation:

- From the initial state, carry the simulation forward x steps.
- Treating the resulting state as the "initial" state, perform y random trials to estimate the probability distribution from this state.
- Repeat all of this z times to characterize the entropy at x steps.
- Sweep x from 1 to n, where n is the maximum number of steps in the simulation.

With multiple phases, the simulation is orders of magnitude more computationally intensive than straight-forward stochastic simulation. Accordingly, it would seem to be a good topic to explore for implementation on Blue Gene. We would also like to explore other information-theoretic measures such as the Kullback-Leibler distance and probability gradients.

7.3 Model Selection

The pheromone-response pathway for yeast is an ideal model to work with both in terms of its complexity and the scientific interest that the results would generate. The pathway describes the intricate chain of biochemical events triggered in a yeast cell when it detects the presence of pheromone molecules produced by other yeast cells in its vicinity. The outcome is a fateful decision for the yeast cell: to mate or not to mate. A state-of-the-art model for this pathway the culmination of decades of experimental investigations in dozens of other labs consists of 97 reactions operating on approximately 276 types of molecules. Other complex models that we are considering include those for viral infection mechanisms and the reproductive life cycle of bacteria.

8 Expected Impact and Long-Term Directions

With an implementation on a massively-parallel machine such as BlueGene, stochastic simulation could be brought to bear on much larger models, in entirely different regimes. Indeed, the ability to simulate systems with thousands of reactions in hundreds of thousands of molecular types could have a tremendous impact in fields ranging from basic biology, to bio-chemical sensing, to pharmaceutical research [32]. Some problems that could be tackled in the short term:

- Simulation of a whole-cell model of E. coli [51];
- Simulation of an HIV-1 model vector [2];
- Simulation of a bacterial pathway for the production of an anti-malarial drug [26].

In the longer term, we plan to tackle the broader challenge of design. Moving beyond analysis, will it be possible to engineer a form of "logical" control in biochemical networks by designing pathways that produce specific outputs in response to different combinations of inputs? Needless to say, success in this endeavor would alter the landscape of biological and medical research. Of course, formidable computing power will be needed to explore the design spaces and perform such "in silico" synthesis trials.

References

[1] J. Anderson et al., "Environmentally Controlled Invasion of Cancer Cells by Engineered Bacteria," *Journal of Molecular Biology*, Vol. 355, No. 4, pp. 619–627, 2006.

[2] L. Weinberger, J. Burnett, J. Toettcher, A. Arkin, and D. Schaffer, "Stochastic Gene Expression in a Lentiviral Positive-Feedback Loop: HIV-1 Tat Fluctuations Drive Phenotypic Diversity," *Cell*, Vol. 122, 2005.

[3] A. Arkin et al., "Stochastic Kinetic Analysis of Developmental Pathway Bifurcation in Phage Lambda-Infected E. Coli Cells," *Genetics*, Vol. 149, No. 1633, 1998.

[4] Y. Benenson et al., "An Autonomous Molecular Computer for Logical Control of Gene Expression," *Nature*, Vol. 429, pp. 423–429, 2004.

[5] J. Burch, E. Clarke, K. McMillan, and D. Dill, "Sequential Circuit Verification Using Symbolic Model Checking," *Design Automation Conference*, 1990.

[6] Y. Cao, H. Li, and L. Petzold, "Efficient Formulation of the Stochastic Simulation Algorithm for Chemically Reacting Systems," *Journal of Chemical Physics*, Vol. 121, No. 9, pp. 4059–4067, 2004.

[7] Y. Cao, L. Petzold, M. Rathinam, and D. Gillespie, "The Numerical Stability of Leaping Methods for Stochastic Simulation of Chemically Reacting Systems," *Journal of Chemical Physics*, Vol. 121, No. 24, pp. 12169–12178, 2004.

[8] Y. Cao, D. Gillespie and L. Petzold, "Multiscale Stochastic Simulation Algorithm with Stochastic Partial Equilibrium Assumption for Chemically Reacting Systems," *Journal of Computational Physics*, Vol. 206, No. 2, 2005.

[9] Y. Cao and L. Petzold, "Accuracy Limitations and the Measurement of Errors in the Stochastic Simulation of Chemically Reacting Systems," *Journal of Computational Physics*, Vol. 212, No. 1, pp. 6–24, 2006.

[10] D. Endy and R. Brent,"Modelling cellular behaviour," *Nature*, Vol. 409, pp. 391–395, 2001.

[11] B. Fett and M. Riedel, "Characterizing the Decisions of Biochemical Systems with Probability Gradients," to appear in *International Conference on Systems Biology*, Dec. 2006.

[12] N. Friedman, "Inferring Cellular Networks Using Probabilistic Graphical Models," *Science*, Vol. 303, 2004.

[13] M. Gibson and J. Bruck, "Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels," *Journal of Physical Chemistry A*, No. 104, 2000.

[14] M. Gibson, "Computational Methods for Stochastic Biological Systems," *Ph.D. Dissertation*, California Institute of Technology, 2001.

[15] D. Gillespie, "The Chemical Langevin Equation," *Journal of Chemical Physics*, Vol. 113, No. 1, 2000.

[16] D. Gillespie, "Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems," *Journal of Chemical Physics*, Vol. 115, No. 4, 2001.

[17] D. Gillespie and L. Petzold, "Improved Leap-Size Selection for Accelerated Stochastic Simulation," *Journal of Chemical Physics*, Vol. 119, No. 16, 2003.

[18] D. Gillespie, "Stochastic Chemical Kinetics," *Handbook of Materials Modeling*, S. Yip, ed., Springer, pp. 1735–1752, 2005.

[19] D. Gillespie, "Exact Stochastic Simulation of Coupled Chemical Reactions," *Journal of Physical Chemistry*, No. 81, No. 25, 1977.

[20] D. Gillespie, "A Rigorous Derivation of the Chemical Master Equation," *Physica A*, Vol. 188, No. 1-3, 1992.

[21] C. Goble, S. Pettifer, R. Stevens, and C. Greenhalgh, "Knowledge Integration: In silico Experiments in Bioinformatics," in *The Grid: Blueprint for a New Computing Infrastructure*, 2nd Edition, eds. Ian Foster and Carl Kesselman, Morgan Kaufman, 2003.

[22] R. Heinrich and S. Shuster, "The Regulation of Cellular Systems," Chapman and Hall, 1996.

[23] A. Hernday, B. Braaten, and D. Low, "The Intricate Workings of a Bacterial Epigenetic Switch," *Advances in Experimental Medicine and Biology*, Vol. 547, No. 83-9, 2004.

[24] I. Herskowitz, "Life Cycle of the Budding Yeast *Saccharomyces cerevisiae*," *Microbiological Reviews*, No. 52, 1988.

[25] S. Kauffman, C Peterson, B. Samuelsson, and C. Troein, "Random Boolean Network Models and the Yeast Transcriptional Network," *Proceedings of the National Academy of Sciences*, Vol. 100, Dec. 2003.

[26] I. Chang, E. Gilbert, N. Eliashberg, and J. Keasling. "A Three-Dimensional, Stochastic Simulation of Biofilm Growth and Transport-related Factors that Affect Structure," *Microbiology*, Vol .149, 2003

[27] D. Knuth, "The Art of Computer Programming, Vol. 2: Seminumerical Algorithms," Addison-Wesley, 1997.

[28] H. Kuwahara, C. Myers, N. Barker, M. Samoilov, and A. Arkin, "Automated Abstraction Methodology for Genetic Regulatory Networks," to appear in *Transactions on Computational Systems Biology.*

[29] T. Larrabee, "Test Pattern Generation Using Boolean Satisfiability," *IEEE Transitions on Computer-Aided Design*, Vol. 11, No. 1, 1992.

[30] C. Lee, "Representation of Switching Circuits by Binary-Decision Programs," *Bell System Technical Journal*, Vol. 38, 1959.

[31] L. Li and H. Lu "Explore Biological Pathways from Noisy Array Data by Directed Acyclic Boolean Networks," *Journal of Computational Biology*, Vol. 12, No. 2, 2005.

[32] L. Lok, "The Need For Speed in Stochastic Simulation," *Nature Biotechnology*, Vol. 22, No. 8, 2004.

[33] L. Lok and R. Brent, "Automatic Generation of Cellular Reaction Networks with Moleculizer 1.0,"*Nature Biotechnology*, Vol. 23, 2005.

[34] L. Lok and R. Brent, "Automatic Generation of Cellular Reaction Networks with Moleculizer 1.0," *Nature Biotechnology*, Vol. 23, 2005.

[35] H. McAdams and A. Arkin, "Stochastic Mechanisms in Gene Expression," *Proceedings of the National Academy of Sciences*, Vol. 94, No. 3, 1997.

[36] R. McDaniel and R. Weiss, "Advances in Synthetic Biology: on the Path from Prototypes to Applications," *Current Opinion in Biotechnology*, Vol. 16, pp. 476-483, 2005.

[37] L. Nagel and D. Pederson, "Simulation Program with Integrated Circuit Emphasis," *Midwest Symposium on Circuit Theory*, 1973.

[38] C. Needham, J. Bradford, A. Bulpitt, and D. Westhead, "Inference in Bayesian Networks," *Nature Biotechnology*, Vol. 24, 2006.

[39] C. Rao, D. Wolf, and A. Arkin, "Control, Exploitation and Tolerance of Intracellular Noise," *Nature*, Vol. 420, No. 6912, pp. 231–237, 2002.

[40] C. Rao and A. Arkin, "Stochastic Chemical Kinetics and the Quasi-Steady-State Assumption: Application to the Gillespie Algorithm," *Journal of Chemical Physics*, Vol. 118, No. 11, pp. 4999–5010, 2003.

[41] M. Rathinam, L. Petzold, Y. Cao, and D. Gillespie, "Stiffness in Stochastic Chemically Reacting Systems: The Implicit Tau-leaping Method," *Journal of Chemical Physics*, Vol. 119, No. 24, pp. 12784–12794, 2003.

[42] M. Riedel and J. Bruck, "Exact Stochastic Simulation with Event Leaping," *International Conference on Systems Biology*, Oct. 2005.

[43] D.-K. Ro et al., "Production of the antimalarial drug precursor artemisinic acid in engineered yeast," *Nature*, Vol. 440, pp. 940–943, 2006.

[44] L. Salwinski and D. Eisenberg, "In Silico Simulation of Biological Network Dynamics," *Nature Biotechnology*, Vol. 22, No. 8, 2004.

[45] M. Schwehm, "Parallel Stochastic Simulation of Whole-Cell Models," *International Conference on Systems Biology*, 2001.

[46] M. Sedlak and N. Ho, "Production of Ethanol from Cellulosic Biomass Hydrolysate Using Genetically Engineered *Saccharomyces* Yeast Capable of Co-Fermenting Glucose and Xylose," *Applied Biochemistry & Biotechnology* Vol. 113, No. 116, pp. 403–416, 2004.

[47] C. Shaffer, N. Ramakrishnan, M. Vass, and L. Watson, "Improving the Development Process for Eukaryotic Cell Cycle Models with a Modeling Support Environment," *Simulation*, Vol. 79, No. 12, 2003.

[48] I. Shmulevich, and S. Kauffman, "Activities and Sensitivities in Boolean Network Models," *Physical Review Letters*, Vol. 93, No. 4, 2004.

[49] T. Thomson, *Personal Communications*, 2005.

[50] T. Tian and K. Burrage, "Binomial Leap Methods for Simulating Stochastic Chemical Kinetics," *Journal of Chemical Physics*, Vol. 121, No. 21, 2004.

[51] M. Tomita, "Whole-Cell Simulation: A Grand Challenge of the 21st Century," *Trends in Biotechnology*, Vol. 19, No. 6, 2001.

[52] E. Voit, "Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists," Cambridge Univ. Press, 2000.

[53] L. Weinberger, J. Burnett, J. Toettcher, A. Arkin, and D. Schaffer,"Stochastic Gene Expression in a Lentiviral Positive-Feedback Loop: HIV-1 Tat Fluctuations Drive Phenotypic Diversity," *Cell*, Vol. 122, 2005.