Decision Procedure Based Discovery of Rare Behaviors in Stochastic Differential Equation Models of Biological Systems

Arup K. Ghosh, Faraz Hussain, Sumit K. Jha
EECS Department
University of Central Florida
Orlando, FL, USA
Email: akghosh,fhussain,jha@eecs.ucf.edu

Christopher J. Langmead
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA, USA
Email: cjl@cs.cmu.edu

Susmit Jha
EECS Department
UC Berkeley
Berkeley CA 94709
Email: jha@eecs.berkeley.edu

***Abstract*—Stochastic Differential Equation (SDE) models are often used to model the dynamics of complex biological systems. The stochastic nature of these models means that some behaviors are more likely than others. It is often the case that a model's primary purpose is to study rare but interesting or important behaviors, such as the formation of a tumor, or the failure of a cyber-physical system. Unfortunately, due to the limited availability of analytic methods for SDEs, stochastic simulations are the most common means for estimating (or bounding) the probability of rare behaviors. Naturally, the cost of stochastic simulations increases with the rarity of the behavior under consideration. To address this problem, we introduce a new algorithm, RESERCHE, that is specifically designed to quantify the likelihood of rare but interesting behaviors in SDE models. Our approach relies on the use of temporal logics for specifying rare behaviors of possible interest, and on the ability of bit-vector decision procedures to reason exhaustively about fixed precision arithmetic. We also compute the probability of an observed behavior under the assumption of Gaussian noise.**

I. Introduction

Many complex biological systems are often described using stochastic differential equation (SDE) models. SDEs are well suited to the investigation of biological phenomena (e.g., bi-stability in genetic circuits) and cyber-physical systems that are sensitive to stochastic effects. Unfortunately, SDE models can be very difficult to analyze, because of their stochastic nature. In particular, SDE models generally do not admit analytic solutions (except for very restricted forms). For this reason, stochastic simulation is generally used to reason about the model. Here, independent and identically distributed (i.i.d.) trajectories are sampled from the model. Each sample can then be evaluated with respect to some *user-specified* behavior (e.g., whether a tumor forms or an autonomously driven vehicle crashes). The statistics of the resulting Bernoulli process can then be used to estimate the probability with which the behavior holds under the model and to estimate conditional probabilities (e.g., the probability that the tumor is caused by mutations to a particular cell type).

IID-sampling is efficient if the behavior under consideration is common, but many important behaviors are rare. Sampling is an inefficient means for studying rare behaviors because the vast majority of the sampled trajectories will not exhibit the desired behavior. For example, consider a stochastic differential equation model of a population of cells. If the model is realistic, then phenomena such as tumor formation will be rare, as it is in nature. Consequently, the cost of generating multiple tumor-forming trajectories for subsequent analysis may become prohibitive. One natural approach to dealing with the cost of studying rare behaviors is to employ biased (i.e., non-i.i.d.) sampling procedures (e.g., [1]). Here, the underlying probability distribution over trajectories is manipulated in some fashion to expose rare behaviors. We note, however, that biased sampling schemes are usually insensitive to the details of the particular behavior under consideration and may not achieve the desired goal of studying rare behaviors. Indeed, a poorly designed biased sampling scheme can actually increase the costs of studying rare behaviors. A related challenge associated with

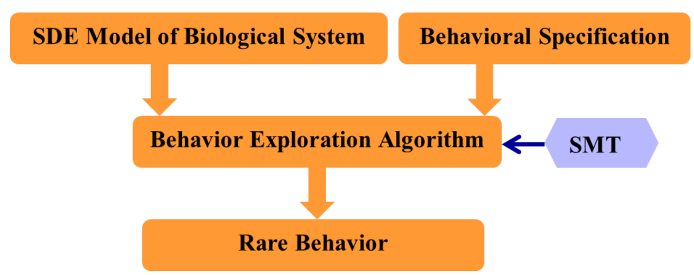

Figure 1. Exploring rare behaviors of SDE model of biological systems.

SDEs (and stochastic models in general) is the need to "debug" models during development. A model is flawed if it cannot exhibit known behaviors. Clearly, sampling cannot certify the non-existence of a rare behavior. On the other hand, decision procedures including Satisfiability Modulo Theories [2] are capable of producing proofs of infeasibility, even for classes of continuous dynamical systems [3]. The algorithm presented in this paper, called RESERCHE, employs decision procedures to address the problem of studying rare behaviors and debugging.

RESERCHE can analyze stochastic differential equation models of complex systems for *a priori* known behavior. The known behavior can be specified using formal spec-

978-1-4673-1321-6/12/$31.00

ifications including probabilistic flavors of temporal logic. Our algorithm then uses Satisfiability Modulo Theory based decision procedures to explore all possible behaviors of the stochastic biological models without explicitly enumerating them. If no behavior with a probability density higher than a given threshold is found, the algorithm reports that the model is incapable of showing the stated behavior with the given probability density. Otherwise, RESERCHE produces the most likely behavior of the model conforming to the given specification.

II. Related Work

Our method draws on concepts and techniques for formal verification, including model checking [4]. The vast majority of these methods are intended for finite-state models. Differential equation models, naturally, are defined on continuous domains. While there are a handful of formal methods suitable for analyzing ODEs (e.g., [5]) and stochastic Continuous Time Markov Chain models [6]–[8], none are suitable for SDE models, with the exception of our previous work [1]. Our previous work on analyzing SDEs [1] uses a combination of non-i.i.d. sampling, Bayesian statistical hypothesis testing, and Girsanov's theorem for change of measures to bound the probability of rare events. As the sampling procedure is not guided by the behavior one is trying to discover, it is not surprising that our previous approach has limitations. In particular, the approach can never prove that a model is incapable of demonstrating a behavior with certainty.

The key difference between our earlier work and this paper is that we employ decision procedures to guide the sampling. Decision Procedures based on Satisfiability Modulo Theory have been previously used for the formal verification of the correctness of software and hardware models, but not for complex stochastic models. Traditional research into SMT solving has focused on linear constraints with simple arithmetic and rich boolean structure. However, the analysis of continuous and probabilistic dynamical systems gives rise to nonlinear constraints. Recent research into nonlinear decision procedures [2] makes our approach feasible.

III. Background

Our proposed algorithm RESERCHE builds on a number of inter-related areas including stochastic differential equations (Sec. III-A), property specifications (Sec. III-B), and decision procedures (Sec. III-C). We now very briefly survey each of these topics.

A. Stochastic Models

The behavior of many complex systems can not be fully captured by deterministic models, such as ODEs. For these systems, non-deterministic models, such as stochastic differential equations can often provide valuable insights. Stochastic models can be broadly partitioned into discrete and continuous state categories. Continuous state space models include stochastic differential equations (SDE) and jump diffusion processes [9]. In such models both the passage of time and the values of state variables are continuous. The algorithms in this paper focusses on stochastic differential equation models.

A stochastic differential equation (SDE) [9] is a differential equation in which some of the terms involve Brownian Motions. A typical SDE is of the following form:

$$dX = b(t, X_t)\ dt\ +\ v(t, X_t)\ dW_t$$

where X is a system variable, b is a Riemann integrable function, v is an Itō integrable function, and W is Brownian Motion. Consider the time between 0 and t as divided into m discrete steps $0, t_1, t_2 \ldots t_m = t$. Further, the solution to a stochastic differential equation [9] is the limit of the following discrete difference equation, as m goes to infinity:

$$X_{t_{k+1}} - X_{t_k} = b(t_k, X_{t_k})(t_{k+1} - t_k) + v(t_k, X_{t_k})(W_{t_{k+1}} - W_{t_k})$$

It has been shown that under technical conditions on the functions b and v, the solution to the stochastic differential equation is well defined, i.e. the limit of the above difference equations exists and is unique. Just like traditional calculus, the rules of Itō integration allow us to compute closed form solutions to *some* stochastic differential equations. As the solutions need not be deterministic, the final closed form solution may include a Brownian Motion or another process, and is itself a stochastic process. We note that our approach could be applied with greater ease on stochastic differential equations that admit a closed form solution. However, in this paper, we will not restrict ourself to only those SDEs for which a closed form solution can be computed.

B. Adapted Finitely Monitorable Specifications

In this subsection, we present the formal definition of the notion of *high-level behavioral specifications* [1] that we later use to describe interesting rare behaviors expected to be observed in the SDE biological model. Our interest lies in behavioral specifications whose truth value can be decided by observing only a finite prefix of a simulation of the SDE model. The logical formulae that capture such properties are known as *adapted finitely monitorable* (AFM) specifications.

A special subclass of AFM specifications on a SDE model $\mathcal{M}$ can be expressed as formulas in *Bounded Linear Temporal Logic* (BLTL). For a biological SDE model $\mathcal{M}$, we can assume that the set of state variables $\mathbf{V}$ is a finite set of variables. A Boolean predicate over $\mathbf{V}$ is a constraint of the form $x \leq v$ or $x \geq v$, where $x \in \mathbf{V}$, and $v \in \mathbb{R}$. A BLTL property is expressed on a finite set of Boolean predicates over $\mathbf{V}$ using Boolean and Temporal operators. The syntax of *Bounded Linear Temporal Logic* (BLTL) [10], [11] is defined using the following formal grammar :
$\phi ::= x \leq v \,|\, x \geq v \,|\, (\phi_1 \vee \phi_2) \,|\, (\phi_1 \wedge \phi_2) \,|\, \neg\phi_1 \,|\, (\phi_1 \mathbf{U^t} \boldsymbol{\phi_2})$,

Here $x \in \mathbf{V}$, $v \in \mathbb{R}$, and t is time. Additionally, we can have other temporal operators such as $\mathbf{G^t}\psi = \neg\mathbf{F^t}\neg\psi$, or $\mathbf{F^t}\psi = \mathbf{True}\,\mathbf{U^t}\,\psi$ in terms of the bounded until $\mathbf{U^t}$. From the definition of the $\mathbf{U}$ operator, we can understand the meaning of these formulas. Explicitly, the formula $\mathbf{G^t}\psi$ implies that ψ holds at all moments for the next t time units into the future and the formula $\mathbf{F^t}\psi$ implies that ψ holds sometime within t time units.

C. Bit-Vector Decision Procedure

Decision procedures are algorithms that decide whether a given logical formula, like those discussed in the previous section, is true or false. Unlike incomplete theorem provers for higher order logic formula, algorithms for deciding the existential fragment of bit-vector first-order logic are quite efficient and practical [2], [12]. Traditionally, two different approaches are used to solve bit-vector arithmetic constraints. In one approach, we translate bit-vectors into Boolean propositions by a process called "bit blasting". This approach is similar to the process by which VLSI circuits for arithmetic are built. In another approach, the decision procedure uses a mathematical programming engine to reason about conjunctions of constraints in the bit-vector Satisfiability Modulo Theory (SMT) formula. Here, the decision procedure may make multiple calls to a mathematical programming engine which, when combined with efficient bookkeeping via the DPLL algorithm, can be used to decide the truth value of the formula. SMT solvers can determine the satisfiability of formulas in expressive logics such as first-order logic and reason about background theories such as nonlinear arithmetic that fix the interpretations of certain predicate and function symbols. Our algorithm uses an SMT-based approach to reason about stochastic differential equations.

Several existing tools including Z3, CVC3, Yices [13], Boolector and Beaver [2] can be used to reason about bit-vector SMT formula. SMT solvers like Beaver have been used to study nonlinear hybrid systems and complex software systems with nonlinear constraints [3], [14], [15].

Bit-vector decision procedures like Beaver can also analyze the feasibility of nonlinear constraints with fixed precision arithmetic connected by arbitrary Boolean connectives like AND, OR and NOT. The ability to analyze constraints connected by Boolean connectives other than conjunction is a key feature of nonlinear bit-vector decision procedures.

In the context of our algorithm, the bit-vector decision procedure is invoked to search for suitable values of the Brownian Motion at various points of time in a simulation such that (i) the SDE models show the behavior in the given specification, and (ii) the overall probability density of observing the behavior is maximized. Our algorithm may invoke the decision procedure several times to ensure that these high level constraints are met.

IV. Algorithms and Results

In this section, we will present our algorithm RESERCHE and key theoretical results that argue the correctness of our algorithm. The high-level idea behind the RESERCHE algorithm is that it transforms both the SDE and the behavior specification into bit-vector SMT formulas, and then uses a decision procedure to decide whether all the formulas can be satisfied (up to a fixed precision). If the formulas cannot be satisfied, then the model does not exhibit the behavior (again, up to fixed precision). If the formulas can be satisfied, the algorithm returns a witness trajectory of maximal probability.

A. Algorithm

We present RESERCHE in Algorithm 1. Our algorithm takes six different inputs: (i) Stochastic differential equation; (ii) Initial state of the variables in the SDE; (iii) Finitely monitorable behavioral specification; (iv) Maximum time for model simulation; (v) Number of discrete time steps in the numerical simulation; and (vi) Lowest Probability Density Behavior we want to investigate.

The algorithm uses these inputs to produce a simulation of the stochastic differential equation that satisfies the given behavioral specification. First, the algorithm uses the maximum time for model simulation t_m and the number of discrete time steps m to compute the temporal discretization Δ at which the SDE solution should be sampled. Then, it computes the threshold T_{max} on the sum of the squares of the Brownian increments using a result that we show in Lemma 4. RESERCHE uses the results sketched in the results section to transform the stochastic differential equation and the specification into Satisfiability Modulo Theory bit-vector constraints. Finally, the SMT solver is used to obtain a possible solution $\overline{X}$ to the constraints put by the SDE, the initial condition, the bound on the probability, and the rare behavior we want to observe. If such a behavior does not exist, the algorithm stops and reports the absence of such a behavior. If the algorithm finds a rare behavior, it continues searching for behaviors with higher probability by reducing the threshold that bounds the sum of the squares of the Brownian increments.

1) SDE to Bit-Vector SMT Formula: A typical SDE is of the following form:

$$dX = b(t, X_t)\ dt\ +\ v(t, X_t)\ dW_t$$

where X is a system variable, b is a Riemann integrable function, v is an Itō integrable function, and W is Brownian Motion. Consider the time between 0 and t as divided into m discrete steps $0, t_1, t_2 \ldots t_m = t$. Recall the solution to a stochastic differential equation (See Sec. III-A), which can be re-written as:

$$X_{t_{k+1}}\ =\ X_{t_k}\ +\ b(t_k, X_{t_k})\ \Delta\ +\ v(t_k, X_{t_k})\ \Delta W_{t_{k+1} \leftrightarrow t_k}$$

Algorithm 1 Generate High Probability Behavior Satisfying Specification ϕ

Require: $dX(t) = b(t, X_t)dt + v(t, X_t)dW_t$: Stochastic Differential Equation
 $X(0)$: Initial Value of the variable X
 ϕ: Behavioral Specification in a Finitely Monitorable Logic like BLTL
 t_m: Maximum Time of Simulation
 m: Number of Discrete Simulation Steps
 $P_{min} \geq 0$: Lowest probability density behavior acceptable as witness to ϕ

Ensure: Produce a behavior τ satisfying ϕ with highest probability density $P(\tau)$
 $\Delta = \frac{t_m}{m}$
 Threshold $T_{max} = 2\Delta \log(\frac{1}{P_{min}(2\pi\Delta)^{m/2}})$
 $\mathcal{B}(\phi) \leftarrow$ Specification2BooleanAssertion(ϕ) //Transform Specification ϕ to Boolean Assertions
 $\mathcal{B}(SDE) \leftarrow \bigwedge_{k=0}^{m-1} \Big(X_{t_{k+1}} = X_{t_k} + b\,(t_k, X_{t_k})\ \Delta + v\,(t_k, X_{t_k})\ \Delta W_{t_{k+1} \leftrightarrow t_k} \Big)$
 //Transform SDE to Boolean Assertions
 $T = T_{max}$ //Initialize threshold
 $\mathcal{B}(Init) \leftarrow (X_0 == X(0))$ //Assert Initial Value
 FeasibleWitness $= 0$
 while $T \geq 0$ **do**
  $\mathcal{B}(Threshold) \leftarrow \sum_{k=0}^{m-1} \left(\Delta W_{t_{k+1}-t_k}\right)^2 \leq T$ //Assert Probability Density of Rare Behavior
  if $\mathcal{B}(Init) \wedge \mathcal{B}(SDE) \wedge \mathcal{B}(\phi) \wedge \mathcal{B}(Threshold)$ **then**
   //A feasible solution has been obtained
   $[\overline{X}, \overline{\Delta W_{t_{k+1}-t_k}}] \leftarrow$ Feasible Model for $\mathcal{B}(Init) \wedge \mathcal{B}(SDE) \wedge \mathcal{B}(\phi) \wedge \mathcal{B}(Threshold)$
   FeasibleWitness $= 1$
   $P = (2\pi)^{-m/2}\Delta^{-m/2}e^{-\frac{1}{2\Delta}(\sum_{i=1}^{m} T^2)}$ //Compute Probability Density from Threshold
  end if
  $T = \lfloor T/2 \rfloor$
 end while
 if FeasibleWitness == 0 **then**
  Print "No feasible behavior found with probability density P_{min} or more"
 else
  Print "Feasible behavior satisfying ϕ found"
  Print model behavior $[\overline{X}, \overline{\Delta W_{t_{k+1}-t_k}}]$
  Print probability density P
 end if

In our approach, we represent each of the variables X_{t_k} and $\Delta W_{t_{k+1} \leftrightarrow t_k}$ using fixed precision bit vectors and use bit-vector arithmetic to reason about mathematical operations. The solution to the Stochastic Differential Equation can be represented by the conjunction of the above constraints $\bigwedge_{k=0}^{m-1} \Big(X_{t_{k+1}} = X_{t_k} + b\,(t_k, X_{t_k})\ \Delta + v\,(t_k, X_{t_k})\ \Delta W_{t_{k+1} \leftrightarrow t_k} \Big)$. In order to ensure that the fixed precision does not lead to stability issues, we consider bit-vectors with varying orders of bit-width. Also, we add constraints to include the possible set of initial values of the variable X.

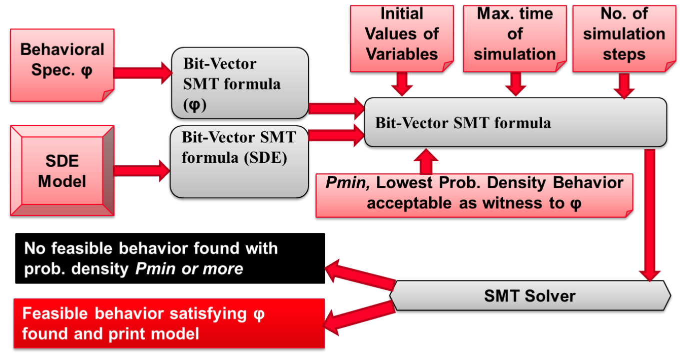

Figure 2. RESERCHE: A decision procedures based algorithm to find rare behaviors in stochastic models

Our use of fixed precision bit-vector arithmetic to reason about SDEs is similar to the study of dynamical hybrid systems with nonlinear arithmetic constraints [3]. However, their use of the decision procedure is limited to the exhaustive analysis of non-deterministic choices made by a possibly adversarial agent while we study the probabilistic outcome of a stochastic process. By adding further constraints to the SMT formula representing the solutions to the SDE, we restrict the solution space of the SDE to those behaviors that actually satisfy the given specification ϕ. Also, by carefully adding a well thought constraint on the magnitudes of the bit-vector variables representing the Brownian Motion, we also enforce the fact that these behaviors have a high probability density.

2) Specification to Bit-vector SMT formula: In order to constrain the behaviors of the stochastic differential equation, we translate the finitely monitorable specification to a SMT formula over bit-vectors. In this subsection, we demonstrate the translation of Bounded Linear Temporal Logic (BLTL) into bit-vector formulae. The translation function $\mathcal{F}$ takes a BLTL formula and a time step as input. Our translation is given by the following recursive rewrite rules:

- $\mathcal{F}(X \geq v, k) = X_k \geq v$;
- $\mathcal{F}(X \leq v, k) = X_k \leq v$;
- $\mathcal{F}(\phi_1 \wedge \phi_2, k) = \mathcal{F}(\phi_1, k) \wedge \mathcal{F}(\phi_2, k)$;
- $\mathcal{F}(\neg\phi, k) = \neg\, \mathcal{F}(\phi, k)$;
- $\mathcal{F}(\phi_1 \mathbf{U}^\tau \phi_2, k) = \bigvee_{i=k}^{m} \Big(0 \leq \sum_{j=k}^{i} \Delta \leq \tau \bigwedge \mathcal{F}(\phi_2, i) \bigwedge \big(\bigwedge_{j=k}^{i} \mathcal{F}(\phi_1, j)\big)\Big)$.

We note that the right hand side of every rewrite rule is either a bit-vector SMT formula or a Boolean combination of a bit-vector SMT formula and the translation function $\mathcal{F}$ over simpler formula. Because our specification formulae have finite sizes and we are only interested in properties over bounded time, our translation is bound to terminate and produce a finite bit-vector SMT formula. One can prove this easily using induction on the length of the formula and the time step parameter in the definition of the translation function.

B. Results

In this section, we will argue the correctness of our algorithm. When the algorithm produces a behavior of the SDE satisfying the given specification ϕ, we will derive the probability density of the behavior obtained. Our results will rely on the independence of increments of Brownian Motion and their Gaussian distribution.

1) Probability Density of Solution to SDE: Given a finely discretized solution $X_{t_0}, X_{t_1} \ldots, X_{t_m}$ to the SDE initial value problem, a natural question to ask is the probability density of observing this solution. Suppose we are given the initial value X_{t_0}, and we want to compute the probability density of observing the value X_{t_1} after t_1 time. Suppose the corresponding values of Brownian Motion at t_0 and t_1 are respectively $\widehat{W_{t_0}}$ and $\widehat{W_{t_1}}$.

$$P(X_{t_1}|X_{t_0}) = P(W_{t_1} - W_{t_0} = \widehat{W_{t_1}} - \widehat{W_{t_0}} \mid X_{t_0}) \qquad (1)$$

$$= (1/\sqrt{2\pi(t_1 - t_0)})\ e^{-\left(\frac{(|\widehat{W_{t_1}} - \widehat{W_{t_0}}|)^2}{2(t_1 - t_0)}\right)} \qquad (2)$$

Equation 1 holds because the only stochastic component of X_{t_1} is completely determined given the increment in Brownian Motion and the initial value X_{t_0} for small values of $t_1 - t_0$. Since X_t is adapted to the stochastic process W_t, and increments in Brownian motion are normally distributed, we know that $\widehat{W_{t_1}} - \widehat{W_{t_0}} \sim \mathcal{N}(0, t_1 - t_0)$. Hence, Eqn 2 gives the desired probability distribution.

Lemma 1: The probability density of a discretized solution to a stochastic differential equation is inversely proportional to the exponential of the sum of squares of the increments of the Brownian motion.

Proof: We compute the probability of observing the sequence of the observed discretized solution given the initial value:

$$P(X_{t_0}, X_{t_1}, \ldots X_{t_m}|X_{t_0})$$
$$= (2\pi)^{-m/2}\Delta^{-m/2} e^{-\frac{1}{2\Delta}\left(\sum_{i=1}^{m}\left(\widehat{W_{t_i}} - \widehat{W_{t_{i-1}}}\right)^2\right)} \qquad (3)$$

■

Equation 3 gives the probability density of observing a given behavior in a stochastic differential equation. It satisfies our intuition that large values of Brownian motion increments should correspond to smaller probability densities while small values of Brownian motion increments correspond to large probability densities. However, Equation 3 shows that the probability density only depends on the sum of squares of the increments of Brownian motion, and not on the individual increments themselves.

Lemma 2: $\sum_{i=1}^{m}\left(\frac{\widehat{W_{t_i}} - \widehat{W_{t_{i-1}}}}{\sqrt{\Delta}}\right)^2$ has a χ-distribution with m degrees of freedom.

Lemma 3: Given the sum of the squares of the magnitudes of the Brownian Motion increments, the probability density of a path is independent of the Brownian motions themselves.

Lemma 4: Given a lower bound on the probability density P of a discretized SDE behavior with m samples every Δ time apart, the sum of squares of increments of Brownian motion $T \equiv \sum_{i=1}^{m}\left(\widehat{W_{t_i}} - \widehat{W_{t_{i-1}}}\right)^2$ should be no more than $2\Delta \log\left(\frac{1}{P(2\pi\Delta)^{m/2}}\right)$.

A key concern in discretizing a continuous stochastic differential equation is the error introduced by sampling a continuous system, and replacing a SDE with a discretized difference equation. The existence and uniqueness of SDE ensures that the solution of a sufficiently discretized SDE approaches the solution of the continuous SDE. The distance between the solution obtained from a finitely discretized SDE and the soution of a continuous SDE, however, remains an issue of interest.

V. Experimental Results

In this section, we explore a stochastic differential equation model representing the dynamics of eukaryotic cell division. In [16], the authors propose a minimal two variable mathematical model that couples the dynamics of cell growth and the cell cycle division:

$$dX(t) = (s_x - (d_x + c_{xy}Y)\ X \quad + \quad a_x X^2)\ dt \quad + \quad \sigma_x\ dW_t$$
$$dY(t) = (-d_y Y + s_y X^2)dt + \sigma_y dW_t$$

The first variable X is a set of components (e.g. the activated cyclin-Cdk) involved in initiating cascade of reactions which trigger the most CC events. The second variable Y is set of components (e.g. the anaphase promoting complex with Cdc20 and Cdh1) that control the activity of X-components and responsible for cell division (cytokinesis).

Threshold (T)	Time (seconds)	Memory (MB)	Is behavior feasible?
1000	11.489	68	Yes
100	28.330	67	Yes
2	98.729	103	No
1	40.289	68	No
0	6.664	54	No

Table I
PERFORMANCE OF BEAVER ON PROBLEMS WITH DIFFERENT THRESHOLDS.

In our analysis, we used the following values for the constants: $s_x = 1$, $c_{xy} = 1$, $a_x = 1$, $s_x = 1$, $s_y = 1$. We used the bitvector decision procedure Beaver with 32 bits of representation, and 6 time steps. Using the constraint that the sum of Brownian Motions is zero, we obtained a solution that corresponds to the ordinary differential equation component of the model. Under the approximate bit-vector arithmetic and the discretization of time, the maximum value of Y obtained is $Y_{max} = 2$. We then asked if it was possible to generate a maximum value of 2^{14} with varying degrees of threshold of the sum of squares of the Brownian Motion increments. The results of our experiments are shown in Table I. Large bounds on the sums of increments of the Brownian motion permit the system to demonstrate this rare behavior, while small bounds on the sum of increments of the Brownian motion produce an unsatisfiable formula.

VI. Conclusion

We have introduced the RESERCHE algorithm for efficiently investigating rare behaviors in SDE models. Informally, our method avoids the computational costs associated with sampling by, in effect, searching for trajectories from the model that satisfy a given behavioral specification. That is, our method *only* generates trajectories that exhibit the behavior (if such trajectories exist), and then estimates the probability of those trajectories. The actual search is performed by converting the SDE and the behavioral specification into bit-vector SMT formulas, and then calling an appropriate decision procedure to find witness trajectories. Thus, our method takes advantage of the efficiency and power of modern SMT-solvers. Consequently, as newer, more powerful decision procedures are created, our method will inherit the benefits of those methods.

Several interesting directions for future research remain open. We are studying the use of Satisfiability Modulo Theory solving techniques to analyze rare behaviors of closed form solutions to stochastic differential equations. Many practical applications require the study of a system where one component is a stochastic differential equation and the other component may be an ODE or even a finite state controller. Our proposed rare behavior sampling approach should be extended to such systems in order to study biologically important cyber-physical systems like artificial pancreas. The development of specialized SMT solving techniques that can solve the decision problems arising from such analysis is also an exciting area.

References

[1] S. Jha and C. Langmead, “Exploring Behaviors of SDE Models of Biological Systems using Change of Measures,” in *Proc. of the 1st IEEE International Conference on Computational Advances in Bio and medical Sciences (ICCABS)*, 2011, pp. 111–117.

[2] S. Jha, R. Limaye, and S. A. Seshia, “Beaver: Engineering an efficient smt solver for bit-vector arithmetic,” in *21st International Conference on Computer-Aided Verification*, 2009, pp. 668–674.

[3] S. Jha, B. A. Brady, and S. A. Seshia, “Symbolic Reachability Analysis of Lazy Linear Hybrid Automata,” in *5th International Conference on Formal Modelling and Analysis of Timed Systems*, 2007, pp. 241–256.

[4] C. Baier and J.-P. Katoen, *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008.

[5] C. Piazza, M. Antoniotti, V. Mysore, A. Policriti, F. Winkler, and B. Mishra, “Algorithmic algebraic model checking i: Challenges from systems biology,” in *17th International Conference on Computer Aided Verification*, 2005, pp. 5–19.

[6] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn, “Probabilistic model checking of complex biological pathways,” in *Proc. Computational Methods in Systems Biology (CMSB’06)*, ser. Lecture Notes in Bioinformatics, C. Priami, Ed., vol. 4210. Springer Verlag, 2006, pp. 32–47.

[7] S. Jha, E. Clarke, C. Langmead, A. Platzer, and P. Zuliani, “A bayesian approach to model checking biological systems,” *Proc. 7th Conf. on Computational Methods in Systems Biology (CMSB)*, pp. 218–234, 2009.

[8] E. Clarke, J. Faeder, L. Harris, C. Langmead, A. Legay, and S. Jha, “Statistical model checking in biolab: Application to the automated analysis of the t-cell receptor signalling pathway,” *Proc. 6th Conf. on Computational Methods in Systems Biology (CMSB)*, pp. 231–250, 2008.

[9] I. Karatzas and S. Shreve, *Brownian Motion and Stochastic Calculus (Graduate Texts in Mathematics)*, 2nd ed. Springer, 1991.

[10] A. Pnueli, “The temporal logic of programs,” in *Proceedings of the 18th IEEE Symposium on Foundation of Computer Science*, 1977, pp. 46–57.

[11] Q.-G. Xu and H.-K. Miao, “Timed automata for metric interval temporal logic formulae in prototype verification system,” *Journal of Shanghai University (English Edition)*, vol. 12, no. 4, pp. 339–346, 2008.

[12] A. K. Ghosh, “Speeding up sat solver by exploring cnf symmetries: Revisited,” *Congressus Numerantium*, vol. 206, pp. 73–84, 2010.

[13] B. Dutertre and L. de Moura, “The Yices SMT solver,” Tool paper at http://yices.csl.sri.com/tool-paper.pdf, SRI International, Aug. 2006.

[14] S. Jha, S. Gulwani, S. A. Seshia, and A. Tiwari, “Oracle-guided component-based program synthesis,” in *Proceedings of the 32nd International Conference on Software Engineering (ICSE)*, 2010, pp. 215–224.

[15] S. Jha, S. Gulwani, S. Seshia, and A. Tiwari, “Synthesizing switching logic for safety and dwell-time requirements,” in *Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*, 2010, pp. 22–31.

[16] B. Pfeuty and K. Kaneko, “Minimal requirements for robust cell size control in eukaryotic cells,” *Physical Biology*, 2007.