Probabilistic hybrid systems verification via SMT and Monte Carlo techniques

Fedor Shmarov and Paolo Zuliani

School of Computing Science, Newcastle University, Newcastle upon Tyne, UK
{f.shmarov, paolo.zuliani}@ncl.ac.uk

Abstract. We develop numerically rigorous Monte Carlo approaches for computing probabilistic reachability in hybrid systems subject to random and nondeterministic parameters. Instead of standard simulation we use δ-complete SMT procedures, which enable formal reasoning for nonlinear systems up to a user-definable numeric precision. Monte Carlo approaches for probability estimation assume that sampling is possible for the real system at hand. However, when using δ-complete simulation one instead samples from an overapproximation of the real random variable. In this paper, we introduce a Monte Carlo-SMT approach for computing probabilistic reachability confidence intervals that are both statistically and numerically rigorous. We apply our technique to hybrid systems involving nonlinear differential equations.

1 Introduction

In this paper we combine statistical (Monte Carlo) techniques and numerically sound decision procedures to reason about hybrid systems with random and nondeterministic parameters. In particular, we devise confidence-interval techniques for *bounded probabilistic reachability*, *i.e.*, we aim at computing statistically valid enclosures for the probability that a hybrid system reaches a given set of states within a given time bound and number of discrete transitions. When nondeterministic parameters are present, a hybrid system will in general feature a range of reachability probabilities, depending on the value of the nondeterministic parameters. Reachability is an important class of behavioural properties, as many verification problems (*e.g.*, proving system safety) can be reduced to reachability questions. A statistical approach to probabilistic reachability is important because statistical techniques trade correctness guarantees with efficiency, and so can scale much better with system size than other rigorous approaches. For example, statistical model checking [15] can be faster than probabilistic model checking, which is based on exhaustive state space search [14]. Also, statistical model checking can handle models for which no efficient verification tools exist, such as cyber-physical systems [2].

Monte Carlo techniques for probability estimation assume that one can sample the random variable representing the true system behaviour. However, while this is possible for certain finite-state systems, for nonlinear systems (*e.g.*, ordinary differential equations (ODEs) with trigonometric functions) it is not. In

fact, sampling the random variable representing the true system behaviour can be as hard as reachability, which is undecidable even for very simple systems (*e.g.*, linear hybrid automata [1]). Thus, one has to deal with numerical imprecisions that could lead to missing important events in the true system evolution. For example, zero-crossings can be indistinguishable from "safe" trajectories [8].

A novel aspect of our work is that we explicitly take into account undecidability and numerical precision by employing δ-complete decision procedures [4], which enable formal reasoning up to a user-defined numerical precision over bounded domains. In this way it is possible to handle in a sound and safe manner complex dynamical systems, such as nonlinear ODEs [6]. Given any $\delta > 0$ and an arbitrary first-order formula ϕ over the reals, a δ-complete decision procedure returns **unsat** if ϕ is false and δ-**sat** if ϕ^δ (a weaker version of formula ϕ) is true. Note that the latter result does not imply satisfiability of the initial formula. Also, the value of δ affects the precision of the result, and large values of δ can cause *false* alarms (*i.e.*, δ-**sat** is returned for a formula which is in fact false). Statistical techniques must therefore take into account that samples are only approximation of the real random variable corresponding to the system evolution. In particular, we introduce an approach for computing statistically *and* numerically rigorous confidence intervals for probabilistic reachability. We exemplify our techniques to hybrid systems with random and/or nondeterministic parameters. For systems with both random and nondeterministic parameters we estimate the (nondeterministic) parameter values that result in the minimal and maximal reachability probabilities. Our algorithms can in principle be applied to other stochastic models (*e.g.*, continuous-time Markov chains) should the corresponding δ-complete decision procedure be available.

Related Work. We focus on works that combine statistical techniques with SMT procedures, which are the main subject areas of the paper. The tool SReach [13] combines statistical estimation with δ-complete simulation procedures. However, SReach only considers overapproximations of the reachability probability, and thus can offer one-sided confidence intervals only. We instead compute confidence intervals that are *guaranteed* to contain both the under- and overapproximation of the reachability probability. Also, SReach does not handle nondeterministic parameters, while we do. In [3] the authors present a statistical model checking approach combined with SMT decision procedures, but it is restricted to fixed-sample size techniques, while we develop a more efficient sequential Bayesian approach and consider δ-complete decision procedures.

2 Bounded Reachability in Hybrid Systems

Hybrid systems provide a framework for modelling real-world systems that combine continuous and discrete dynamics [1]. We consider parametric hybrid systems as a variant of hybrid systems featuring continuous and discrete parameters whose values are set in the initial state and do not change during the system's evolution. Such parameters can be random when there is a probability measure associated with them, and nondeterministic otherwise. We now formally define the systems we consider in this paper.

Definition 1. (PHS) *A Parametric Hybrid System is a tuple*

$$H =< Q, \Upsilon, X, P, Y, R, \text{jump}, \text{goal} >$$

where

- $Q = \{q_0, \cdots, q_m\}$ *a set of modes (discrete components of the system),*
- $\Upsilon = \{(q, q') : q, q' \in Q\}$ *a set of transitions between modes,*
- $X = [u_1, v_1] \times \cdots \times [u_n, v_n] \subset \mathbb{R}^n$ *a domain of continuous variables,*
- $P = [a_1, b_1] \times \cdots \times [a_k, b_k] \subset \mathbb{R}^k$ *the parameter space of the system,*
- $Y = \{\mathbf{y}_q(\mathbf{p}, t) : q \in Q, \mathbf{p} \in X \times P, t \in [0, T]\}$ *the continuous system dynamics where* $\mathbf{y}_q : X \times P \times [0, T] \to X$*,*
- $R = \{\mathbf{g}_{(q,q')}(\mathbf{p}, t) : (q, q') \in \Upsilon, \mathbf{p} \in X \times P, t \in [0, T]\}$ *'reset' functions* $\mathbf{g}_{(q,q')} : X \times P \times [0, T] \to X \times P$ *defining the continuous state at time* $t = 0$ *in mode* q' *after taking the transition from mode* q*.*

and predicates (or relations)

- $\text{jump}_{(q,q')}(\mathbf{x})$ *defines a discrete transition* $(q, q') \in \Upsilon$ *which may (but does not have to) occur upon reaching the jump condition in state* $(\mathbf{x}, q) \in X \times P \times Q$*,*
- $\text{goal}_q(\mathbf{x})$ *defines the goal state* $\mathbf{x}$ *in mode* q*.*

The continuous system dynamics Y is represented by initial value problems with Lipschitz-continuous ODEs, which by the well-known Picard-Lindelöf theorem have a unique solution for any given initial condition $\mathbf{p} \in X \times P$ and $t_0 \in [0, T]$. We treat system parameters as any other variable, except that their derivatives are zero. Thus, the parameters are part of the initial conditions.

Bounded reachability in PHSs aims to decide whether, for given initial conditions, the system reaches a goal state in a finite number of discrete transitions. Given a PHS and a reachability depth l we can derive the set $\mathbf{Path}(l)$ of all paths π of length $|\pi| = l + 1$ whose first ($\pi(0)$) and last ($\pi(l)$) elements are the initial and the goal mode, respectively. The *bounded reachability property* for a path $\pi \in \mathbf{Path}(l)$ and initial condition $\mathbf{p}$ can be checked by evaluating the formula:

$$\begin{aligned}
&\phi(\pi, \mathbf{p}) := \exists^{[0,T]} t_0, \cdots, \exists^{[0,T]} t_{|\pi|-1} : \big(\mathbf{x}^t_{\pi(0)} = \mathbf{y}_{\pi(0)}(\mathbf{p}, t_0)\big) \wedge \\
&\bigwedge_{i=0}^{|\pi|-2} \Big[\text{jump}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)}) \wedge \big(\mathbf{x}^t_{\pi(i+1)} = \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)}, t_i), t_{i+1})\big)\Big] \\
&\wedge \text{goal}_{\pi(|\pi|-1)}(\mathbf{x}^t_{\pi(|\pi|-1)}) \ .
\end{aligned} \tag{1}$$

where $\exists^{[0,T]} t_i$ is a shorthand for $\exists t_i \in [0, T]$.

Note that the terms $\mathbf{x}^t_{\pi(i+1)} = \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)}, t_i), t_{i+1})$ and $\mathbf{x}^t_{\pi(0)} = \mathbf{y}_{\pi(0)}(\mathbf{p}, t_0)$ are purely syntactic substitutions. Formulas over the reals like (1) are undecidable in general [9], but a relaxed version (δ-weakening [4]) is instead decidable.

Definition 2. (δ-Weakening [4]) *Given a bounded Σ_1 sentence and an arbitrarily small positive δ*

$$\exists^X \mathbf{x} : \bigwedge_{i=1}^{m} \Big(\bigvee_{j=1}^{k_i} (f_{i,j}(\mathbf{x}) = 0) \Big)$$

(where the $f_{i,j}$ are Type-2 real computable functions) its δ-weakening is

$$\exists^X \mathbf{x} : \bigwedge_{i=1}^{m} \Big(\bigvee_{j=1}^{k_i} (|f_{i,j}(\mathbf{x})| \leq \delta) \Big)$$

It is easy to see that the bounded reachability property (1) can be rewritten in the format of Definition 2 (see [4]). A δ-complete decision procedure [4] *correctly* decides whether an arbitrary bounded Σ_1 (existentially quantified) sentence is false (**unsat** answer) or its δ-weakening is true (δ-**sat** answer). Note that with a δ-complete decision procedure **unsat** can always be trusted, while δ-**sat** might in fact be a false alarm due to a coarse overapproximation characterised by δ.

Evaluating (1) by a δ-complete decision procedure returns **unsat** only if for the given parameter value $\mathbf{p}$ the path does not reach a goal state. If δ-**sat** is returned, we may try to sharpen the answer by checking an appropriate formula. For example, an **unsat** answer to formula $\phi^{\forall}(\pi, \mathbf{p})$ below implies reachability:

$$\begin{aligned}
&\phi^{\forall}(\pi, \mathbf{p}) := \forall^{[0,T]} t_0, \cdots, \forall^{[0,T]} t_{|\pi|-1} : \big(\mathbf{x}^t_{\pi(0)} \neq \mathbf{y}_{\pi(0)}(\mathbf{p}, t_0)\big) \vee \\
&\bigvee_{i=0}^{|\pi|-2} \Big[\neg \mathrm{jump}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)}) \vee \big(\mathbf{x}^t_{\pi(i+1)} \neq \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)}, t_i), t_{i+1})\big) \Big] \\
&\vee \neg \mathrm{goal}_{\pi(|\pi|-1)}(\mathbf{x}^t_{\pi(|\pi|-1)})
\end{aligned}$$

In the previous formula the time variables are quantified universally. Current implementations of δ-complete decision procedures [5] can only handle formulas where the universal quantification is introduced over a single time variable. The goal predicate in $\phi^{\forall}(\pi, \mathbf{p})$ depends on $|\pi|$ variables and thus cannot be handled directly. To resolve this issue we instead evaluate a series of formulas ψ_j:

$$\begin{aligned}
&\psi_j(\pi, \mathbf{p}) := \exists^{[0,T]} t_0, \cdots, \forall^{[0,T]} t_j : \big(\mathbf{x}^t_{\pi(0)} = \mathbf{y}_{\pi(0)}(\mathbf{p}, t_0)\big) \wedge \\
&\bigwedge_{i=0}^{j-1} \Big[\mathbf{x}^t_{\pi(i+1)} = \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)}, t_i), t_{i+1}) \Big] \wedge \neg \mathrm{jump}_{(\pi(j),\pi(j+1))}(\mathbf{x}^t_{\pi(j)})
\end{aligned} \tag{2}$$

if $j < |\pi| - 1$ and

$$\begin{aligned}
&\psi_j(\pi, \mathbf{p}) := \exists^{[0,T]} t_0, \cdots, \forall^{[0,T]} t_j : \big(\mathbf{x}^t_{\pi(0)} = \mathbf{y}_{\pi(0)}(\mathbf{p}, t_0)\big) \wedge \\
&\bigwedge_{i=0}^{j-1} \Big[\mathbf{x}^t_{\pi(i+1)} = \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)}, t_i), t_{i+1}) \Big] \wedge \neg \mathrm{goal}_{\pi(j)}(\mathbf{x}^t_{\pi(j)})
\end{aligned} \tag{3}$$

if $j = |\pi| - 1$. The next proposition establishes a stronger formula for reachability.

Proposition 1. *With the definitions in (1), (2) and (3) we have*

$$\bigwedge_{j=0}^{|\pi|-1} \neg\psi_j(\pi,\mathbf{p}) \;\Rightarrow\; \phi(\pi,\mathbf{p})$$

Proof. Consider the case $|\pi| = 1$. It can be seen that $\neg\psi_0(\pi,\mathbf{p}) \Leftrightarrow \phi(\pi,\mathbf{p})$ as

$$\neg\psi_0(\pi,\mathbf{p}) := \exists^{[0,T]} t_0 : \mathrm{goal}_{\pi(0)}(\mathbf{x}^t_{\pi(0)}) \Leftrightarrow \phi(\pi,\mathbf{p})$$

Consider now the case $|\pi| > 1$.

$$\bigwedge_{j=0}^{|\pi|-1} \neg\psi_j(\pi,\mathbf{p}) := \bigwedge_{j=0}^{|\pi|-2} \Bigg[\forall^{[0,T]} t_0,\cdots,\forall^{[0,T]} t_{j-1},\exists^{[0,T]} t_j : \big(\mathbf{x}^t_{\pi(0)} \neq \mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\vee$$
$$\bigvee_{i=0}^{j-1} \Big(\mathbf{x}^t_{\pi(i+1)} \neq \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})\Big) \vee \mathrm{jump}_{(\pi(j),\pi(j+1))}(\mathbf{x}^t_{\pi(i)})\Bigg] \wedge$$
$$\Bigg[\forall^{[0,T]} t_0,\cdots,\forall^{[0,T]} t_{|\pi|-2},\exists^{[0,T]} t_{|\pi|-1} : \big(\mathbf{x}^t_{\pi(0)} \neq \mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\vee$$
$$\bigvee_{i=0}^{|\pi|-2} \Big(\mathbf{x}^t_{\pi(i+1)} \neq \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})\Big) \vee \mathrm{goal}_{\pi(|\pi|-1)}(\mathbf{x}^t_{\pi(|\pi|-1)})\Bigg]$$

We recall that terms $\mathbf{x}^t_{\pi(0)} = \mathbf{y}_{\pi(0)}$ and $\mathbf{x}^t_{\pi(i+1)} = \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})$ are just a syntactic substitution which cannot be falsified as the system dynamics always exist (by the Picard-Lindelöf theorem). Hence, the formula above implies the following:

$$\bigwedge_{j=0}^{|\pi|-2} \Bigg[\forall^{[0,T]} t_0,\cdots,\forall^{[0,T]} t_{j-1},\exists^{[0,T]} t_j : \big(\mathbf{x}^t_{\pi(0)} = \mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\wedge$$
$$\bigwedge_{i=0}^{j-1} \Big(\mathbf{x}^t_{\pi(i+1)} = \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1),t_i)}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})\Big) \wedge \mathrm{jump}_{(\pi(j),\pi(j+1))}(\mathbf{x}^t_{\pi(i)})\Bigg] \wedge$$
$$\Bigg[\forall^{[0,T]} t_0,\cdots,\forall^{[0,T]} t_{|\pi|-2},\exists^{[0,T]} t_{|\pi|-1} : \big(\mathbf{x}^t_{\pi(0)} = \mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\wedge$$
$$\bigwedge_{i=0}^{|\pi|-2} \Big(\mathbf{x}^t_{\pi(i+1)} = \mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1),t_i)}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})\Big) \wedge \mathrm{goal}_{\pi(|\pi|-1)}(\mathbf{x}^t_{\pi(|\pi|-1)})\Bigg]$$

The next step can be equivalently derived by moving universal quantifiers from the second part of the formula (square brackets containing the goal predicate)

outside the entire formula:

$$\forall^{[0,T]}t_0,\cdots,\forall^{[0,T]}t_{|\pi|-2}:\bigwedge_{j=0}^{|\pi|-2}\Bigg[\exists^{[0,T]}t_j:\big(\mathbf{x}^t_{\pi(0)}=\mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\wedge$$

$$\bigwedge_{i=0}^{j-1}\Big(\mathbf{x}^t_{\pi(i+1)}=\mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})\Big)\wedge\mathrm{jump}_{(\pi(j),\pi(j+1))}(\mathbf{x}^t_{\pi(i)})\Bigg]\wedge$$

$$\Bigg[\exists^{[0,T]}t_{|\pi|-1}:\big(\mathbf{x}^t_{\pi(0)}=\mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\wedge$$

$$\bigwedge_{i=0}^{|\pi|-2}\Big(\mathbf{x}^t_{\pi(i+1)}=\mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})\Big)\wedge\mathrm{goal}_{\pi(|\pi|-1)}(\mathbf{x}^t_{\pi(|\pi|-1)})\Bigg]$$

The existential quantifiers $\exists^{[0,T]}t_j$ can be eliminated as variables t_j are already quantified universally. Also $\exists^{[0,T]}t_{|\pi|-1}$ can be moved in front of the formula as its first part (square brackets containing jump predicates) does not depend of $t_{|\pi|-1}$. Hence, the formula above can be written as:

$$\forall^{[0,T]}t_0,\cdots,\forall^{[0,T]}t_{|\pi|-2},\exists^{[0,T]}t_{|\pi|-1}:\bigwedge_{j=0}^{|\pi|-2}\Bigg[\big(\mathbf{x}^t_{\pi(0)}=\mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\wedge$$

$$\bigwedge_{i=0}^{j-1}\Big(\mathbf{x}^t_{\pi(i+1)}=\mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})\Big)\wedge\mathrm{jump}_{(\pi(j),\pi(j+1))}(\mathbf{x}^t_{\pi(i)})\Bigg]\wedge$$

$$\Bigg[\big(\mathbf{x}^t_{\pi(0)}=\mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\wedge$$

$$\bigwedge_{i=0}^{|\pi|-2}\Big(\mathbf{x}^t_{\pi(i+1)}=\mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})\Big)\wedge\mathrm{goal}_{\pi(|\pi|-1)}(\mathbf{x}^t_{\pi(|\pi|-1)})\Bigg]\Leftrightarrow$$

By idempotency of conjunction ($A\wedge A=A$) terms $\mathbf{x}^t_{\pi(0)}=\mathbf{y}_{\pi(0)}$ and $\mathbf{x}^t_{\pi(i+1)}=\mathbf{y}_{\pi(i)}(\mathbf{g}_{(\pi(i),\pi(i+1))}(\mathbf{x}^t_{\pi(i)},t_i),t_{i+1})$ can be merged:

$$\forall^{[0,T]}t_0,\cdots,\forall^{[0,T]}t_{|\pi|-2},\exists^{[0,T]}t_{|\pi|-1}:\big(\mathbf{x}^t_{\pi(0)}=\mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\wedge$$

$$\bigwedge_{j=0}^{|\pi|-2}\Bigg[\Big(\mathbf{x}^t_{\pi(j+1)}=\mathbf{y}_{\pi(j)}(\mathbf{g}_{(\pi(j),\pi(j+1))}(\mathbf{x}^t_{\pi(j)},t_j),t_{j+1})\Big)\wedge\mathrm{jump}_{(\pi(j),\pi(j+1))}(\mathbf{x}^t_{\pi(i)})\Bigg]\wedge$$

$$\mathrm{goal}_{\pi(|\pi|-1)}(\mathbf{x}^t_{\pi(|\pi|-1)})$$

Finally, the following is implied:

$$\exists^{[0,T]}t_0,\cdots,\exists^{[0,T]}t_{|\pi|-2},\exists^{[0,T]}t_{|\pi|-1}:\big(\mathbf{x}^t_{\pi(0)}=\mathbf{y}_{\pi(0)}(\mathbf{p},t_0)\big)\wedge$$

$$\bigwedge_{j=0}^{|\pi|-2}\Bigg[\Big(\mathbf{x}^t_{\pi(j+1)}=\mathbf{y}_{\pi(j)}(\mathbf{g}_{(\pi(j),\pi(j+1))}(\mathbf{x}^t_{\pi(j)},t_j),t_{j+1})\Big)\wedge\mathrm{jump}_{(\pi(j),\pi(j+1))}(\mathbf{x}^t_{\pi(i)})\Bigg]\wedge$$

$$\mathrm{goal}_{\pi(|\pi|-1)}(\mathbf{x}^t_{\pi(|\pi|-1)})\Leftrightarrow\phi(\pi,\mathbf{p})$$

□

Proposition 1 enables us to define an **evaluate** procedure (Algorithm 1) which, given a parametric hybrid system H, reachability depth l, a parameter value $\mathbf{p} \in X \times P$ and a precision δ for the δ-complete decision procedure, returns **sat** if $\exists \pi \in \mathbf{Path}(l) : \phi(\pi, \mathbf{p})$, **unsat** if $\forall \pi \in \mathbf{Path}(l) : \neg\phi(\pi, \mathbf{p})$ and **undet** if neither of the above two can be concluded. In general, the **undet** outcome suggests that either the chosen precision δ is not sufficient to decide the satisfiability of $\phi(\pi, \mathbf{p})$, or that $\phi(\pi, \mathbf{p})$ is undecidable (*i.e.*, non-robust [4]).

The **evaluate** procedure is crucial for building the random variables that under- and over-approximate the true system behaviour on the reachability question, as we show in the next section.

Algorithm 1: evaluate$(H, l, \mathbf{p}, \delta)$

```
input: H - PHS, l - reachability depth, p - parameter value, δ - precision;
output: sat / unsat / undet;
Path(l) = get_all_paths(H, l) ;     // compute all paths of length l for H
for π ∈ Path(l) do
    if φ(π, p) - δ-sat then
        for i ∈ [0, l] do
            if ψ_i(π, p) - δ-sat then
                return undet;
        return sat ;          // all ψ_i(π, p) are unsat for the current π
return unsat ;                                        // all φ(π, p) are unsat
```

3 Monte Carlo Probability Estimation

In this section we consider hybrid systems with random parameters only, so that the reachability probability is well-defined. We add nondeterministic parameters in the next section. For any given $\delta > 0$ and any $\mathbf{p}$ from the parameter(s) distribution we introduce the Bernoulli random variables:

$$X = \begin{cases} 1 & \text{if system } H \text{ reaches the goal in } l \text{ steps for a given } \mathbf{p} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$X_{sat} = \begin{cases} 1 & \text{if } \mathbf{evaluate}(H, l, \mathbf{p}, \delta) = \mathbf{sat} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

$$X_{usat} = \begin{cases} 0 & \text{if } \mathbf{evaluate}(H, l, \mathbf{p}, \delta) = \mathbf{unsat} \\ 1 & \text{otherwise.} \end{cases} \tag{6}$$

Thus, for a given parameter $\mathbf{p}$, X_{sat} is 1 if we can correctly decide that system H reaches the goal, while X_{usat} is 0 if we can correctly decide that H does *not* reach the goal. If no decision can be made (because of the precision δ being used or of the nature of the reachability question), X_{sat} and X_{usat} take 0 and 1, respectively. From the definition of **evaluate** it follows directly that:

$$X_{sat} \leq X \leq X_{usat} \ . \tag{7}$$

We now introduce a Bayesian technique for calculating confidence intervals for the reachability probability $p = \mathbb{E}[X]$ without sampling X, which is not possible in general, but instead sampling X_{sat} and X_{usat}. For n random variables iid (independent and identically distributed) as X_{sat} and X_{usat}, we define the random variables:

$$\hat{S}_n = \frac{\Sigma_{i=1}^{n} X_{sat,i}}{n} \qquad \hat{U}_n = \frac{\Sigma_{i=1}^{n} X_{usat,i}}{n} \quad .$$

The Bayesian approach assumes that the (unknown) reachability probability p is itself a random quantity (here we give a brief overview only, more details can be found in [17]). Bayes' theorem enables computing the *posterior* distribution of the unknown quantity given its *prior* distribution and the likelihood of the data (*i.e.*, samples of X). The posterior distribution of p can be directly used to build confidence (credibility) intervals. In our setting we cannot sample X, so we aim at bounding the posterior of p by the posteriors built from X_{sat} and X_{usat}, as we show below. We use Beta distribution priors since they are conjugate to the Bernoulli likelihood; the cumulative distribution function (CDF) of a Beta with parameters $\alpha, \beta > 0$ is denoted $F_{(\alpha,\beta)}(\cdot)$. We first need a technical lemma about the Beta CDF.

Lemma 1. *For any $n > 0$, $s \leq x \leq u \leq n$, $\alpha, \beta > 0$ ($n, s, x, u \in \mathbb{N}$), $t \in [0, 1]$ the following holds:*

$$F_{(u+\alpha,n-u+\beta)}(t) \leq F_{(x+\alpha,n-x+\beta)}(t) \leq F_{(s+\alpha,n-s+\beta)}(t) \tag{8}$$

Proof. We prove the LHS inequality of (8); the proof of the RHS follows similar steps. When $s = x$ the inequality holds trivially.

Consider the case $s < x$. By definition of the Beta distribution function:

$$F_{(s+\alpha,n-s+\beta)}(t) = \int_0^t \frac{v^{s+\alpha-1}(1-v)^{n-s+\beta-1}}{B(s+\alpha, n-s+\beta)} dv \tag{9}$$

In the proof below we refer to the following formulas from [7]:

$$B_y(a, b) = \int_0^y t^{a-1}(1-t)^{b-1}\, dt \tag{8.17.1}$$

$$I_y(a, b) = \frac{B_y(a, b)}{B(a, b)} \tag{8.17.2}$$

$$I_y(a+1, b-1) = I_y(a, b) - \frac{y^a(1-y)^{b-1}}{aB(a, b)} \tag{8.17.18}$$

By 8.17.1 and 8.17.2 the Beta distribution function (9) can be presented as an *incomplete* Beta function $I_t(s+\alpha, n-s+\beta)$ (the Beta distribution functions for the variables x and u can be written in the same form). Now we show by induction that the following holds:

$$I_t(s+\alpha, n-s+\beta) \geq I_t(x+\alpha, n-x+\beta) \tag{10}$$

As $s < x$, $s, x \in \mathbb{N}$ and $s, x > 0$ the base case is $s = 0$ and $x = 1$. Thus, we need to prove that $I_t(\alpha, n+\beta) \geq I_t(\alpha+1, (n+\beta)-1)$. By 8.17.18:

$$I_t((\alpha)+1, (n+\beta)-1) = I_t(\alpha, n+\beta) - \frac{t^\alpha (1-t)^{n+\beta-1}}{\alpha B(\alpha, n+\beta)}$$

It is easy to see that $\frac{t^\alpha (1-t)^{n+\beta-1}}{\alpha B(\alpha, n+\beta)} \geq 0$, and therefore, the base case holds.

Suppose now that $x = s+1$. By the same formula 8.17.18 [7]:

$$I_t((s+\alpha)+1, (n-s+\beta)-1) = I_t(s+\alpha, n-s+\beta) - \frac{t^{s+\alpha}(1-t)^{n-s+\beta-1}}{(s+\alpha)B(s+\alpha, n-s+\beta)}$$

As $\frac{t^{s+\alpha}(1-t)^{n-s+\beta-1}}{(s+\alpha)B(s+\alpha, n-s+\beta)} \geq 0$ the induction step holds as well. Hence, for any $s \leq x$ and $s, x > 0$ (10) holds, and the proof is complete. □

Now, Proposition 2 below tells us how to bound the posterior distribution of the unknown probability p, by using the posteriors built from X_{sat} and X_{usat}. Given n samples of X_{sat}, X_{usat} and a Beta prior with parameters $\alpha, \beta > 0$ it is easy to show that the posterior means are:

$$\hat{p}_{sat} = \frac{s+\alpha}{n+\alpha+\beta} \qquad \hat{p}_{usat} = \frac{u+\alpha}{n+\alpha+\beta} \tag{11}$$

where $s = \sum_{i=1}^{n} X_{sat,i}$ and $u = \sum_{i=1}^{n} X_{usat,i}$.

Proposition 2. *Given $\xi > 0$, the posterior probability with respect to n samples of X of the interval $[\hat{p}_{sat} - \xi, \hat{p}_{usat} + \xi]$ is bounded below as follows*

$$\mathbf{Pr}(P \in [\hat{p}_{sat} - \xi, \hat{p}_{usat} + \xi] | X_1, \ldots, X_n) \geq F_{(u+\alpha, n-u+\beta)}(\hat{p}_{usat} + \xi) - F_{(s+\alpha, n-s+\beta)}(\hat{p}_{sat} - \xi)$$

where $X_1, \ldots, X_n$ are iid as X, and $\hat{p}_{sat}$ and $\hat{p}_{usat}$ are the posterior means (11).

Proof. By definition of posterior CDF and Lemma 1:

$$\mathbf{Pr}(P \leq \hat{p}_{sat} - \xi | X_1, \ldots, X_n) \leq F_{(s+\alpha, n-s+\beta)}(\hat{p}_{sat} - \xi)$$
$$\mathbf{Pr}(P \geq \hat{p}_{usat} + \xi | X_1, \ldots, X_n) \leq 1 - F_{(u+\alpha, n-u+\beta)}(\hat{p}_{usat} + \xi)$$

and therefore

$$\mathbf{Pr}(P \in [\hat{p}_{sat} - \xi, \hat{p}_{usat} + \xi] | X_1, \ldots, X_n) =$$
$$1 - \mathbf{Pr}(P \leq \hat{p}_{sat} - \xi | X_1, \ldots, X_n) - \mathbf{Pr}(P \geq \hat{p}_{usat} + \xi | X_1, \ldots, X_n) \geq$$
$$1 - F_{(s+\alpha, n-s+\beta)}(\hat{p}_{sat} - \xi) - 1 + F_{(u+\alpha, n-u+\beta)}(\hat{p}_{usat} + \xi) =$$
$$F_{(u+\alpha, n-u+\beta)}(\hat{p}_{usat} + \xi) - F_{(s+\alpha, n-s+\beta)}(\hat{p}_{sat} - \xi)$$

□

Our algorithm is shown in Algorithm 2. Differently from SReach [13], our algorithm first uses procedure **evaluate** to compute under- *and* overapproximations of the system behaviour (line 7), and then builds upper and lower posterior probability estimates (lines 13, 14). The posterior probability of the computed interval (line 15) is guaranteed not to exceed the true posterior by Proposition 2, so when the algorithm terminates we know that the returned interval contains the true probability with the required (or a larger) confidence. Our algorithm is sequential as SReach [13], since it only stops when the desired confidence is achieved. We show its (probabilistic) termination in the next proposition.

Proposition 3. *Algorithm 2 terminates almost surely.*

Proof. Recall that Algorithm 2 generates two sequences of random variables $\{X_{sat,n}\}_{n\in\mathbb{N}}$ and $\{X_{usat,n}\}_{n\in\mathbb{N}}$. From [17, Theorem 1] we get that $X_{sat,n}$ ($X_{usat,n}$) converges a.s., for $n \to \infty$, to the *constant* random variable $\mathbb{E}[X_{sat}]$ ($\mathbb{E}[X_{usat}]$). In particular, the posterior probability of any open interval containing the posterior mean (11) must converge to 1. Therefore, the posterior probability of any interval not including the posterior mean must converge to 0.

Now, the interval $(0, \hat{p}_{usat} + \xi)$ contains the posterior mean ($\hat{p}_{usat}$) of $X_{usat,n}$ and therefore the posterior probability $F_{(u+\alpha,n-u+\beta)}(\hat{p}_{usat} + \xi)$ converges to 1. Also, the interval $(0, \hat{p}_{sat} - \xi)$ does not contain the mean ($\hat{p}_{sat}$) of $X_{sat,n}$, so $F_{(s+\alpha,n-s+\beta)}(\hat{p}_{sat} - \xi)$ tends to 0, and this concludes the proof. □

Algorithm 2: Bayesian Estimation Algorithm

```
1  input: system H, δ - solver precision, l-reachability depth, c - confidence, ξ -
   accuracy, α, β - Beta distribution parameters;
2  output: confidence interval with posterior probability not smaller than c;
3  n = 0; s = 0; u = 0; v = 0;
4  repeat
5      p = get_random_sample();            // sample the initial parameters
6      n = n + 1;
7      switch evaluate(H, l, p, δ) do              // δ-complete evaluation
8          case sat
9              s = s + 1;
10         case unsat
11             v = v + 1;
12     u = n − v;
13     p̂_sat = (s+α)/(n+α+β); p̂_usat = (u+α)/(n+α+β);   // posterior means for X_sat,n, X_usat,n
       // calculate confidence
14     p̂_sat = max(ξ, p̂_sat); p̂_usat = min(1 − ξ, p̂_usat);
15     p = F_(u+α,n−u+β)(p̂_usat + ξ) − F_(s+α,n−s+β)(p̂_sat − ξ);
16 until p ≥ c;
17 return [p̂_sat − ξ, p̂_usat + ξ];
```

In the next section we extend our technique to hybrid systems that feature *both* nondeterministic and random parameters.

4 Cross-Entropy Algorithm

We perform probabilistic reachability analysis for hybrid systems featuring both random and nondeterministic parameters by solving an optimisation problem aimed at finding parameter values for which the system achieves maximum (minimum) reachability probability. We present an algorithm (Algorithm 3) based on the cross-entropy (CE) method [10], a powerful stochastic technique for solving estimation and optimisation problems. The main idea behind the CE method is obtaining the optimal parameter distribution by minimizing the distance between two probability density functions. The cross-entropy (or Kullback-Leibler divergence) between two probability density functions g and f is:

$$\Theta(g, f) = \int g(\boldsymbol{\lambda}) \ln \frac{g(\boldsymbol{\lambda})}{f(\boldsymbol{\lambda})} d\boldsymbol{\lambda} \ .$$

The CE is nonnegative and $\Theta(g, f) = 0$ iff $g = f$, but it is not symmetric ($\Theta(g, f) \neq \Theta(f, g)$), so it is not a distance in the formal sense.

The optimisation problem solved by the CE method can be formulated as the following: given a family of densities $\{f(\cdot; \mathbf{v})\}_{\mathbf{v} \in V}$ find the value $v^* \in V$ that minimizes $\Theta(g^*, f(\cdot; \mathbf{v}))$ (where g^* is the optimal density). The CE method comprises two general steps: 1) generating random samples from some initial distribution; 2) updating the distribution based on the obtained samples in order to obtain better samples in the next iteration. Note that for solving optimisation problems it is necessary that the family $\{f(\cdot; \mathbf{v})\}_{\mathbf{v} \in V}$ contains distributions that can approximate arbitrarily well single-point distributions.

In Algorithm 3 we use a parametrized family of normal distributions $f(\boldsymbol{\lambda}; \mathbf{v})$ (the first element of $\mathbf{v}$ is the mean and the second element is the standard deviation). Initially the standard deviation should be relatively large in order to sample a larger space on the first iteration of the algorithm. Let D be the definition domain of the nondeterministic parameters (obtained by projecting the hybrid system parameter space P over the nondeterministic parameters only). Starting with $\mathbf{v}_0 = \{\mu_0, \sigma_0\}$ such that μ_0 is the center of D and each element of σ_0 is half-size the corresponding interval from D the algorithm draws s samples from $f(\boldsymbol{\lambda}|\mu_0, \sigma_0)$ and evaluates them using the *sample performance function*:

$$P(\boldsymbol{\lambda}) = \begin{cases} \text{probability that } H(\boldsymbol{\lambda}) \text{ reaches the goal} & \text{if } \boldsymbol{\lambda} \in D \\ -\infty & \text{otherwise.} \end{cases}$$

To compute $P(\cdot)$ we run Algorithm 2 and take the mid point of the returned interval. Note that when solving probability minimization problems the second option in the definition of $P(\cdot)$ should be changed to ∞.

Given a number of samples, it is easy to see that as the number of nondeterministic parameters increases, the more difficult it becomes to draw samples lying inside of D. In fact, given n nondeterministic parameters the probability that a sample $\boldsymbol{\lambda}$ belongs to D is equal to:

$$\mathbf{Pr}(\boldsymbol{\lambda} \in D) = \prod_{j=1}^{n} \int_{D_j} f(\lambda_j | \mu_j, \sigma_j) d\lambda_j \tag{12}$$

where D_j is the domain of the j-th parameter, and we assumed that the parameters are sampled independently. Hence, in order to increase the likelihood that s samples lie in D it is sufficient to generate $s^* = \lceil \frac{s}{\eta} \rceil$ samples, where $\eta = \mathbf{Pr}(\boldsymbol{\lambda} \in D)$ is obtained using (12). The performance of each sample is then evaluated, and the samples are sorted in descending order (ascending in the case of probability minimization) according to their performance value. We label a number $k = \lceil \rho s^* \rceil$ of them as *elite* samples, where $\rho \in [10^{-2}, 10^{-1}]$ is a positive constant chosen by the user. The set of elite samples E is then used for updating the distribution parameters μ_i and σ_i on the i-th iteration of the algorithm using the formulas from [10, Chapter 8.7]:

$$\begin{aligned} \mu_i &= \frac{\sum_{j \in [1,k]} E_j}{k} \\ \sigma_i &= \sqrt{\frac{\sum_{j \in [1,k]} (E_j - \mu_i)^2}{k}} \end{aligned} \tag{13}$$

The algorithm terminates when the largest element of vector σ reaches a user-defined precision $\hat{\sigma}$, and it outputs the estimated maximum reachability probability $\mathbf{P}$ and a (nondeterministic) parameter value $\boldsymbol{\lambda}$ for which $P(\boldsymbol{\lambda}) = \mathbf{P}$.

5 Experiments

We apply our algorithms to three models (two of which are hybrid), a model of irradiation therapy for psoriasis [16], a car collision scenario and a model of human starvation [12]. The algorithms have been implemented in C++, and the experiments have been carried out on a 32-core (2.9GHz) Linux machine.

UVB Irradiation Therapy. We consider a simplified version of a hybrid UVB irradiation therapy model [16] used for treating psoriasis, an immune system-mediated chronic skin condition which is characterised by overproduction of keratinocytes. The simplified model comprises of three (six in the original model) categories of normal and three (five in the original model) categories of psoriatic keratinocytes whose dynamics is presented by nonlinear ODEs. The therapy consists of several episodes of UVB irradiation, which is simulated in the model by increasing the apoptosis rate constants (β_1 and β_2) for stem cells (SC) and transit amplifying (TA) cells by In_A times. Every such episode lasts for 48 hours and is followed by 8 hours of rest ($In_A = 1$) before starting the next irradiation. The efficiency of the therapy depends on the number of alternations between the irradiation and rest stages. An insufficient number of treatment episodes can result into early psoriasis relapse: The deterministic version of this model predicts psoriasis relapse for the number of therapy episodes less than seven [16]. We consider the parameter In_A characterising the therapy strength to be normally distributed with mean value $6 \cdot 10^4$ and standard deviation 10^4 and $\lambda \in [0.2, 0.5]$ characterising the strength of psoriatic stem cells to be nondeterministic, and we calculate the maximum and the minimum probabilities of psoriasis relapse within 2,000 days after the last therapy episode for nine alternations ($l = 9$) between the 'on' and 'off' therapy modes (five therapy cycles). The results (Table

Algorithm 3: Cross-Entropy Algorithm

input: hybrid system H, δ - solver precision, l - reachability depth, α, β - Beta distribution parameters, c - confidence, ξ - accuracy, s - sample size, ρ - elite samples ratio, $\hat{\sigma}$ - maximum variance ;
output: (parameter value, maximum probability) ;
$\mu = \{\frac{\min(D_1)+\max(D_1)}{2}, \cdots, \frac{\min(D_n)+\max(D_n)}{2}\}$;
$\sigma = \{\frac{|D_1|}{2}, \cdots, \frac{|D_n|}{2}\}$; $\sigma' = \sigma$;
while $(\max_{1 \leq j \leq n} \sigma'_j) > \hat{\sigma}$ **do**
    $\eta = \prod_{j=1}^{n} \int_{D_j} f(x_j|\mu_j, \sigma_j) dx_j$;
    $m = \lceil \frac{s}{\eta} \rceil$; $k = \lceil \rho \frac{s}{\eta} \rceil$; `// adjusting sample size`
    for $i = 1 : m$ **do**
        $\boldsymbol{\lambda} = get\_random\_normal\_sample()$;
        if $\boldsymbol{\lambda} \notin D$ **then**
            $\mathbf{P} = [-\infty, -\infty]$;
        else
            $\mathbf{P} = \text{mid}(\textbf{bayes}(H(\boldsymbol{\lambda}), \delta, l, \alpha, \beta, \xi, c))$; `// applying Algorithm 2`
        $Q.push(\boldsymbol{\lambda}, \mathbf{P})$;
    sort(Q) ; `// sorting in descending order by the probability value`
    $res = Q[1]$; `// updating the result`
    $\mu = \frac{\sum_{i \in [1,k]} Q[i]}{k}$; `// updating the mean`
    $\sigma' = \sigma$; `// saving current value of standard deviation`
    $\sigma = \sqrt{\frac{\sum_{i \in [1,k]} (Q[i]-\mu)^2}{k}}$; `// updating the standard deviation`
    clear(Q);
return res;

1) show that the estimated maximum probability lies in the interval $[0.8268, 1]$ for $\lambda = 0.4953$ and the minimum probability is in the interval $[0, 0.1086]$ for $\lambda = 0.1303$. Algorithm 3 required two iterations in both cases and generated 24 (out of total 26) and 23 (out of 26) samples from the domain of nondeterministic parameters D.

Table 1. UVB irradiation therapy: results with $\xi = 10^{-1}$, $c = 0.99$, $\delta = 10^{-3}$, $\rho = 10^{-1}$, $s = 10$ and $\hat{\sigma} = 10^{-2}$, where λ – estimated value of nondeterministic parameter, μ_λ and σ_λ – mean and standard deviation of the resulting distribution, CI – confidence interval returned, s_R – total number of random samples used during s_N^* executions of Algorithm 2, s_N – total number of nondeterministic samples, s_N^* – number of nondeterministic samples drawn from D, i – number of iterations of Algorithm 3, **Time** – CPU time in seconds.

λ	μ_λ	σ_λ	CI	s_R	$s_N(s_N^*)$	**Time**
0.4953	0.4878	0.0089	[0.8268,1]	3,118	26(24)	13,492
0.1303	0.1347	0.0079	[0,0.1086]	2,880	26(23)	12,550

Cars Collision Scenario. We consider a taking over and deceleration scenario modelled as a hybrid system. Initially two cars are moving with speed $v_{A0} = v_{B0} = 11.12\ m/s$ at a distance $v_A \cdot \tau_{safe}$ from each other, where $\tau_{safe} \in [1, 2]\ s$ is nondeterministic. In the initial mode $CarA$ changes the lane and starts accelerating until it gets ahead of $CarB$ by $v_B \cdot \tau_{safe}$ meters. After that $CarA$ changes the lane back and starts decelerating with normally-distributed acceleration $a_{dA} \sim N(-2, 0.2)$. The driver in $CarB$ reacts within $\tau_{react} \in [0.5, 1.5]\ s$ and starts decelerating with acceleration $a_{dB} \sim N(-1.35, 0.1)$ until both cars stop completely.

The model contains three modes: $CarA$ overtakes $CarB$, $CarA$ decelerates while $CarB$ keeps moving for τ_{react} second, and both cars decelerate until they stop. There are two nondeterministic (τ_{safe} and τ_{react}) and two random (a_{d1} and a_{d2}) parameters in the system. We aim at determining whether there is a non-zero probability of the cars colliding ($l = 2$).

Table 2. The minimum probability for the cars collision scenario with $\xi = 5 \cdot 10^{-3}$, $c = 0.99$, $\delta = 10^{-3}$, $\rho = 10^{-1}$ and $\hat{\sigma} = 10^{-1}$, where τ_{react} – $CarB$ driver reaction time, τ_{safe} – time interval between the cars, $\mu_{\tau_{react}}$ and $\sigma_{\tau_{react}}$ – mean and standard deviation of the resulting distribution for τ_{react}, $\mu_{\tau_{safe}}$ and $\sigma_{\tau_{safe}}$ – mean and standard deviation of the resulting distribution for τ_{safe}, CI – confidence interval returned, s_R – total number of random samples used during s_N^* executions of Algorithm 2, s_N – total number of nondeterministic samples, s_N^* – number of nondeterministic samples drawn from D, **Time** – CPU time in seconds.

τ_{react}	$\mu_{\tau_{react}}$	$\sigma_{\tau_{react}}$	τ_{safe}	$\mu_{\tau_{safe}}$	$\sigma_{\tau_{safe}}$	CI	s_R	s	$s_N(s_N^*)$	**Time**
0.609	0.619	0.011	1.791	1.753	0.019	[0.0252,0.0352]	658,528	10	43(32)	18,005
0.522	0.583	0.077	1.953	1.795	0.079	[0.0121,0.0221]	952,057	20	90(57)	27,126

We apply Algorithm 3 to this model with different values of s, the CE sample size. The obtained results (Table 2) confirm that choosing smaller values of τ_{react} and larger values of τ_{safe} decreases the probability value. Also, choosing a larger s increases the accuracy of the obtained result from $P(0.609, 1.791) = [0.0252, 0.0352]$ for $s = 10$ to $P(0.522, 1.953) = [0.0121, 0.0221]$ for $s = 20$. The execution of the algorithm took three iterations in both cases drawing 32 (out of 43) and 57 (out of 90) samples lying in D for $s = 10$ and $s = 20$ respectively.

Human Starvation. The human starvation model [12] tracks the amount of fat (F), protein in muscle mass (M), and ketone bodies (K) in the human body after glucose reserves have been depleted from three to four days of fasting. These three variables are modelled using material and energy balances to ensure that the behaviour of the model tracks what is observed in actual experiments involving fasting. Randomising two model parameters we evaluate the probability of a 40% decrease in the muscle mass by the τ_g's day of fasting where $\tau_g \in [20, 27]$ is a nondeterministic parameter. The reachability depth value l is 0. The results (Table 3) demonstrate that the maximum probability of losing 40% of the muscle mass is within the interval $[0.99131, 1]$ for $\tau_g = 26.47$ and the minimum probability is inside $[0, 0.0057]$ for $\tau_g = 20.22$. The execution of the algorithm took three iterations in both cases drawing 31 (out of 37) and

34 (out of 36) samples from D for calculating the minimum and the maximum probabilities respectively.

Table 3. The minimum and the maximum reachability probabilities for the human starvation model with $\xi = 5 \cdot 10^{-3}$, $c = 0.99$, $\delta = 10^{-3}$, $\rho = 10^{-1}$, $s = 10$ and $\hat{\sigma} = 10^{-1}$, where τ_g – time (days) from the beginning of fasting, μ_{τ_g} and σ_{τ_g} – mean and standard deviation of the resulting distribution, CI – confidence interval returned, s_R – total number of random samples used during s_N^* executions of Algorithm 2, s_N – total number of nondeterministic samples, s_N^* – number of nondeterministic samples drawn from D, **Time** – CPU time in seconds.

τ_g	μ_{τ_g}	σ_{τ_g}	CI	s_R	$s_N(s_N^*)$	**Time**
20.2264	20.2125	0.068	[0,0.0057]	408,061	37(31)	2,703
26.4713	26.5146	0.033	[0.99131,1]	485,721	36(34)	4,360

Discussion. From our results we see that the chosen value of δ did not affect the length (2ξ) of the returned confidence intervals in any experiment. Also choosing a larger number of samples per iteration (s) in Algorithm 3 and a higher precision (ξ) for Algorithm 2 increases the accuracy of the obtained result. The sample size adjustment in Algorithm 3 increases the likelihood of drawing the desired number of samples from the domain of nondeterministic parameters. For example, in the cars collision scenario featuring two nondeterministic parameters almost a third of all drawn samples were outliers. However, the desired number of samples belonging to the domain of nondeterministic parameters was still provided. Finally, the performance of Algorithm 2 and Algorithm 3 significantly depends of the complexity of the system's dynamics. For example, the UVB irradiation therapy model is more complex in comparison to other two models. As a result, Algorithm 1 required more CPU time for evaluating each pair (random and nondeterministic) of samples.

Implementation. All algorithms presented in this paper were implemented in our tool `ProbReach` [11], which can be downloaded from `https://github.com/dreal/probreach`. We also used `dReal` [5] as an SMT solver (δ-complete decision procedure). The models used in this section can be found at `https://github.com/dreal/probreach/tree/master/model/hvc2016`.

6 Conclusions and Future Work

We introduce novel Monte Carlo (*i.e.*, statistical) techniques for computing both numerically *and* statistically rigorous confidence intervals for bounded reachability probability in hybrid systems with random and nondeterministic parameters. To enable formal numerical reasoning we employ δ-complete SMT decision procedures, and we combine them with sequential Bayesian estimation and the cross-entropy method. We exploit δ-complete procedures to build under- and over-approximations of the reachability probability. We prove the correctness of such approximations, the statistical validity of our techniques, and termination of our Bayesian algorithm. Our techniques compute confidence intervals that are formally and statistically correct *independently* of the numeric precision (δ) used. This offers users the choice of trading accuracy of the returned interval for

computational cost, thereby enabling faster verification. Our experiments with highly nonlinear hybrid systems show that our techniques are useful in practice.

For future work, understanding the relation between the numerical precision (δ) and the returned interval size is an important avenue of research. Also, we plan to extend the range of models analysable (*e.g.*, probabilistic jumps and stochastic differential equations).

Acknowledgements. F.S. was supported by award N00014-13-1-0090 of the US Office of Naval Research; P.Z. was supported by EPSRC grant EP/N031962/1.

References

1. R. Alur, C. Courcoubetis, T. A. Henzinger, and P.-H. Ho. Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems. In *Hybrid Systems*, volume 736 of *LNCS*, pages 209–229, 1992.
2. E. M. Clarke and P. Zuliani. Statistical model checking for cyber-physical systems. In *ATVA*, volume 6996 of *LNCS*, pages 1–12, 2011.
3. C. Ellen, S. Gerwinn, and M. Fränzle. Statistical model checking for stochastic hybrid systems involving nondeterminism over continuous domains. *International Journal on Software Tools for Technology Transfer (STTT)*, 17(4):485–504, 2015.
4. S. Gao, J. Avigad, and E. M. Clarke. Delta-decidability over the reals. In *LICS*, pages 305–314, 2012.
5. S. Gao, S. Kong, and E. M. Clarke. dReal: An SMT solver for nonlinear theories over the reals. In *CADE-24*, volume 7898 of *LNCS*, pages 208–214, 2013.
6. S. Gao, S. Kong, and E. M. Clarke. Satisfiability modulo ODEs. In *FMCAD*, pages 105–112, 2013.
7. F. W. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark. *NIST Handbook of Mathematical Functions*. Cambridge University Press, 1st edition, 2010.
8. A. Platzer and E. M. Clarke. The image computation problem in hybrid systems model checking. In *HSCC*, volume 4416 of *LNCS*, pages 473–486, 2007.
9. D. Richardson. Some undecidable problems involving elementary functions of a real variable. *J. Symb. Log.*, 33(4):514–520, 1968.
10. R. Y. Rubinstein and D. Kroese. *Simulation and the Monte Carlo Method*. Wiley, 2008.
11. F. Shmarov and P. Zuliani. ProbReach: Verified probabilistic δ-reachability for stochastic hybrid systems. In *HSCC*, pages 134–139. ACM, 2015.
12. B. Song and D. Thomas. Dynamics of starvation in humans. *Journal of Mathematical Biology*, 54(1):27–43, 2007.
13. Q. Wang, P. Zuliani, S. Kong, S. Gao, and E. M. Clarke. SReach: A bounded model checker for stochastic hybrid systems. In *CMSB*, volume 9308 of *LNCS*, pages 15–27, 2015.
14. H. L. S. Younes, M. Z. Kwiatkowska, G. Norman, and D. Parker. Numerical vs. statistical probabilistic model checking. *STTT*, 8(3):216–228, 2006.
15. H. L. S. Younes and R. G. Simmons. Statistical probabilistic model checking with a focus on time-bounded properties. *Inf. Comput.*, 204(9):1368–1409, 2006.
16. H. Zhang, W. Hou, L. Henrot, S. Schnebert, M. Dumas, C. Heusèle, and J. Yang. Modelling epidermis homoeostasis and psoriasis pathogenesis. *Journal of The Royal Society Interface*, 12(103), 2015.
17. P. Zuliani, A. Platzer, and E. M. Clarke. Bayesian statistical model checking with application to Stateflow/Simulink verification. *Formal Methods in System Design*, 43(2):338–367, 2013.