Quantitative Risk Assessment of Safety-Critical Systems via Guided Simulation for Rare Events

Stefan Puch[2(✉)], Martin Fränzle[1(✉)], and Sebastian Gerwinn[2]

[1] Carl von Ossietzky Universität Oldenburg, 26111 Oldenburg, Germany
fraenzle@informatik.uni-oldenburg.de
[2] OFFIS e.V., Escherweg 2, 26121 Oldenburg, Germany
puch@offis.de

Abstract. For developers of assisted or automated driving systems, gaining specific feedback and quantitative figures on the safety impact of the systems under development is crucial. However, obtaining such data from simulation of their design models is a complex and often time-consuming process. Especially when data of interest hinge on extremely rare events, an estimation of potential risks is highly desirable but a non-trivial task lacking easily applicable methods. In this paper we describe how a quantitative statement for a risk estimation involving extremely rare events can be obtained by guiding simulation based on reinforcement learning. The method draws on variance reduction and importance sampling, yet applies different optimization principles than related methods, like the cross-entropy methods against which we compare. Our rationale for optimizing differently is that in quantitative system verification, a sharper upper bound of the confidence interval is of higher relevance than the total width of the confidence interval.

Our application context is deduced from advanced driver assistance system (ADAS) development. In that context virtual driver simulations are performed with the objective to generate quantitative figures for the safety impact in pre-crash situations. In order to clarify the difference of our technique to variance reduction techniques, a comparative evaluation on a simple probabilistic benchmark system is also presented.

1 Introduction

The global volume of road traffic is growing faster than ever. This contrasts with the ongoing effort to reduce the number of deadly injured people in road traffic. The EU commission announced the ambitious target of halving the overall number of road deaths in the EU by 2020 starting from 2010 at a number of 27.000 [4]. But Eurostat, the statistical office of the EU, states at a total number of 26.100 people who died in road accidents in 2016, which indicate that it is still a long

This research was supported by the Ministry of Science and Culture of Lower Saxony within the research center Critical Systems Engineering for Sociotechnical Systems.

T. Margaria and B. Steffen (Eds.): ISoLA 2018, LNCS 11245, pp. 305–321, 2018.
https://doi.org/10.1007/978-3-030-03421-4_20

way to their target [5]. Many research institutes and the automotive industry are working hard on new Advanced Driver Assistance Systems (ADAS) in particular for the pre-crash phase to reduce the number of traffic victims. While some emergency braking systems from different car manufacturers are already available on the market, harmonized development methods for design, evaluation and assessment of pre-crash systems, which should speed up the development process, are still nascent [11]. Harmonized methods within a model-based design approach shall support the ADAS developer and ensure that the final implementation meets its safety target, thus leading to shorter time-to-market. While exhaustive formal verification of ADAS and their interaction with a human driver is far out of scope due to their complex model structures, which overburden current formal verification frameworks both with respect to the expressiveness of the modeling languages supported and to scalability, a simulation-based approach can in principle be used to validate an assistance system and provide a quantitative estimation of potential risks[1]. The extreme scarceness of actually hazardous situations in human-operated road traffic (e.g., more than 1.64×10^6 km between accidents involving human injuries according to [16]), however, requires adequate identification and statistical treatment of extremely rare situations, which can be achieved by criticality-driven guided simulation [13]. Within this paper, we add quantitative error margins to the quantitative figure provided by such guided simulation. We furthermore demonstrate the superiority compared to naive sampling, especially concerning tightness of the upper bounds of the confidence intervals as relevant to statistical model checking. Additionally we compare different guiding strategies within a rare event simulation by benchmarking them against each other on a simple hybrid-state probabilistic process. The aim is to characterize the performance of different guiding algorithms, providing a rationale for selecting the most appropriate algorithm.

2 Background and Related Work

Estimating rare event frequencies in complex domains is a frequent problem in empirical evaluations. Established approaches employ variance reduction techniques [10] avoiding intractable scaling with respect to the number of samples necessary to characterize rare events. In simulation-based studies, methods like importance sampling, importance splitting, etc., render rare events more likely than in direct Monte Carlo (MC) simulation, because the sample size, i.e., the number of simulations grows too large when the event probability tends to zero. The individual research contributions, however, differ in their application contexts and the transferability to other domains.

In this paper we focus on approaches based on importance sampling (IS). The basic idea of importance sampling is to draw the samples according to a proposal distribution rather than their native distribution and re-normalize the statistics obtained afterwards using importance weights. The expectation $E[g]$

[1] Like in formal verification we have to assume the model used for simulation is correct.

of a random variable g estimated by N samples using importance sampling is

$$\hat{E}[g] = \frac{1}{N} \sum_{i=1}^{N} g(X_i)\ w(X_i),$$

where $w(x) = p(x)/q(x)$ is the likelihood ratio with $p(x)$ being the original probability of the sample x and $q(x)$ the probability assigned to the sample when it is generated according to the proposal instead.

The most challenging problem of IS is to find a proposal q such that the variance of the IS estimator is significantly smaller than the variance from pure MC estimation. In [7,8] the authors present different variants of adaptive importance sampling (AIS) for the validation of ADAS illustrated on a simple adaptive cruise control problem. All variants have in common that they draw an initial number of N samples (a batch) before they derive an adapted proposal distribution based on a kernel density estimator. A problem of the approach is that he indicator function used to determine whether a critical event has occurred is only interpreted in a binary way (true or false). Consequently, potential information about the closeness to the rare event cannot be used and adaptation stays uninformed till the first random hits of the rare event. Despite this weakness of the approach, the authors demonstrate that AIS can increase simulation efficiency roughly a tenfold in their problem context.

The work of Zuliani et al. in [20] presents an approach exploiting the cross-entropy (CE) method [14] for generating approximately optimal biasing densities for statistical model checking of hybrid systems. Their approach comprises two steps: First they use the CE to determine a proposal density function which empirically minimizes the Kullback-Leibler divergence to the optimal proposal density. Then importance sampling with that proposal is performed to estimate the expectation $E[g]$ of a random variable g. In order to demonstrate that the proposed method is applicable to stochastic hybrid systems, the authors applied the cross-entropy method to a Simulink-Stateflow example of a fault-tolerant avionics system. It is shown that by increasing the sample size, the relative error (RE) decreases and that with a feasible sample size of 10^4 it is possible to estimate probabilities in the order of 10^{-14} with reasonable accuracy (RE = 0.24). Although CE provides a theoretical basis for selecting proposal distributions adaptively, the effectiveness of such an approach depends heavily on well chosen parameterization of the proposal distributions and additional algorithmic parameters such as batch-size and an appropriately guessed "tilting parameter" providing an initial proposal yielding informative rare-event rates in step 1.

Both the aforementioned approaches do draw on empirical estimates of the variance or cross-entropy obtained from a binary evaluation (satisfaction or violation of a requirement by a sampled trace) of an initial batch of samples, which likely remains uninformative in the case of extremely rare events, which have to be found first before that statistics becomes informative. The focus of our work reported here in contrast is on means helping to find such rare events even in an initial batch. To this end, we employ a continuous approximation of the binary trace evaluation that statistical model checking targets and exploit this approximation in guiding the simulation. Such continuous approximations can

either be derived from continuous interpretations of temporal logic [2,6] or from risk functions known from traffic psychology [15].

Jegourel et al. in [9] present an importance-sampling framework combining symbolic analysis with simulation to estimate expected costs in stochastic priced timed automata (SPTA). The framework is integrated into UPPAL SMC. Its first step is a symbolic reachability analysis in order to identify states never leading to trace completions satisfying the desired property. This is feasible as SPTA, in contrast to stochastic hybrid automata, have a decidable qualitative reachability problem which can be represented as a zone-based graph permitting identification of such "dead end" states. In a second step, that knowledge is exploited for pruning expansion of such states in the simulations underlying statistical model checking (SMC). This reduces variance compared to crude Monte-Carlo (MC) simulation as all simulations only expand potentially satisfying states. To estimate effectiveness of the approach the authors compare the empirical variance with that of direct MC simulation. While the empirical variance typically is reduced, the method induces considerable overhead for set-up, state-exploratory analysis of models, and additional storage and simulation costs.

The method does unfortunately not transfer to our problem domain as it, first, would require a full white-box model of the ADAS and environment not normally available when OEMs or tier-1 suppliers cooperate with subordinate suppliers in automotive and, second, as SPTA are not expressive enough to model the full-fledged feedback dynamics involving non-linear system dynamics, non-linear control, and human cognition. The UPPAAL benchmarks provided do also feature a very limited number of discrete locations (some tens of locations) which is considerably below the enormous size of the discrete state-space spanned by cognitive architectures [18] as used in our setting.

3 Application Context

As a specific application context we are interested in estimating the probability of causing a critical situation as a result of the cognitive load induced by cooperation with an advanced driver assistance system (ADAS) in automobiles. This is a crucial question in ADAS design, as the expected positive safety impact of such a system may easily become negated by additional cognitive load induced by the ADAS. Such cognitive load stems from effects like disturbance and distraction, effort for interpretation of system reactions and interventions, effort for mode tracking, or even mode confusion, all of which are standard side effects of assistance and automation. Hazardous effects induced by such systems are, however, a small additive risk and thus at least as rare as fatal hazards in normal driving. Without appropriate importance sampling, model-based simulation studies, as in Monte Carlo statistical model checking, are consequently bound to fail due to the excessive number of simulation traces necessary for a reasonable statistics. The problem with applying importance sampling is that it is in general unclear how to modify proposal probabilities in order to enhance the rare-event statistics in these settings: disturbances by the ADAS, e.g., will only impact safety

if occurring at very specific moments, as the human driver (or its substitution by a validated cognitive model) generally is very effective in canceling out temporary deviations from an optimal track. The problem thus is to find and then emphasize in probability those few situations where overall risk is sensitive to interaction with the ADAS.

If we succeed in finding such a proposal distribution, then importance sampling improves our statistics by investigating more samples in "interesting" regions of the sample space. If the goal is enhanced accuracy of the estimated expectation of a random variable, where enhanced accuracy is interpreted as a narrow confidence interval, then the way to go with the proposal distribution is variance reduction. Techniques like adaptive importance sampling under the cross-entropy method or importance splitting address this issue with different algorithmic means.

It should be noted that improving accuracy of an expectation estimate is correlated with, yet not identical to improving the reliability of the related SMC-based quantitative safety verdict: in statistical model checking, we exploit a confidence interval $E \in [a, b]$ with confidence c, where E is the expectation/probability of an outcome violating the requirement specification, to decide with confidence c whether $E \leq \theta$ for a safety target θ. For answering this question, only the upper bound b of the confidence interval is of importance; a confidence interval $E \in [a', b']$ with $b' < b$ would thus convey more information even it were wider than $[a, b]$. We conclude that variance reduction is not necessarily the most effective mode of designing a proposal distribution in importance sampling and design two experiments for benchmarking a reinforcement learning approach more greedily searching for samples violating the safety specification. The benchmarks are as follows:

Cognitive Driver Model in the Loop: In this example, we set up a heterogeneous co-simulation comprising a cognitive architecture instantiated to simulate a human car driver, an off-the-shelf interactive driving simulator providing real-time simulation and rendering of driving dynamics and environment, and a side task representative of ADAS distraction (see [13] for details). The cognitive driver model contains a variety of sub-components ranging from models of perception and motoric action, short-term memory for perceived items, long-term memory for procedural knowledge, driving skills at the control-theoretic layer modeled by differential or difference equations, to rule-based behavior recursively decomposing complex tasks into conditional sub-tasks and finally skills. It has been validated against extensive sets of observed behavior from 17 human drivers [18,19]. In our simulation scenario, it is driving along 1,1 km of a winding road with curve radii between 375 m and 750 m and has the obligation to keep track and a target speed close to the speed limit. The environment for this driving scenario was modeled in the interactive driving simulator SILAB [17] which provides real-time visualization of the environment, visualization of the road, environmental traffic (not used in this experiment), and an interactive car model incorporating a realistic car kinematics which the virtual driver model then steers. During simulation the attention of the driver (model) has

to alternate between three competing goals: (1) Keep the car in the middle of the lane. (2) Keep a constant speed of 100 km/h as closely as possible. (3) As soon as possible solve some side tasks displayed in varying time intervals on an in-vehicle display at the center console. The third goal is a typical proxy used by cognitive psychologists as a representative for interactions of the driver with an ADAS installed in the center console of the car.

To meet the requirements of all goals, the visual and cognitive attention has to alternate between these three tasks and their respective areas of interest (road through windscreen and mirrors, speedometer, in-vehicle display). If insufficient attention is paid to keeping the car within the lane, the driver might cross the lane border which might lead to critical situations. A highly critical point within the scenario was added by placing a bridge over the road (see Fig. 1). The pillar of the bridge is placed 2.5 m away from the center of the right lane, which therefore corresponds to the expected distance between car and pillar when the car is passing the bridge during normal drive. We learned from naïve sampling using a pure MC strategy that the distance between the car and the bridge pillar was above 2.4 m in 7,272 out of 10,000 runs and furthermore that nearly all (namely 9994 of 10,000) deviations from the middle of the lane stayed well within the lane boundaries, irrespective of the driver model being distracted by performing the secondary task. The closest distance to the bridge pillar which could be observed during the whole simulation batch of 10,000 simulation runs was about 0.7 m, which is still far from a hit of the pillar and occurred only two times in 10,000 runs. The likelihood that corrective actions by the driver saves the situation after a distraction thus is overwhelmingly high; so high indeed that a simulation time of 1 week, which the 10,000 runs amounts to, cannot reveal a single accident (not even a near-accident) caused by the side task representing ADAS interaction. Pure MC simulation consequently is inapt of quantifying the safety impact of ADAS interaction in this rather typical traffic scenario. Taking as a verification goal the Signal-Temporal-Logic-like [2] formula

$$\Box(||(x, y) - (p_x, p_y)|| > 0.5\,\mathrm{m}), \tag{1}$$

where x and y represent the current longitudinal and lateral position of the car and p_x and p_y the corresponding positions of the bridge pillar, naïve statistical model checking would after a week simulation time estimate the likelihood of violation as zero. Unfortunately, this does also mean that such a simulation batch would remain completely uninformative for adaptive importance-sampling.

In our setting, we instead added a simulation guide into the simulation framework that employs a by-now standard continuous interpretation [2,6] of formula (1), namely the minimum over time (due to the $\Box$ operator) of the distance to the bridge pillar (due to term $||(x, y) - (p_x, p_y)||$) minus the —in the context of minimization irrelevant— offset 0.5 m, as a continuous objective function to be minimized. Such minimization then is achieved by modifying the probabilities associated to the various probabilistic elements of the cognitive driver model, which are introduced to reflect human behavior in a psychologically plausible way. These probabilistic elements serve to emulate the variations in human

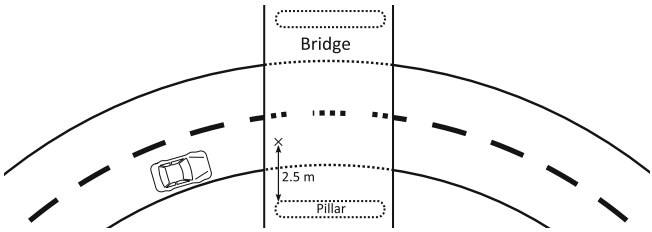

Fig. 1. Distance of bridge pillar to the center of the lane in the driving scenario

behavior which were observed when performing simulator studies with human drivers. They go down to the level of deciding at a rate of 20 Hz between options for gaze attention. Such decision are taken in a goal-directed manner, yet are far from deterministic. They exhibit stochasticity, with the mutual probabilities of the options being assigned situationally based on cognitive priorities between competing processes. The number of probabilistic decisions taken during a car ride of 1,1 km thus is enormous, and it is a search for the needle in the haystack to identify those which actually impact safety. For a deeper look into the underlying concepts and the architecture of the cognitive driver model itself, the reader is referred to [12].

The strategy of the simulation guide is to increase the probability of situations which lead to small distances between the car and the bridge pillar by applying reinforcement learning by the TUTS algorithm [13] explained in Sect. 4. An evaluation of another 10,000 runs using TUTS in the scenario demonstrated that nearly 10% of the simulation runs had a distance smaller than 0.5 m to the bridge pillar, thus being highly critical and violating formula (1). After thus improving the rate of critical situations revealed, we are able to derive a reasonable statistics and thus a quantitative risk statement in a subsequent step, see Sect. 4.

Randomly Bouncing Ball: Since the above simulation setup is rather complex and since it seems useful to compare the approach to other guiding strategies as well as to pure Monte Carlo sampling, we compared different approaches on a much simpler benchmark where we can compute the ground truth and its variance along the stochastic elements. Therefore we took a simple stochastic bouncing ball which starts from an initial height falling down towards a reflective surface. When hitting the floor, the rebounce of the ball is scattered due to a rebound angle varying stochastically within a fixed range. Thus the ball can bounce along an axis in a fixed direction but with different heights and horizontal speeds in between resulting from the varying modes of deflection (see Fig. 2).

The ballistic curve of the ball is defined by following the equation:

$$x(t) = x(0) + vt\cos(\theta) \quad y(t) = y(0) + vt\sin(\theta) - \frac{1}{2}gt^2, \tag{2}$$

where θ and v are given by the initial velocity vector $\boldsymbol{v}_0$ as follows:

$$\theta = \arctan(\boldsymbol{v}_0) \qquad v = ||\boldsymbol{v}_0||_2 \tag{3}$$

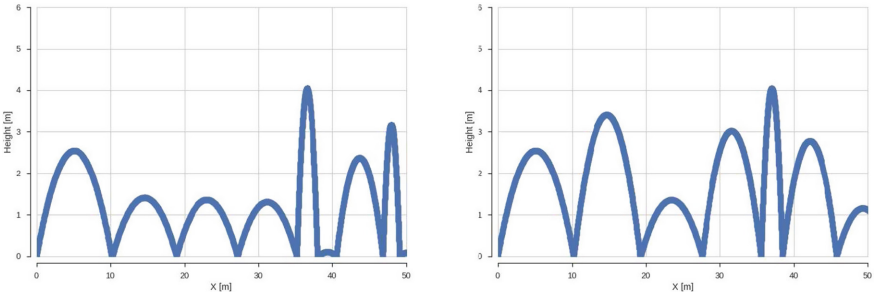

Fig. 2. Two random trajectories of the bouncing ball

Given the initial velocity vector and position, we can therefore calculate when the ball hits the ground ($y(t) = 0$) the next time. When this happens, we reverse sign of the velocity vector's y-coordinate, damp the speed with a factor ρ and add a random perturbation to the resulting angle to model an irregular surface. More precisely, the velocity vector at any point in time is given by

$$\boldsymbol{v}(t) = \frac{\partial(x(t), y(t))}{\partial t} = (v\cos(\theta), v\sin(\theta) - gt)\,. \tag{4}$$

In particular, we are interested in the next time t_{n+1} of hitting the surface. This time is given by setting the y-coordinate to zero:

$$t_{n+1} = \sqrt{\left(\frac{\sin(\theta)v}{g}\right)^2 + \frac{2y(t_n)}{g}} + \frac{v\sin\theta}{g} \tag{5}$$

To bounce, dampen, and perturb the angle, we simply set the speed and angle at the next time t_{n+1} as follows:

$$||v(t_{n+1})||_2 = ||\boldsymbol{v}(t_n)||_2\rho \qquad \theta(t_{n+1}) = \eta \tag{6}$$

Here, η is a random perturbance. For simplicity, we choose a random (uniform) perturbation from a pre-specified list: $\eta \sim \mathcal{U}\{\eta_1, \ldots, \eta_m\}$.

Next we define the rare event : we are interested in the probability that the ball will hit a small range on the surface (e.g. a hole), described by height 0 and a small interval for the x-coordinate. When the interval is sufficiently small, the probability of reaching this target is equal to the probability of drawing an exactly defined sequence of angles for each bounce on the surface.

4 Simulation Guiding

As mentioned in the previous section, our guiding method explained subsequently differs from variance reduction techniques in that it tries to guide towards rare events even if such have not yet been encountered, while the latter adapt once a non-zero initial statistics has been obtained. In order to explain the difference, we expose the underlying algorithms in the following.

The TUTS Algorithm: In order to obtain a quantitative estimate of the probability that a severe event happens in the driving example, we employed the TUTS algorithm from [13]. TUTS requires a continuous function indicative of (in the sense of roughly correlated to) the criticality $\tilde{c}$ of a simulation run. Such functions can either be designed by the user, or sensible criticality functions from traffic psychology [15] may be used, or quantitative interpretations of temporal logic specifications [2,6] can be employed.

A user-defined criticality threshold τ defines the separation between acceptable and unacceptable situations. The TUTS algorithm attempts to guide the simulation into a region close to the threshold τ, where the variance of satisfaction of the binary verification goal is high if the threshold τ used coincides with the borderline between satisfaction and violation. It therefore employs a tree representing all simulation runs observed so far, and it tries to assign adequately modified probabilities to the decisions in that tree. Note that the use of such a tree allows to assign different probabilities to the same decision at different time instants: a move from state v_1 to state v_2 may happen multiple times along a run, yet may be assigned different probabilities by the guiding algorithm at different times. This property is extremely relevant for the setting of cognitive architectures as, to take our example, the decision whether to address the side task is drawn some thousand times within a single test drive, yet only a handful of those decision points has measurable influence on the risk — some raising risk (distractions in the unknown critical distance before the bridge), others lowering risk (slightly earlier distractions, which reduce the probability of again engaging into the side task during the critical moments).

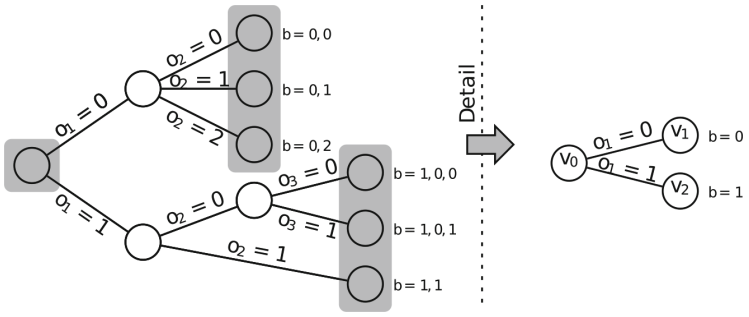

Fig. 3. Event tree spanned by the options o_*. b records the history of the probabilistic choices o_* along the path.

In the course of the simulation, let $C(v)$ be the set of criticality values that have been observed in all simulation runs that passed the node v of the above tree. As an example consider the event tree on the right side of Fig. 3. Being in node v_0, the simulation guide's aim is to give preference, in the sense of boosting its likelihood, to an action (in the example $o_1 = 0$ or $o_1 = 1$ are the possible actions) that more likely results in a criticality close to τ. We measure

the closeness to τ using a studentization of the observed deviations according to the distribution of the observations $C(v)$, i.e., calculate the z-score of the distance to τ as follows:

$$z(v) = \frac{\tau - \mu(C(v))}{\sigma(C(v))}$$

To increase the likelihood of observing a criticality of τ when passing v, the guide should prefer options which lead to small absolute z-scores. This is done in a probabilistic way by appropriately putting weight on all options that the guide can select. A weight $w(v)$ is defined for each node v already existing in the tree.[2] The function $t : V \times O \longrightarrow V$ defines the parent-child relationship in the tree: $t(v, o)$ gives the node that is reached when action o is selected while in node v.

The guiding algorithms uses the weights of the nodes to modify the probabilistic selection of actions from the current set of actions $\tilde{O}$. In detail, the probability of selecting action $o \in \tilde{O}$ if the current node is v is defined as:

$$\mathrm{P}_v(o) = \frac{w(t(v, o))}{\sum_{p \in \tilde{O}} w(t(v, p))}. \tag{7}$$

This means that options that lead to highly weighted child nodes are selected with higher probability. Therefore the nodes with low z-values should have high weights. The weights are defined by:

$$w(v) = \frac{1}{(|z| + 1)^{f(v_p)}} \tag{8}$$

Unless v has been visited twice, $\sigma(C(v))$ does not exist and z is undefined. Therefore, if any selectable child node has not yet been visited twice, the guide selects one of these randomly with their original probability assigned by the unmodified probabilistic model. In this way the guide explores each branch at least two times before deciding about its relative boost factor in the further exploration. The $f(v_p)$ exponent is used to adjust the weights the more confidence is gained about the distribution of criticality values in $C(v)$. Hereby v_p is the parent node of v such that each sibling uses the same exponent.

Especially for nodes at the top of the event tree the variance of criticality values $\sigma(C(v))$ is high and sibling nodes often have similar mean values $\mu(C(v))$. These nodes are at the beginning of the simulations. Many subsequent decisions influence the criticality of a simulation. This results in high variances wide confidence intervals for early nodes. In order to take the confidence about the z-values into account the function f is used. This function should rise with the empirical precision of the z-values and lead to a spreading of weights, the more confident

[2] Note that hitherto unseen paths in the tree can arise during simulation due to the probabilistic nature of the model being simulated. Therefore, the set of nodes in the tree grows incrementally.

we are in the z-values. For our use case scenario we used a simple definition with free parameters a and b used to adjust the search speed:

$$f(v_p) = a + b \cdot n_{min}, \quad (a, b = 0.5) \tag{9}$$

where n_{min} represents the minimal number of visits of any $v|v_p$ is parent.

Cross-Entropy Method: Cross-entropy [14] is a method which uses adaptive importance sampling (AIS) to adjust the current proposal distribution (denoted by its density, or probability mass function q) such that it converges to the optimal proposal distribution. Here, optimal means that a single sample is sufficient to estimate the expectation of interest exactly. The resulting estimator therefore has zero variance. As this optimal proposal, however, is not available, the cross-entropy method estimates this optimal proposal based on the samples already drawn. To evaluate the proximity of the current proposal to this estimation, the Kullback-Leibler divergence is used. As the Kullback-Leibler divergence is also called cross-entropy, the method is called AIS using cross-entropy.

Instead of reviewing the cross-entropy method in general, we illustrate its application to the bouncing ball. Let p_i denote the probability under the bouncing ball model to draw the i-th possible angle η_i. Under the stochastic bouncing ball model, angles are independent across time-points. Hence the probability of a trace of multiple angles $x_t, t = 1 \ldots, T$ is simply given by the product over $\prod_t p_i \delta(x_t, \eta_i)$. Here δ denotes the Kronecker-delta, which evaluates to 1 if $x_t = \eta_i$ and 0 else. Similarly, we chose q to represent the probability of drawing different angles. As the occurrence of the rare event effectively couples the random events across time, it might be beneficial to allow for inter-time-dependency within the proposal distribution. However, as this increases the number of parameters exponentially, we use the same independence assumption also for the proposal distribution. Specifically, in order to analytically compute the cross-entropy update, we use the following parameterization of q:

$$q(x) = \frac{\exp(\gamma_i \delta(x, \eta_i))}{\sum_k \exp(\gamma_k)} \tag{10}$$

The probability of generating a particular angle $x \in \{\eta_1, \ldots \eta_m\}$ can thus be adjusted by choosing different values of γ. γ can be interpreted as the natural parameter of the exponential family with $\delta(x, \eta_i)$ as the sufficient statistics, which enables us to easily compute updates of the parameters γ, see below.

In the first step, N_0 simulation runs are drawn using the current proposal q^n. Here N_0 is a free parameter of the algorithm, to which we refer to as the batch-size. Each of these simulation runs have an associated criticality value c_i. For the bouncing ball example, we used the Euclidean distance between the vector of sampled angles to the (known) vector of angles that would lead to the bouncing ball hitting the small area associated with the rare event. Using this criticality, the cross-entropy method now selects the α-most critical simulation runs, i.e., the index set

$$I_\alpha := \{i : |\{j : c_j < c_i\}| \leq \alpha N_0\} \tag{11}$$

This index set in turn is used to estimate the optimal proposal distribution q^*, i.e., the proposal that would lead to a zero-variance estimator[3]. Due to the exponential family form of our representation, we only need to compute the empirical means of the sufficient statistics to compute new parameters γ to obtain an updated proposal. This in turn is due to the fact that moment matching is equivalent to minimizing the Kullback-Leibler divergence between the empirical zero-variance distribution and the proposal across different parameter settings of the proposal, see [14].

Hence, the new parameters γ^{n+1} of the proposal distribution can be set by calculating the empirical averages of the sufficient statistics, where we have to account for the re-weighting according to the current proposal distribution q^n.

$$\gamma_i^{n+1} = \frac{1}{|I_\alpha|} \sum\nolimits_{k \in I_\alpha} \frac{1}{T_k} \sum\nolimits_{t}^{T_k} \frac{p(x_t^k)}{q^n(x_t^k)} \delta(x_t^k, \eta_i) \tag{12}$$

Here, x_t^k denote the (random) choices at time k within the k-th trace of the generated batch. Note that we can use the inner sum $\sum_t$ as we assume independence and therefore treat each draw along the k-th trace equally. Having new parameters and thus a new proposal distribution q^{n+1}, we can use this new distributions to generate a new batch of samples of size N_0.

Using this update, the parameters capture the frequencies with which different choices η occurred within the α most critical traces of the generated batch. This information in turn is used to generate those choices more frequently within the next batch. However, even if the certain frequencies have not been observed, due to the exponential structure, the corresponding probability would never be set to zero. Therefore, using this parametrization, we cannot converge to an optimal distribution completely ignoring certain choices unless we use arbitrary large batch-sizes.

5 Confidence Intervals

In order to compare results of different simulation guiding techniques, we need to be able to calculate their confidence intervals (CI), as explained in this section.

Binomial Confidence Interval. For computing the confidence interval of the naïve estimator, we employ the simple binomial CI, also known as the Clopper-Pearson confidence interval, or exact confidence interval [1].

Bootstrap Confidence Intervals. When applying importance sampling, the originally binomial distribution is modified to a multinomial one, as samples are no longer evaluated with just 0 (safe run) or 1 (safe or bad run), but with a plethora of different importance weights. In order to calculate an approximation of the corresponding CI, we compute bootstrap confidence intervals [3]. In order to compute bootstrap confidence intervals for an estimator on samples

[3] Note that, due to the finite amount of samples used, this is only an approximation.

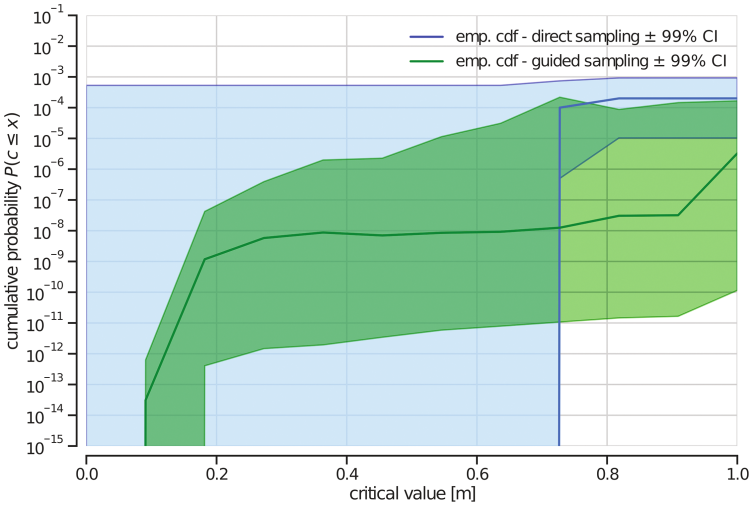

Fig. 4. Estimated probabilities and 99% confidence intervals for approaching the bridge pillar closer than a specified critical distance. Blue: naïve SMC, green: TUTS guiding. (Color figure online)

$S_0 = (x_1, \ldots, x_n)$, one creates multiple samples of the same size by drawing from $\{x_1, \ldots, x_n\}$ with replacement. On these re-sampled samples one then computes the variability of the estimator evaluated on each of the re-samples. Specifically, let $S_1, \ldots S_m$ new samples be obtained via re-sampling and let f be the estimator which takes as argument a sample and provides the estimate as an output. To obtain a $1-\alpha$ confidence interval, one then orders the estimator outputs $f(S_1), \ldots, f(S_m)$ from lowest to highest. The confidence interval in turn is then given by $[f(S_l), f(S_u)]$, where l and u are the indices corresponding to the $(\alpha/2)m$ and $(1-\alpha/2)m$ entries in the ordered list respectively.

6 Results

In the driving example, a critical event X_i occurs whenever the distance between car and bridge pillar falls below a specified distance. The likelihood $\hat{p}$ can be computed using the unbiased importance sampling estimator

$$\hat{p} \approx \frac{1}{N} \sum_{i=1}^{N} \left(\begin{cases} 1, \text{ if } x_i \text{ is critical} \\ 0, \text{ else} \end{cases} \right) \frac{p(x_i)}{q(x_i)}, \quad x_i \sim q_i.$$

N represents the number of simulation runs, $p(x_i)$ the original probability and $q(x_i)$ the weighted probability of x_i when simulated under measure q.

To show the likelihood of different criticalities, we plot in Fig. 4 the estimated probabilities and 99% confidence intervals for approaching the bridge pillar closer than a specified critical distance. The results have been computed independently

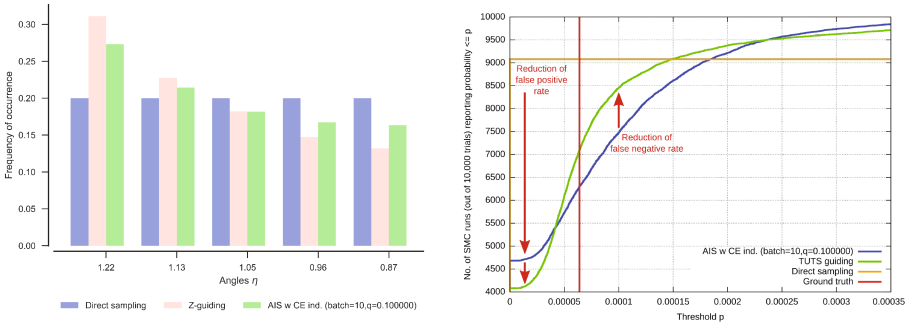

Fig. 5. Randomly bouncing ball with 1,000 samples per batch and 10,000 repetitions. Left: Histogram of the frequency of selecting different bounce angles η. Right: Frequency of reporting a hit probability of at most p, counted over 10,000 independent SMC runs of 1,500 samples each. (Color figure online)

using naïve Monte Carlo SMC (blue graph) and TUTS guiding (green graph). The latter obviously presents two significant enhancements:

1. Significantly better assessment of critical distances below 0.7 m, which represent the rare events in the scenario. The smallest distance was recorded at approx. 0.1 m with an estimated probability of 5×10^{-13} and the corresponding 99% bootstrap confidence interval spanning $[10^{-15}, 10^{-11}]$. In the same regime, naïve sampling can only provide 99% confidence that the probability is below 7×10^{-4}.
2. Tighter upper bounds on the likelihood of reaching critical distances above 0.7 m. A quantitative safety specification like "the likelihood of getting closer than 1 m to the bridge pillar should be less than 10^{-4}" can thus be verified with 99% confidence by TUTS, while naïve sampling remains inconclusive. Naïve sampling well reports tighter lower bounds than TUTS, but these are of no use in quantitative verification: both acceptance and refutation of quantitative safety targets depend on whether the threshold is exceeded by the upper bound of the confidence interval.

This comparison demonstrates that deliberately asymmetric CIs can be beneficial within statistical verification: preferring sharp upper bounds of CIs over narrowing the CIs would be a sensible optimization goal for sampling strategies in statistical model-checking.

We can demonstrate this effect also by a threefold comparison on the bouncing ball example, where we compare naïve sampling, adaptive importance sampling driven by cross-entropy, and TUTS guiding. The left part of Fig. 5 demonstrates that TUTS actually employs a significantly different importance sampling strategy than the cross-entropy method. This strategy leads to higher hit rates, as witnessed by Table 1. This higher hit rate, at correspondingly lower importance weight assigned to each hit, generates a steeper increase of the distribution of test outcomes around the true probability, as depicted in the right part of

Table 1. Hits to target achieved over 10,000 batch runs of 1,000 samples each.

Algorithm	Batches featuring ≥ 1 hit	Total number of hits
Naïve Monte Carlo sampling	648	671
AIS driven by cross-entropy	3909	5452
TUTS guiding	4840	25867

Fig. 5. This graph shows for each algorithm the frequency (counted over 10,000 independent runs of 1,500 samples each) of reporting a hit probability below the threshold given on the horizontal axis. Due to quantization, naïve sampling is inapt of computing any positive probability less than $\frac{1}{1500}$. As the actual hit probability is considerably smaller than $\frac{1}{1500}$ at 0.000064 (marked by the red perpendicular line), naïve sampling is likely to report a massive underestimate of 0; this happens on approx. 91.2% of the runs. Being based on importance sampling, both TUTS and AIS can yield probability estimates close to the actual probability and are thus more informative. For the small sample size underlying the graph ($\leq 2,000$ per batch) TUTS is significantly less likely to generate considerable underapproximations below 0.000042, thus reducing the false-positive rate when employing acceptance thresholds in that range. TUTS also has by a fair margin the highest probability of generating relatively exact approximations: Some 3,340 estimates provided by TUTS fall into the range of $\pm 25\%$ around the true probability, while AIS features only 2,065 within that range (and naïve sampling none). As might be expected, the likelihood of massive underestimation by AIS decreases when AIS is given significantly more time for adaptation by increasing batch sizes. For the bouncing ball example we found this to happen when batch sizes considerably exceed 2,000. In that regime, AIS starts to outperform TUTS concerning the number of massive underestimates generated, though TUTS continues to yield the steepest curve around the true probability. Given that the rare events in our actual application domain of ADAS are multiple orders of magnitude more rare than for the bouncing ball, it is, however, unclear whether the corresponding batch sizes guaranteeing convergence of AIS would be a practical option. The faster initial convergence of TUTS seems an interesting property to explore.

7 Conclusion

Within this article, we have extended the TUTS guiding algorithm for identifying extremely rare events in statistical model checking [13] by rigorous confidence bounds. We argue that within quantitative verification contexts, not the actual width of the confidence bounds is relevant, but tightening the single bound relevant to the verification problem. In verification contexts this is the upper bound on violating the requirement or, equivalently, the lower bound on satisfaction. This implies that classical means of variance reduction in importance sampling

frameworks tend to aim at the wrong goal, namely achieving a precise probability estimate by reduction of the width of the confidence interval, which is correlated with, yet not identical to the goal of tightening the single bound of the confidence interval that is of relevance to verification. A complex traffic benchmark from the development of advanced driver assistance systems provides witness of this effect: here TUTS guiding provides much sharper upper bounds on accident probabilities throughout the whole regime even though providing relatively wider confidence intervals than naïve sampling in parts of the regime (cf. Fig. 4). As simulation guiding by the optimal adaptive importance sampling method, namely the cross-entropy approach, could not be realized on this complex example, we addressed a second, artificial example of a bouncing ball, where we compared naïve sampling, adaptive importance sampling guided by the cross-entropy method, and TUTS guiding. The results confirm that the TUTS algorithm provides a sampling scheme that converges rapidly even for batch sizes that are small relative to the actual probability. For such small batches, it outperforms both naïve sampling and the cross-entropy method. Beneficial combinations with the latter, where TUTS would foster fast early convergence and the cross-entropy method could then take over, remain an issue of further research.

References

1. Clopper, C.J., Pearson, E.S.: The use of confidence or fiducial limits illustrated in the case of the binomial. Biometrika **26**(4), 404–413 (1934)
2. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: Chatterjee, K., Henzinger, T.A. (eds.) FORMATS 2010. LNCS, vol. 6246, pp. 92–106. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15297-9_9
3. Efron, B., Tibshirani, R.J.: An Introduction to the Bootstrap. Monographs on Statistics and Applied Probability, no. 57. Chapman & Hall/CRC, London (1993)
4. European Commission: Towards a European road safety area: policy orientations on road safety 2011–2020 (2010). http://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52010DC0389
5. Eurostat: Slightly over 26 000 victims of road accidents in the EU in 2015. Eurostat Press Office Vincent (2016). http://ec.europa.eu/eurostat/documents/2995521/7734698/7-18112016-BP-EN.pdf
6. Fränzle, M., Hansen, M.R.: A robust interpretation of duration calculus. In: Van Hung, D., Wirsing, M. (eds.) ICTAC 2005. LNCS, vol. 3722, pp. 257–271. Springer, Heidelberg (2005). https://doi.org/10.1007/11560647_17
7. Gietelink, O., De Schutter, B., Verhaegen, M.: Adaptive importance sampling for probabilistic validation of advanced driver assistance systems. In: 2006 American Control Conference, vol. 19, 6 pp. (2006)
8. Gietelink, O., De Schutter, B., Verhaegen, M.: Probabilistic validation of advanced driver assistance systems. In: Proceedings of the 16th IFAC World Congress, vol. 19 (2005)
9. Jegourel, C., Larsen, K.G., Legay, A., Mikučionis, M., Poulsen, D.B., Sedwards, S.: Importance sampling for stochastic timed automata. In: Fränzle, M., Kapur, D., Zhan, N. (eds.) SETTA 2016. LNCS, vol. 9984, pp. 163–178. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47677-3_11

10. Kahn, H.: Use of different Monte Carlo sampling techniques, p. 766 (1955)
11. Page, Y., et al.: A comprehensive and harmonized method for assessing the effectiveness of advanced driver assistance systems by virtual simulation: the P.E.A.R.S. initiative. In: The 24th International Technical Conference on the Enhanced Safety of Vehicles (ESV). NHTSA, Gothenburg (2015)
12. Puch, S., Wortelen, B., Fränzle, M., Peikenkamp, T.: Using guided simulation to improve a model-based design process of complex human machine systems. In: Modelling and Simulation, ESM 2012, pp. 159–164. EUROSIS-ETI, Essen (2012)
13. Puch, S., Wortelen, B., Fränzle, M., Peikenkamp, T.: Evaluation of drivers interaction with assistant systems using criticality driven guided simulation. In: Duffy, V.G. (ed.) DHM 2013. LNCS, vol. 8025, pp. 108–117. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39173-6_13
14. Rubinstein, R.: The cross-entropy method for combinatorial and continuous optimization. Methodol. Comput. Appl. Probab. **1**, 127–190 (1999)
15. Vogel, K.: A comparison of headway and time to collision as safety indicators. Accid. Anal. Prev. **35**(3), 427–433 (2003)
16. Vorndran, I.: Unfallstatistik - Verkehrsmittel im Risikovergleich. DESTATIS (2010). https://www.destatis.de/DE/Publikationen/WirtschaftStatistik/Monatsausgaben/WistaDezember10.pdf?__blob=publicationFile
17. WIVW GmbH: Fahrsimulationssoftware SILAB. https://wivw.de/de/silab
18. Wortelen, B., Baumann, M., Lüdtke, A.: Dynamic simulation and prediction of drivers' attention distribution. Transp. Res. Part F Traffic Psychol. Behav. **21**, 278–294 (2013)
19. Wortelen, B., Lüdtke, A., Baumann, M.: Integrated simulation of attention distribution and driving behavior. In: Proceedings of the 22nd Annual Conference on Behavior Representation in Modeling & Simulation, pp. 69–76. BRIMS Society, Ottawa (2013)
20. Zuliani, P., Baier, C., Clarke, E.M.: Rare-event verification for stochastic hybrid systems. In: Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control, pp. 217–226. ACM, New York (2012)