Multi-objective Parameter Synthesis in Probabilistic Hybrid Systems

Martin Fränzle[1,2], Sebastian Gerwinn[2(✉)], Paul Kröger[1], Alessandro Abate[3], and Joost-Pieter Katoen[4]

[1] Department of Computing Science, Carl von Ossietzky University, Oldenburg, Germany
{fraenzle,paul.kroeger}@informatik.uni-oldenburg.de
[2] OFFIS Institute for Information Technology, Oldenburg, Germany
sebastian.gerwinn@offis.de
[3] Department of Computer Science, Oxford University, Oxford, UK
aabate@cs.ox.ac.uk
[4] Department of Computer Science, RWTH Aachen University, Aachen, Germany
katoen@cs.rwth-aachen.de

Abstract. Technical systems interacting with the real world can be elegantly modelled using probabilistic hybrid automata (PHA). Parametric probabilistic hybrid automata are dynamical systems featuring hybrid discrete-continuous dynamics and parametric probabilistic branching, thereby generalizing PHA by capturing a family of PHA within a single model. Such system models have a broad range of applications, from control systems over network protocols to biological components. We present a novel method to synthesize parameter instances (if such exist) of PHA satisfying a multi-objective bounded horizon specification over expected rewards. Our approach combines three techniques: statistical model checking of model instantiations, a symbolic version of importance sampling to handle the parametric dependence, and SAT-modulo-theory solving for finding feasible parameter instances in a multi-objective setting. The method provides statistical guarantees on the synthesized parameter instances. To illustrate the practical feasibility of the approach, we present experiments showing the potential benefit of the scheme compared to a naive parameter exploration approach.

1 Introduction

Systems engineering frequently calls for finding parameters on event probabilities, like frequencies of inspections or halts for maintenance, under constraints on expected values of costs and rewards, like expected maintenance cost and expected loss due to unscheduled downtime. In this article, we propose a method which can systematically address this problem for probabilistic hybrid automata

This work is funded by the EU FP7 projects MoVeS, SENSATION and AMBI, by the DFG through the collaborative research center SFB-TR 14 AVACS, and by the John Fell OUP Research Fund.

S. Sankaranarayanan and E. Vicario (Eds.): FORMATS 2015, LNCS 9268, pp. 93–107, 2015.
DOI: 10.1007/978-3-319-22975-1_7

featuring parametric discrete probability distributions governing choices within the automaton's control flow. We are able to devise instances of the parametric distribution guaranteeing multi-objective specifications concerning expected costs and rewards over a bounded horizon, i.e., enforcing that expectations on a multi-dimensional vector of costs/rewards incurred within the bounded horizon satisfy a first-order specification over these costs and rewards. Such specifications may place bounds on individual costs/rewards as well as relate them arithmetically, e.g., enforcing a relation between maintenance costs and system availability. The bounded horizon may deal with a number of computation steps or with a time bound, provided the system is non-Zeno.

For confined settings, such parameter fitting could be reduced to SAT modulo theory (SMT) solving, based on a parametric extension of the encodings pioneered by Wimmer *et al.* [23]. This would, however, require that both the system dynamics and the reward functions, as well as their dependency on probabilistic choices, can be encoded in the arithmetic theory supported by the SMT solver, and that the bounded horizon is given in terms of a step bound in order to facilitate a symbolic unravelling of the transition tree. Such an approach would for example require SMT over polynomials to deal with parametric probabilistic linear hybrid automata (featuring piecewise constant differential equations, linear guards, etc.). It is thus confined to systems with rather simple dynamics and, given the complexity of polynomial constraint solving, of rather small size under rather restrictive bounds on the temporal horizon.

To overcome these shortcomings, our method is based on ideas from *statistical model checking* (SMC) [24], which in its traditional setup deals with non-parametric probabilistic (hybrid) systems. The strength of SMC is that it can tackle arbitrary system dynamics, as long as a simulator is available, and is rather insensitive to system size. The underlying principle is to run a number of simulations of the system under investigation within a simulator faithfully representing the — then necessarily non-parametric — probabilistic choices in the system as well as its state dynamics, and to exploit the set of traces obtained from the simulations for computing an estimate of the expected values of reward or cost variables by means of averaging over the individual traces.

Extensions of SMC to parametric probabilistic hybrid systems could in principle be addressed by sampling the parameter space, yet this would induce the curse of dimensionality, confining such a method to fitting isolated parameters. We avoid this problem by adequately adapting the concept of *importance sampling*, which permits factoring out the parameter dependency of the distributions by sampling a fixed substitute distribution instead. The method is based on a combination of statistical model checking of a substitute model devoid of parametricity, a symbolic version of importance sampling providing an *SMT representation of the parameter dependencies*, and SMT solving for finding feasible parameter instances satisfying the constraints imposed on expected values.

Organization of the Paper. Section 2 introduces parametric probabilistic hybrid automata and multi-objective specifications on expected rewards. Section 3 explains importance sampling for parametric distributions and presents the

equations that are key to our approach. Section 4 considers the specific case of parametric distributions in (finite- or infinite-state) Markov chains and provides statistical guarantees in the form of confidence intervals. Section 5, finally, sketches how SMT solving can be applied so as to find (a) feasible parameter instances satisfying the multi-objective specification and (b) the confidence in the provided solution. We close with discussing related work and general conclusions.

2 Parametric Probabilistic Hybrid Automata

Probabilistic hybrid automata (PHA). [19] extend hybrid automata with discrete probabilistic branching. This enables modeling of, e.g., random component failures and data packet losses. Similar to hybrid automata, PHA feature a finite set of discrete locations (or modes), each of which comes decorated with a differential equation governing the dynamics of a vector of continuous variables while residing in that mode. Modes change through instantaneous transitions guarded by conditions on the current values of the continuous variables, and may yield discontinuous updates of the continuous variables. Aiming at simulation-based evaluation methods as in SMC, transition selection here is assumed to be deterministic, i.e., guard conditions at each mode are mutually exclusive. To prevent nondeterminism between possible time flows and transitions, we also assume that transitions are urgent, i.e., they are taken as soon as they are enabled (which furthermore renders mode invariants redundant). In addition to these mechanisms from deterministic hybrid automata, PHA allow for the probabilistic selection of a transition variant based on a discrete random experiment. Following the idea of Sproston [19,20], the selected transition entails a randomized choice between transition variants according to a discrete probability distribution. The different transition variants can lead to different follow-up locations and different continuous successors, as depicted in Figure 1, where the guard condition determining transition selection is depicted along the straight arrows leading to a potential branching annotated with probability terms denoting the random experiment.

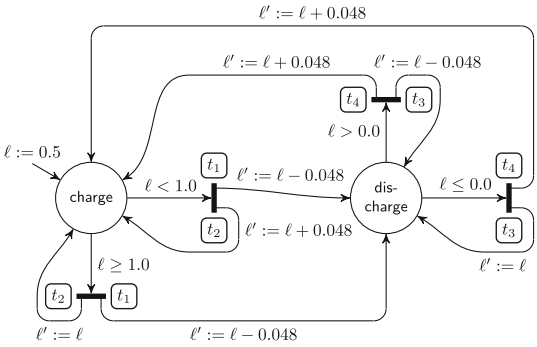


Fig. 1. PPHA model of a charging station. Modes are labeled with labels `charge` and `discharge` abbreviating ODE (not shown explicitly) representing corresponding dynamics over a continuous capacity ℓ. Modes can switch according to guarded transitions leading to a probabilistic branch. Probabilities are summarized as terms $t_1, \ldots, t_4$ indicating their parameter dependencies.

Parametric probabilistic hybrid automata (PPHA) extend PHA with the presence of parameters. Whereas in PHA the probability distributions are constants, PPHA allow the branching probabilities to be terms over a set *Param* of parameter names. The viable parameter instantiations $\theta : \mathit{Param} \to \mathbb{R}$ are constrained by an arithmetic first-order predicate ϕ over *Param*, defining their mutual relation. Let $\Theta = \{\theta : \mathit{Param} \to \mathbb{R} \mid \theta \models \phi\}$ denote the set of all viable parameterizations. Arithmetic terms over *Param* are subject to the constraint that for all viable parameter valuations $\theta \models \phi$, the sum of outgoing probabilities assigned to each transition is one, i.e., $\phi \implies \sum_{i=1}^{n} t_i(\theta) = 1$ holds for the probability terms $t_1, \ldots, t_n$ associated to each transition t. Note that the probability terms need not contain free variables θ: ordinary non-parametric distributions are thus special cases of parametric distributions and do not require special treatment.

2.1 Interpretation as Parametric Infinite-State Markov Chain

A PPHA engages in a sequence of continuous flows and discrete jumps. The continuous flows are solutions of the ordinary differential equations assigned to the current location. The discrete jumps originate from taking enabled transitions, thereby eliciting a transition as soon as it is triggered, and then probabilistically deciding among the different transition variants, with their associated target locations and resets to continuous variables. For the sake of formal analysis, we formalize the semantics of PPHA through a reduction to a parametric infinite-state Markov chain. For a PPHA with location set Λ and continuous variables $x_1, \ldots, x_D$, the states of the Markov chain are given by $\Sigma = \Lambda \times \mathbb{R}^D$ and the initial state distribution is inherited from the PPHA. Each state $\sigma = (l, \boldsymbol{x}) \in \Sigma$ gives rise to a parameter-dependent distribution[1] $p_\sigma : \Sigma \times \Theta \to [0, 1]$ of successor states:

$$p_\sigma(\sigma', \theta) = \begin{cases} t(\theta) & \text{if a transition } (\sigma, \sigma') \text{ labeled with probability term } t \text{ is enabled,} \\ 1 & \text{if } \sigma' = (l, g(t)), \text{ where } g \text{ is a solution to the ODE associated to} \\ & l \in \Lambda \text{ with } g(0) = \boldsymbol{x}, \text{ no transition is enabled in } (l, g(t')) \text{ for any} \\ & t' \in [0, t[, \text{ and a transition is enabled in } \sigma' = (l, g(t)), \\ 0 & \text{otherwise.} \end{cases}$$

Given a parametric infinite-state Markov chain M with its initial (state) distribution given by a density $\iota : \Sigma \to \mathbb{R}_{\geq 0}$ and a parametric next-state distribution $p_\sigma : \Sigma \times \Theta \to [0, 1]$, the *density function associated to finite runs* $\langle \sigma_0, \sigma_1, \ldots, \sigma_k \rangle \in \Sigma^*$ *given a parameter instance* $\theta \in \Theta$ is

$$p_M(\langle \sigma_0, \sigma_1, \ldots, \sigma_k \rangle; \theta) = \iota(\sigma_0) \cdot \prod_{i=0}^{k-1} p_{\sigma_i}(\sigma_{i+1}, \theta).$$

Note that while we represented the parametric dependence by a single parameter θ, this can be vector valued, thereby encoding potentially different dependencies for different nodes.

[1] Note that due to the finite probabilistic branching in PPHA, we deal with distributions rather than densities here.

2.2 Parameter Synthesis

Let $f : \Sigma \to \mathbb{R}$ be a scalar function on states, to be evaluated on the last state of a run and called the *reward* f of the run,[2] and let $k \in \mathbb{N}$. The *k-bounded expected reward for f in a parameter instance $\theta \in \Theta$* is

$$\mathcal{E}_{M,k}[f;\theta] = \int_{\Sigma^k} f(\sigma_{k-1}) p_M(\langle \sigma_0, \sigma_1, \ldots, \sigma_{k-1} \rangle; \theta) \, \mathrm{d}\langle \sigma_0, \sigma_1, \ldots, \sigma_{k-1} \rangle,$$

where Σ^k denotes the sequences over Σ of length k. We will subsequently drop the index M in $\mathcal{E}_{M,k}$ and p_M whenever it is clear from the context.

Rewards represent quantitative measures of the system's performance, and therefore mutual constraints on their values can be used for capturing design goals. The design problem we are thus facing is, given a vector $f_1, \ldots, f_n : \Sigma \to \mathbb{R}$ of rewards in Markov chain M, to ensure via adequate instantiation of the parameter that the expected rewards meet the design goal. The following definition captures this intuition.

Definition 1 (Parameter Synthesis Problem). *Let $f_1, \ldots, f_n : \Sigma^k \to \mathbb{R}$ be a vector of rewards in a Markov chain M and let C be a* design goal *in the form of a constraint on the expected rewards, i.e., an arithmetic predicate containing $f_1, \ldots, f_n$ as free variables. A parameter instance $\theta : Param \to \mathbb{R}$ is* feasible (wrt. M and C) *iff*

$$\theta \models \phi \quad \textit{and} \quad [f_1 \mapsto \mathcal{E}_{M,k}(f_1;\theta), \ldots, f_n \mapsto \mathcal{E}_{M,k}(f_n;\theta)] \models C.$$

The multi-objective parameter synthesis problem *is to find a feasible parameter instance θ, if it exists, or to prove its absence otherwise.*

Stated in words, a parameter instance θ is feasible wrt. ϕ and C iff the parameters are in the range defined by ϕ and the expected rewards resulting from the instantiation meet the multi-objective C. Note that the aim is to find a parameter instance meeting our design goal; we are not considering determining all instantiations. In the sequel, we will focus on a single reward f rather than n such functions and indicate whenever appropriate how to deal with a vector.

3 Estimating Expectations by Sampling

In order to introduce the general concept of importance sampling [21], we mostly abstract from our PPHA setting in this section. We instead assume that the parametric probability distribution of the random variable $x \in X$ is given in terms of a density function $p(\cdot;\theta)$ which depends on a vector θ of bounded real-valued parameters. Permissible values of θ are defined by a first-order constraint ϕ.

[2] Despite the generality of the PPHA model, defining rewards exclusively on the final state σ_k of a run $\langle \sigma_0, \sigma_1, \ldots, \sigma_k \rangle \in \Sigma^*$ is as expressive as defining them via functions $f(\langle \sigma_0, \sigma_1, \ldots, \sigma_k \rangle)$, where $f : \Sigma^{k+1} \to \mathbb{R}$. Such rewards can be alternatively encoded by augmenting the state-space of the PPHA model with additional variables accumulating the quantities of interest along the trajectory.

Classical Sampling. Given an arbitrary (bounded) function $f : X \to \mathbb{R}$, we are interested in estimating expected values of f under all parameter values $\theta \models \phi$. The expectation $\mathcal{E}[f;\theta]$ for reward f given parameter vector θ is

$$\mathcal{E}[f;\theta] = \int_X f(x)p(x;\theta)\,\mathrm{d}x \ . \tag{1}$$

Given a specific parameter instance θ^* and a process sampling x_i according to the distribution $p(\cdot;\theta^*)$, the expectation $\mathcal{E}[f;\theta^*]$ can be estimated by

$$\tilde{\mathcal{E}}[f;\theta^*] = \frac{1}{N}\sum_{i=1}^{N} f(x_i) \ , \tag{2}$$

which is the empirical mean of the sampled f-values. In our PPHA setting, a reasonable process for generating such samples x_i according to the distribution $p(\cdot;\theta^*)$ would be a simulator for non-parametric PHA, applied to the instance of the PPHA under investigation obtained by substituting θ^* for the free parameters.

For sufficiently large N, we expect $\mathcal{E}[f;\theta^*] \approx \tilde{\mathcal{E}}[f;\theta^*]$ due to the law of large numbers. We can quantify the quality of the approximation in (2) using Hoeffdings inequality [13], provided that f has a bounded support $[a_f, b_f]$:

$$P\left(\mathcal{E}[f;\theta^*] - \tilde{\mathcal{E}}[f;\theta^*] \geq \varepsilon\right) \leq \exp\left(-2\frac{\varepsilon^2 N}{(b_f - a_f)^2}\right) \geq P\left(\tilde{\mathcal{E}}[f;\theta^*] - \mathcal{E}[f;\theta^*] \geq \varepsilon\right) \tag{3}$$

Therefore, the empirical mean (2) yields a very reliable estimate of the actual expectation when the number of samples is large, with the accuracy given by (3).

Importance Sampling. While determining the empirical mean (2) by repeated simulation is an adequate procedure for assessing non-parametric PHA, it is bound to fail for PPHA when applied naïvely, as it would require to sufficiently densely cover the parameter space Θ with parameter instances θ_j^* and generating $j = 1, \ldots, N$ samples for each instance θ_j^*. This is thus subject to the curse of dimensionality. Fortunately, importance sampling [21] provides a means of using substitute probability distributions in sampling processes. We will exploit this for dealing with parameters. Importance sampling was originally designed to enhancing the quality of empirical estimates by artificially drawing according to their (assumed) importance for the estimate and later correcting the estimate by weighting the individual samples by that importance. In our setting, we will use importance sampling for estimating the parameter-dependent expectation $\mathcal{E}[f;\theta]$ defined in equation (1). Instead of sampling X according to the distribution p, importance sampling uses a different distribution q to sample from. It then calculates the empirical mean of the samples (over q), but weighs each sample x_i by its importance weight $\frac{p(x_i)}{q(x_i)}$, in order to obtain an estimate of the expectation under the original distribution p. Applying this idea to our parametric setting, we can pursue a single round of sampling wrt. some non-parametric distribution

q and estimate the expected value $\mathcal{E}[f;\theta]$ for arbitrary θ as follows:

$$\mathcal{E}[f;\theta] = \int_X f(x)p(x;\theta)\,\mathrm{d}x = \int_X \frac{f(x)p(x;\theta)}{q(x)}q(x)\,\mathrm{d}x$$
$$\approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)p(x_i;\theta)}{q(x_i)} =: \hat{\mathcal{E}}[f;\theta], \qquad \text{where } x_i \sim q. \tag{4}$$

Note that all the samples $\{x_1,\ldots,x_N\}$ are drawn according to the substitute[3] distribution q (indicated by $x_i \sim q$); nevertheless, (4) still keeps the parameter dependence $\hat{\mathcal{E}}[f;\theta]$ for arbitrary values of θ.

4 Symbolic Representation of Importance Sampling

The purpose of this section is to derive a symbolic characterization of a solution to our parameter synthesis problem. This is achieved by using samples drawn from the proposal distribution to construct a symbolic constraint system by means of the importance sampling expression for the expectation.

A Symbolic Constraint System. Let $p(x;\theta)$ have a closed-form representation given as term t. (Typically t contains one or more free occurrences of x and θ.) A symbolic representation of the parameter dependency of $\hat{\mathcal{E}}[f;\theta]$, and (due to the sampling error) an approximate symbolic representation of the parameter dependency of $\mathcal{E}[f;\theta]$ can now readily be obtained as follows. We replace all occurrences of $p(x_1;\theta)$ through $p(x_N;\theta)$ in (4) by the terms $t[x_1/x]$ through $t[x_N/x]$ respectively, and substitute the concrete values for N, $(x_i)_{i=1\ldots N}$, and $(f(x_i))_{i=1\ldots N}$. The resulting term, referred to as η, is a large sum with multiple occurrences of θ in different instances of the sub-term t. Let C be a constraint on the expected reward $\mathcal{E}$, i.e., C is a formula with free variable f formalizing the requirements on the expectation $\mathcal{E}[f;\theta]$. A parameter instance $\theta \models \phi$ statistically guaranteeing C can now in principle be found — or conversely, the infeasibility of C over ϕ be established — by solving the constraint system

$$(\mathcal{E}[f;\theta] = \eta[f;\theta]) \wedge \phi \wedge C \tag{5}$$

using an appropriate constraint solver. Note that (5) enforces $\theta \models \phi$ through the conjunct ϕ and guarantees $[f \mapsto \hat{\mathcal{E}}[f;\theta]] \models C$ due to the construction of η and the presence of the constraints $\mathcal{E} = \eta$ and C.

As $\hat{\mathcal{E}}[f;\theta] \approx \mathcal{E}[f;\theta]$, the instance θ of the parameterized system under investigation then intuitively is likely to also satisfy $\mathcal{E}[f;\theta] \models C$, as desired. However, the resulting parameter instances might suffer from being biased towards the particular samples, which will be investigated in detail in the next section.

The generalization of (5) to multiple rewards $f_j : X \to \mathbb{R}$ and a corresponding multi-objective constraint C containing arbitrary arithmetic and Boolean combinations of the expected rewards is straightforward, albeit potentially higher in computational cost.

[3] In principle, an arbitrary distribution q can serve as a substitute. In our setting, it is natural to use an instance $q = p(\cdot;\theta^*)$ of the parametric distribution, where $\theta^* \models \phi$.

Simplified Constraint System. In practice, the constraint (5) may become unwieldy due to the large number of samples x_i necessary for obtaining a sufficiently tight confidence bound in equation (3), as the number of samples directly translates into a corresponding number of summands in η. This problem can be alleviated in our setting as we consider Markov processes.

For the sake of illustration, let us assume that there is a (non-empty) subset Δ of the state set Σ of the Markov chain M, where the chain has a parameter-dependent probabilistic choice between just two transition alternatives, taking alternative one with probability t, where t is a term dependent on θ, and alternative two with probability $1-t$, and that all other states in $\Sigma \setminus \Delta$ feature non-parametric distributions.[4] During sampling, we substitute these parameter-dependent probabilities t and $1-t$ by the static substitute probabilities q and $1-q$, respectively, where $q \in]0,1[$ is a constant.

During a simulation providing N samples, we now keep track of how many times a run takes transition alternatives one and two. Let $T_{n,m}$ denote the set of simulated trajectories taking n times alternative one and m times alternative two. Note that there are finitely many $T_{n,m} \neq \emptyset$, and that in practice the number of non-empty $T_{n,m}$ is considerably smaller than N. Let $\Sigma_{n,m} = \sum_{x_i \in T_{n,m}} f(x_i)$ denote the sum of the rewards seen on all trajectories in $T_{n,m}$. This quantity can easily be computed during sampling. With these notations in place, we can partition the sum (4) in terms of the necessarily pairwise disjoint sets $T_{n,m}$, obtaining the following equivalent formulation of (4):

$$\hat{\mathcal{E}}[f;\theta] = \frac{1}{N} \sum_{n,m \in \mathbb{N}} \left(\Sigma_{n,m} \left(\frac{t}{q}\right)^n \left(\frac{1-t}{1-q}\right)^m \right) \tag{6}$$

Note that θ freely occurs in the right-hand side of (6), as it does so in t. If the number of non-empty $T_{n,m}$ is considerably smaller than N, the right-hand side of equation (6), after dropping summands for which $T_{n,m} = \emptyset$ and thus $\Sigma_{n,m} = 0$, provides us with a much shorter sum than (4), which still characterises $\hat{\mathcal{E}}[f;\theta]$. A symbolic representation η of the parameter-dependency of $\hat{\mathcal{E}}[f;\theta]$ can again be obtained by substituting the specific values for N, q, and $\Sigma_{n,m}$ into (6). Based on the resulting expression η, we can construct a logically equivalent, yet syntactically shorter formulation of the constraint (5):

$$\left(\mathcal{E} = \frac{1}{N} \sum_{n,m \in \mathbb{N}} \left(\Sigma_{n,m} \left(\frac{t}{q}\right)^n \left(\frac{1-t}{1-q}\right)^m \right) \right) \wedge \phi \wedge C. \tag{7}$$

This constraint expresses $[f \mapsto \hat{\mathcal{E}}[f;\theta]] \models C$ subject to $\theta \models \phi$, and thus approximates the feasibility condition on θ up to the inaccuracies incurred through sampling and rescaling due to importance sampling. In Sect. 6, we will demonstrate that (7) is amenable to constraint solving for a set of interesting PPHA.

[4] The generalization to arbitrary discrete distributions (fan-out larger than two) controlled by a finite-dimensional vector of parameters is straightforward, as is tackling multiple different subsets $\Delta_i \subset \Sigma$.

5 Existence or Absence of Parameter Instances

As constraint (7) is an arithmetic constraint containing addition, multiplication, and the operations found in the term t as well as in the parameter domain constraint ϕ and the design goal C, it can be solved by SMT solvers addressing the corresponding subset of arithmetic, e.g., iSAT [8]. This provides an automatic feasibility check for (7), i.e., a test whether there is a parameter instance within the domain defined by ϕ which guarantees (modulo sampling errors) satisfaction of constraint C over the expectations. Should this test succeed, it will also deliver a parameter instance. Additional optimization wrt. the expectation $\mathcal{E}[f;\theta]$ can be added on top by a branch-and-prune algorithm, as available in the HySAT II tool [12]. It is, however, obvious that a solution to (7), if existing, guarantees the approximate feasibility condition $[f \mapsto \hat{\mathcal{E}}[f;\theta]] \models C$ only, rather than the desired $[f \mapsto \mathcal{E}[f;\theta]] \models C$. Therefore, we have to account for the statistical approximation error associate to the empirical estimate.

For the sake of simplicity, assume in the sequel that the constraint C on the expectation $\mathcal{E}[f;\theta]$ is of the form $\mathcal{E}[f;\theta] < c$, for some constant $c \in \mathbb{R}$. The generalization to arbitrary constraints, including constraints on multiple different expectations, is straightforward by adopting the concept of δ-weakening discussed in the context of robust interpretations of arithmetic logics [17].

In the following, we consider the case that symbolic checking of the empirical constraint system (7) could not be satisfied. In this case, we know that $\min_\theta \hat{\mathcal{E}}[f;\theta] > c + \varepsilon$, with an additional slackness ε to be defined shortly. We are then interested in the probability that there nevertheless exists a θ' with $\mathcal{E}[f;\theta'] < c$, i.e., $\min_{\theta'} \mathcal{E}[f;\theta'] < c$. To quantify this, let the subindex $S = (X_1, \ldots, X_N)$ indicate the dependency of the estimated expectation on the ensemble of samples drawn from the proposal q, i.e., $\hat{\mathcal{E}}_S[f;\theta] = \frac{1}{N}\sum_i f(x_i)\frac{p(x_i;\theta)}{q(x_i)}$:

$$
\begin{aligned}
P_S\left(\min_\theta \hat{\mathcal{E}}_S[f;\theta] \geq \varepsilon + c \ \wedge\ \min_\theta \mathcal{E}[f;\theta] < c\right) &\leq P_S\left(\min_\theta \hat{\mathcal{E}}_S[f;\theta] \geq \min_\theta \mathcal{E}[f;\theta] + \varepsilon\right) \\
&\overset{\text{Jensen ineq.}}{\leq} P_S\left(\underbrace{\min_\theta \hat{\mathcal{E}}_S[f;\theta]}_{=:g(S)} - \mathcal{E}\left[\min_\theta \hat{\mathcal{E}}_S[f;\theta]\right] \geq \varepsilon\right) = P_S\left(g(S) - \mathcal{E}\left[g(S)\right] \geq \varepsilon\right) \\
&\overset{\text{McDiarmid ineq.}}{\leq} \exp\left(-\frac{\varepsilon^2 N}{4B^2}\right) =: \delta; B := \max_{x,\theta}\frac{p(x;\theta)}{q(x)} \overset{\text{Markov}}{\leq} \left(\max_\theta\left\{\frac{p_\theta}{q}, \frac{1-p_\theta}{1-q}\right\}\right)^{k-1}
\end{aligned}
\tag{8}
$$

The last of these inequalities is specific to our case of a binary Markov chain with samples consisting of k probabilistic transitions.

Therefore, if we are aiming for a confidence of $1-\delta$, we can use equation (8) and check the symbolic constraint system with an adapted threshold $c' = c + \varepsilon(\delta, N) = \sqrt{\frac{\log(\frac{1}{\delta})}{N}} 2B$. If this constraint system is unsatisfiable, we know with probability at least $1-\delta$ that the original constraint system is also unsatisfiable.

Obtaining similar bounds in case we have found a parameter instance for which the empirical constraint system is indeed satisfiable is more involved.

In fact, these bounds are tightly coupled to the generalization bounds within statistical learning theory (see [1,5,22]). Therefore, in case we find a potential satisfying parameter setting using the symbolic constraint system, we simply check the validity of this parameter statistically using another round of naïve sampling, similar to [25], thereby avoiding restricting ourselves to particular function classes for the parameter dependence of the terms t.

Using the above two tests, we can iteratively solve for a parameter satisfying the desired constraint system until one of the tests succeeds, giving us a (statistically) reliable answer. If a parameter instance satisfying (7) is found that nevertheless fails to pass the statistical check, we use the fresh samples to build another symbolic constraint system as in equation (7). We can then use this constraint system to solve for a new parameter value, which we can then check subsequently. To use as much information as possible from the samples, instead of building a completely fresh symbolic constraint system, we simply add the newly constructed constraint system to the previous one (see Algorithm 1). To retain the confidence statement with respect to unsatisfiability when adding more clauses to the constraint system, we have to account for sequential hypothesis testing. This can be achieved by using a Bonferroni correction, i.e., requiring the individual tests to be more confident, relative to the maximal amount of tests to be performed ($\delta^c = \frac{\delta}{I}$ in Algorithm 1).

Algorithm 1. Parameter Fitting by Symbolic Importance Sampling

function SYM-IMP(ϕ, C, confidence δ, number of samples N, max. iterations I)
  $\delta^c \leftarrow \frac{\delta}{I}$; $\theta_0 \leftarrow$ DrawUniform(Θ); $\varepsilon \leftarrow \sqrt{\frac{\log(\frac{1}{\delta^c})}{N}} 2B$; $n \leftarrow 0$; $\hat{\phi}_0 \leftarrow \phi$
  while $n \leq I$ **do**
    $q \leftarrow p(\cdot; \theta_n)$
    $S = (x_1, \ldots, x_N) \leftarrow$ DrawSamples(q, N)
    if CheckSamples(S, δ, ϕ, C) **then**
      return θ_n ▷ Found parameterization satisfying C with prob. $\geq 1 - \delta$
    else
      $\hat{\phi}_{n+1} \leftarrow \hat{\phi}_n \wedge (\eta(S) < c + \varepsilon)$ ▷ Add samples to empirical system
      $\theta_{n+1} \leftarrow$ SolveConstraintSystem($\hat{\phi}_{n+1}$)
      if $\hat{\phi}_{n+1}$ is unsatisfiable **then**
        return Unsat ▷ Original system is unsatisfiable with prob. $\geq 1 - \delta$
      else $n \leftarrow n + 1$
      end if
    end if
  end while
  return Unknown ▷ Reached maximal iterations I
end function

Altogether, we arrive at Algorithm 1 which upon termination within the specified maximal number of iterations either delivers a parameter instance satisfying the design objective with the desired confidence or proves with the desired confidence that no such instance exists.

6 Experiments

In this section, we explore the potential and limitations of the presented approach, exemplified on the PPHA depicted in Figure 1. The PPHA simplistically models a battery being `charge`d and `discharge`d, where switching between those two modes happens randomly, thereby reflecting external influences like weather on a solar panel. As the climatic conditions might change depending on the region, the probabilistic transitions are parameterized. Initially, the battery is in `charge` mode. As long as its capacity has not reached its maximal value (guard: $\ell < 1$), it can switch randomly with probability t_1 to the `discharge` mode. During the transition from `charge` to `discharge` mode, the capacity is reduced. For simplicity, we assume a fixed value of 0.048 (see Figure 1), which reduces the continuous variable ℓ to the new value after the transition to ℓ'. For the probabilistic transitions, we assume the following parametric dependence to include non-linear dependence of the parameters h: $t_1 = t_3 = \sin(h)$, and $t_2 = t_4 = 1 - t_1$, where $h \in [0.0, 0.1]$ is the parameter of interest. As an illustrative objective for this PPHA, we are interested in finding a parameter value such that the charge of the battery reaches a sufficient level at a certain time, e.g., exceeding a threshold level at sunset which provides sufficient power for the following night. This property can be formalized as follows:

> **Goal:** The battery is sufficiently charged at sunset (indicated by time step K) in 90% of the days. The corresponding reward function takes the current time into account: it evaluates to 1 if $\ell \geq 0.98$ at time K, and 0 otherwise. We require this condition to hold with probability $c \geq 0.9$.

Using this formalization, we evaluate the presented approach in terms of both accuracy and efficiency. Using Algorithm 1, we can obtain the following results from our solver characterizing the solution: 'unknown', 'candidate solution', 'unsatisfiable'. If the problem is satisfiable, we expect the solver to return, either 'unknown' or 'candidate solution' due to the general undecidable nature of the problem. However, if the solver returns 'unsatisfiable', we know that with a high likelihood $(1-\delta)$ there is no parameter value within the given domain, such that the constraint system is satisfiable.

Fortunately, the simplicity of the model allows us to calculate maximal and minimal values for the expected values as a function of the parameter quite accurately, thereby enabling us to determine the satisfiability of the problem analytically in some of the settings. Using this analytical result we can determine the fraction of simulations in which we have obtained the most informative result with respect to these analytically obtained satisfiability statements.

For the model described above, we are able to compute the satisfiability for both constraint variants by calculating the expected values using $h = 0.0$ and $h = 0.1$, due to the monoticity of the properties: From these expectations, we can conclude on the existence of a parameter instances for which $\theta \models C$ or $\theta \not\models C$

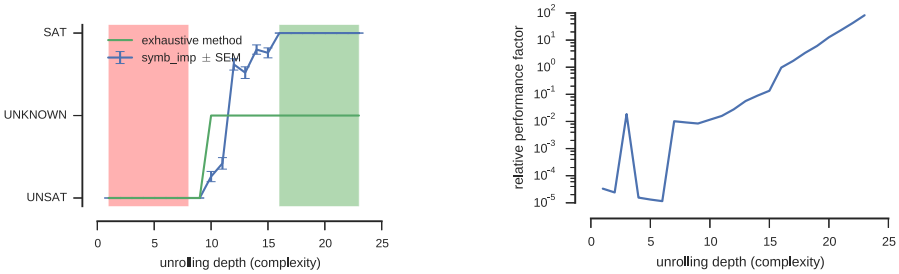
holds[5]. Using these bounds we evaluate the frequency with which the algorithm found the most informative solution as a proxy for the accuracy of the approach.

To judge the improvement of the presented approach, we compare the sampling based verification with a parameter-state exploration approach. As we are considering a bounded model checking approach within a Markov transition system, we can fully unroll the probabilistic transitions and use the same SMT solver to check the existence of a parameter value that result in a satisfaction of the desired property. As the complexity of the satisfaction property increases with increasing unrolling depth, we vary the maximal number of transitions as well as the level of confidence. As we expect the choice of the proposal distribution to crucially influence the effectiveness of the approach, we compare two different choices: one minimizing the possible range B of the fraction $\frac{q(x)}{p(x;\theta)}$ (as a function of the parameter as well as a function of the sampled path x, cf. (8)), and a slightly perturbed version. As a backend SMT solver, we used iSAT3[6].

In Figure 2, we compare the accuracy as well as the run-time of our sampling-based method against a full exploration approach. In the left panel, the obtained (and averaged obtained) result are plotted for either method. For the sampling based method the results are averaged across 50 repetitions, for each of which we used a confidence of $1 - \delta = 0.7$, $N = 15000$ samples, and a maximal number of $I = 3$ iterations. Although none of the algorithms, both sampling and unwinding based, produced any wrong results, the sampling-based algorithm was able to provide more informative results for models with large complexity. Although we know the problem for the settings with higher unrolling depth (≥ 16) to be satisfiable, only the sampling based approach was able to provide a corresponding certificate, while the unwinding scheme returned a 'potentially satisfiable' result, indicated by the UNKNOWN result. The drop in accuracy around unrolling depth $K \approx 15$ can be explained by the fact that the range of possible expected values $\mathcal{E}$ as a function of the parameters overlaps with the confidence interval, rendering the problem harder to answer. As the problem size for naïve scheme of fully unwinding the transition system grows exponentially with the maximal length of paths (K), we expect the speedup compared to the run-time of the sampling based scheme also to be exponential as a function of K. It can be observed that the speedup in the run-time increases with the complexity, as shown in Figure 2b. However, for settings with small number of possible paths, the speedup is less pronounced. In fact, it could also happen that the speedup is below 1 (a slower performance), as the sampling based approach needs to simulate more than actual possible paths for small unrolling depths, leading to an overhead in computation.

[5] These regions are indicated in Figure 2 in green and red respectively. For the region inbetween, we could not analytically calculate the true result. As the sampling-based method returned some satisfiable parameter values, it suggests the satisfiable region to be larger than depicted in the Figure.

[6] https://projects.avacs.org/projects/isat3



(a) Solution found by either method

(b) Sampling-based speed-up

Fig. 2. Comparison of exhaustive and sample based method wrt. accuracy and running-time. For the accuracy (left panel), the analytically guaranteed UNSAT and SAT regions are marked red (left) and green (right) respectively. For the sample-based algorithm an interpolated average ± 1 standard error is plotted.

7 Related Work

The verification of parametric probabilistic models in which certain transition probabilities are given as parameters (or functions thereof) has received considerable attention recently. Most approaches focus on parameter synthesis: for which parameter instances does a given (LTL or probabilistic CTL) formula hold? This question has been tackled in several different settings, varying in the properties, the forms of parameter dependence, as well as the class of systems considered. Han *et al.* [3,11] considered the problem for timed reachability in continuous-time Markov chains, Hahn *et al.* [9] and Pugelli *et al.* [18] for discrete-state Markov decision processes (MDPs). Benedikt *et al.* [2] considered the parameter synthesis as a maximization problem for the probability of satisfying ω-regular properties within an interval Markov chain without further constraints on the parameter dependence. Hahn *et al.* [10] provide an algorithm for computing the rational function of the parameters expressing the probability of reaching a given set of states in a parametric (reward) MDP based on exploiting regular expressions, as initially proposed by Daws [6]. For non-probabilistic systems and linear arithmetic dependence on parameters, the synthesis (i.e., reachability as the dual problem) has been analyzed in [4]. Similarly, in [15] and [14] reachability is analyzed for parametric probability distribution in a finite-state Markov chain. To increase the efficiency of the synthesis problem, [14] restricted the parametric dependence to rational functions. Zhang et *al.,* [25] considered the following problem: Find parameters u such that for a given black box function r, the following holds: $P_X(r(u,x) \in [a,b]) \geq \theta$. For this single objective the probability distribution of x needs to be known and independent of the design parameters u. The presented procedure is similar, as it iterates between optimization and a simulation-based verification step, however, it cannot provide an unsatisfiability statement. The parametric dependence of the probability distributions presented in this paper can also be integrated into a hierarchical optimization

procedure, see [7] for more details. There however, only single objectives were considered instead of multiple constraints as specified in equation (5).

To the best of our knowledge, synthesis wrt. arbitrary first-order objectives over expected rewards has not been considered so far. Parameter synthesis in PHA also seems to be a mostly unexplored research arena.

8 Conclusion

We have discussed a method for automatically finding parameter instances satisfying arbitrary first-order, multi-objective specifications on expected rewards, given a probabilistic hybrid system with parametric probability distributions. Although our approach is based on simulations and hence can only provide statistical guarantees of the property being satisfied, we found that such an approach can rapidly find parameter instances at similar or even better accuracy than exhaustive, safely overapproximating procedures. The probable reason is that the overall number of paths to be analyzed is drastically reduced by the sampling process, thereby rendering the approach less sensitive to the overapproximations typically used by the internal mechanics of the solver used.[7] For this reason, it is to be expected that this accuracy benefit gets even more pronounced for higher-dimensional optimization problems. As our approach effectively tames the dimensionality barrier, which inevitably is hit by both exhaustive procedures and procedures sampling the parameter space, by employing a form of symbolic importance sampling it should scale well to such higher-dimensional problems. This, however, remains subject to future investigations.

References

1. Bartlett, P.L., Mendelson, S.: Rademacher and gaussian complexities: Risk bounds and structural results. J. of Machine Learning Research **3**, 463–482 (2003)
2. Benedikt, M., Lenhardt, R., Worrell, J.: LTL model checking of interval Markov chains. In: Piterman, N., Smolka, S.A. (eds.) TACAS 2013 (ETAPS 2013). LNCS, vol. 7795, pp. 32–46. Springer, Heidelberg (2013)
3. Češka, M., Dannenberg, F., Kwiatkowska, M., Paoletti, N.: Precise parameter synthesis for stochastic biochemical systems. In: Mendes, P., Dada, J.O., Smallbone, K. (eds.) CMSB 2014. LNCS, vol. 8859, pp. 86–98. Springer, Heidelberg (2014)
4. Cimatti, A., Griggio, A., Mover, S., Tonetta, S.: Parameter synthesis with IC3. In: FMCAD (2013)
5. Cortes, C., Mansour, Y., Mohri, M.: Learning bounds for importance weighting. In: Advances in Neural Information Processing Systems (NIPS), pp. 442–450 (2010)
6. Daws, C.: Symbolic and parametric model checking of discrete-time Markov chains. In: Liu, Z., Araki, K. (eds.) ICTAC 2004. LNCS, vol. 3407, pp. 280–294. Springer, Heidelberg (2005)
7. Ellen, C., Gerwinn, S., Fränzle, M.: Statistical model checking for stochastic hybrid systems involving nondeterminism over continuous domains. International Journal on Software Tools for Technology Transfer **17**(4), 485–504 (2015)

[7] The iSAT solver we used for the experiments uses interval arithmetics. Its internal mechanics [8] is closely akin to, yet predates δ-decision procedures [16].

8. Fränzle, M., Herde, C., Teige, T., Ratschan, S., Schubert, T.: Efficient Solving of Large Non-linear Arithmetic Constraint Systems with Complex Boolean Structure. JSAT **1**, 209–236 (2007)
9. Hahn, E.M., Han, T., Zhang, L.: Synthesis for PCTL in parametric Markov decision processes. In: Bobaru, M., Havelund, K., Holzmann, G.J., Joshi, R. (eds.) NFM 2011. LNCS, vol. 6617, pp. 146–161. Springer, Heidelberg (2011)
10. Hahn, E.M., Hermanns, H., Zhang, L.: Probabilistic reachability for parametric Markov models. STTT **13**(1), 3–19 (2011)
11. Han, T., Katoen, J.-P., Mereacre, A.: Approximate parameter synthesis for probabilistic time-bounded reachability. In: IEEE Real-Time Systems Symposium (RTSS), pp. 173–182. IEEE Computer Society (2008)
12. Herde, C., Eggers, A., Fränzle, M., Teige, T.: Analysis of hybrid systems using HySAT. In: Third International Conference on Systems, ICONS 2008, pp. 196–201. IEEE (2008)
13. Hoeffding, W.: Probability inequalities for sums of bounded random variables. Journal of the American Statistical Association, 13–30 (1963)
14. Jansen, N., Corzilius, F., Volk, M., Wimmer, R., Ábrahám, E., Katoen, J.-P., Becker, B.: Accelerating parametric probabilistic verification. In: Norman, G., Sanders, W. (eds.) QEST 2014. LNCS, vol. 8657, pp. 404–420. Springer, Heidelberg (2014)
15. Lanotte, R., Maggiolo-Schettini, A., Troina, A.: Parametric probabilistic transition systems for system design and analysis. Formal Aspects of Computing **19**(1), 93–109 (2007)
16. Liu, B., Kong, S., Gao, S., Zuliani, P., Clarke, E.M.: Parameter synthesis for cardiac cell hybrid models using δ-decisions. In: Mendes, P., Dada, J.O., Smallbone, K. (eds.) CMSB 2014. LNCS, vol. 8859, pp. 99–113. Springer, Heidelberg (2014)
17. Liu, J., Ozay, N.: Abstraction, discretization, and robustness in temporal logic control of dynamical systems. In: Proceedings of the 17th International Conference on Hybrid Systems: Computation and Control, pp. 293–302. ACM (2014)
18. Puggelli, A., Li, W., Sangiovanni-Vincentelli, A.L., Seshia, S.A.: Polynomial-time verification of PCTL properties of MDPs with convex uncertainties. In: Sharygina, N., Veith, H. (eds.) CAV 2013. LNCS, vol. 8044, pp. 527–542. Springer, Heidelberg (2013)
19. Sproston, J.: Decidable model checking of probabilistic hybrid automata. In: Joseph, M. (ed.) FTRTFT 2000. LNCS, vol. 1926, pp. 31–45. Springer, Heidelberg (2000)
20. Sproston, J.: Model Checking for Probabilistic Timed and Hybrid Systems. PhD thesis, School of Computer Science, The University of Birmingham (2001)
21. Tokdar, S.T., Kass, R.E.: Importance sampling: a review. Wiley Interdisciplinary Reviews: Computational Statistics **2**(1), 54–60 (2010)
22. Vapnik, V.N.: Statistical learning theory, vol. 1. Wiley, New York (1998)
23. Wimmer, R., Derisavi, S., Hermanns, H.: Symbolic partition refinement with dynamic balancing of time and space. In: Rubino, G. (ed.) Quantitative Evaluation of Systems (QEST), pp. 65–74. IEEE Computer Science Press (2008)
24. Younes, H.L.S., Kwiatkowska, M., Norman, G., Parker, D.: Numerical vs. statistical probabilistic model checking. International Journal on Software Tools for Technology Transfer (STTT) **8**(3), 216–228 (2006)
25. Zhang, Y., Sankaranarayanan, S., Somenzi, F.: Statistically sound verification and optimization for complex systems. In: Cassez, F., Raskin, J.-F. (eds.) ATVA 2014. LNCS, vol. 8837, pp. 411–427. Springer, Heidelberg (2014)