




Scenario Approach for Parametric Markov Models

Ying Liu^{1,2} , Andrea Turrini^{1,3} , Ernst Moritz Hahn⁴ , Bai Xue¹  ,
and Lijun Zhang^{1,2,3}  

¹ State Key Laboratory of Computer Science, Institute of Software,
Chinese Academy of Sciences, Beijing, China
{xuebai,zhanglj}@ios.ac.cn

² University of Chinese Academy of Sciences, Beijing, China

³ Institute of Intelligent Software Guangzhou, Guangzhou, China

⁴ Formal Methods and Tools, University of Twente, Enschede, The Netherlands

Abstract. In this paper, we propose an approximating framework for analyzing parametric Markov models. Instead of computing complex rational functions encoding the reachability probability and the reward values of the parametric model, we exploit the scenario approach to synthesize a relatively simple polynomial approximation. The approximation is probably approximately correct (PAC), meaning that with high confidence, the approximating function is close to the actual function with an allowable error. With the PAC approximations, one can check properties of the parametric Markov models. We show that the scenario approach can also be used to check PRCTL properties directly – without synthesizing the polynomial at first hand. We have implemented our algorithm in a prototype tool and conducted thorough experiments. The experimental results demonstrate that our tool is able to compute polynomials for more benchmarks than state-of-the-art tools such as PRISM and STORM, confirming the efficacy of our PAC-based synthesis.

1 Introduction

Markov models (see, e.g., [52]) have been widely applied to reason about quantitative properties in numerous domains, such as networked, distributed systems, biological systems [37], and reinforcement learning [4, 59]. Properties analyzed on Markov models can either be simple, such as determining the value of the probability that a certain set of unsafe states is reached and how an expected reward value compares with a specified threshold, or complex, involving employing temporal logics such as PCTL [10, 35] and PRCTL [1]. To verify these properties, various advanced tools have been developed, such as PRISM [44], STORM [24, 36], MRMC [42], CADP 2011 [28], PROPhESY [23] and ISCASMC [33].

In this paper we consider *parametric* discrete time Markov chains (pDTMCs), whose transition probabilities are not required to be constants, but can depend

A. Turrini—Co-first author.

on a set of parameters. For this type of model, the value of the analyzed property can be described as a *function* of the parameters, mapping either to truth values or to numbers. In many cases, these functions are *rational functions*, that is, fractions of co-prime polynomials. The exact rational function is commonly challenging to compute as it often involves polynomials with very high degree [5, 41]. Moreover, the Markov models applied in real life may be very large, with thousands of states, so it is very complicated to compute the rational functions accurately. However, to analyze the properties of such Markov models in practical applications, we often allow for a certain acceptable level of error, without guaranteeing that the given property holds absolutely.

Contribution of the Paper. In this work, we propose an alternative approach to replace the exact solution of function f_φ that describes the value of the analyzed property φ in the given pDTMC and pDTMRM. The main idea is exploiting the scenario approach [16, 18] to learn an *approximating function* – polynomial \tilde{f}_φ with low degree to approximate the actual function f_φ with probably approximately correct (PAC) guarantee, i.e., with high confidence $1 - \eta$, the probability that the approximation error is within an error margin λ is at least $1 - \epsilon$.

We demonstrate how to use PAC approximation \tilde{f}_φ to synthesize parameters and analyze the properties of original functions f_φ . We use PAC approximation to check the safe region of the parameter space and some global probability properties. We also extend to reward properties: we show how to use PAC approximation to estimate the lower bound of the expectation of f_φ over the domain of the parameters. Extending our approach to parametric MDPs is also feasible, as long as we treat the MDP strategy as in [2], we allow the strategy to change for the different MDP instances.

Experimental results show that our prototype PACPMA can solve more properties under the same conditions than the state-of-the-art verification tools STORM and PRISM, and provide PAC approximations with a statistical guarantee. We demonstrate that as the degree of the polynomial approximation increases, the computed error margin λ approaches zero, indicating that the polynomial with higher degree provides a more accurate approximation of the original function. As for the accuracy of the approximation, we show that PAC approximation \tilde{f}_φ can approximate f_φ two to three orders of magnitude more accurately than the Taylor expansion of the actual function f_φ . We also demonstrate that the PAC approximation can approximately capture the lower bound of the expected reward of pDTMRM with high confidence, thus can help to verify the reward properties.

Related Work. Model checking of parametric Markov models is not a new area and a number of related works exist, each with different strengths and weaknesses. In the following, we demarcate our work from the existing ones.

Daws has devised a language-theoretic approach to solve the reachability problem of parametric Markov chains [22], where the model is viewed as a finite

automaton. Based on the state elimination approach [38], the regular expression describing the language of such an automaton is computed, which is transformed into a rational function over the parameters of the model in a postprocessing step. The method has been improved by intertwining the state elimination and the computation of the rational function [32], this improved algorithm has been implemented in the tool PARAM [31]. PARAM also supports bounded reachability, relying on matrix-vector multiplication with rational function entries, and reachability rewards [11, 25]. All these works [31, 32] compute the precise rational function that describes the property of interest. Unfortunately, it is challenging to evaluate it, due to the large coefficients and high exponents. Moreover, the works discussed above do not consider properties specified by a temporal logic.

Several improvements have been proposed in later works. Jansen et al. [39] perform the state elimination in a more systematic order, often leading to better performance in practice. The work [53] provides the first sound and feasible technique for parameter synthesis of Markov decision processes, Spel et al. [57] achieved efficient parameter optimization by combining monotonicity checking and parameter lifting, allowing for further evaluation of the safety of parameter space. The work [27] uses arithmetic circuits, which are DAG-like structures, to represent such rational functions. A further work [30] follows a related approach to solve (potentially nested) PRCTL formulas for Markov decision processes: the state space is divided into hyperrectangles, and one has to show that a particular decision is optimal for a whole region. The work [5] improves the computation of the rational function by means of a fraction-free Gaussian elimination; the experimental evaluation confirms its effectiveness. There are also methods for checking parametric continuous time Markov chains [34], by using a scenario approach [3] or by being based on Gaussian processes [13, 14]. A recent work [26] proposes a fast parametric model checking method named fPMC, which extends the current parametric model checking approaches to systems with complex behaviors and multiple parameters. A recent survey [40] provides an overview of parameter synthesis for Markov models. In the above Markov model checking methods, the accurate algorithm is computationally complex, while the approximate algorithms rarely provide a probability guarantee. However, our proposed method for verifying parametric Markov models based on scenario approach provides both efficiency and probabilistic guarantee.

The scenario approach was first introduced in [15], based on constraint sampling to deal with uncertainty in optimization. The works [16, 18, 19] study a probabilistic solution framework for robust properties, the work [50] proposes a method to solve chance constrained optimization problems lying between robust optimization and scenario approach, which does not require prior knowledge of the probability distribution of the parameters. Based on [15, 16], the work [17] allows violating some of the sampled constraints in order to improve the optimization value, and the work [58] expands the scenario optimization problem to multi-stage problems. Recently, the scenario approach has been applied to verify safety properties of black-box continuous time dynamical systems [60] and the robustness of neural networks [48].

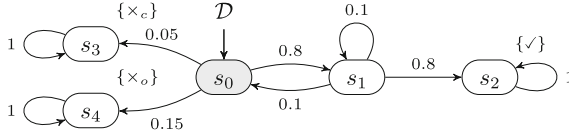


Fig. 1. An example of discrete time Markov chain

The most related to our work is [2], which also applies the scenario approach for analyzing parametric Markov chains and Markov decision processes. The main difference with our work is that in [2], the authors compute the probability that the instances of the parametric MDP satisfy a given property φ , by sampling the parameter values according to some assumed distribution. Instead, our work targets at computing an approximation of the complicated function f_φ depending on the parameters – such as the one corresponding to the reachability probability φ . Our framework can bound the error between the actual function and the approximation we compute. Moreover, as a side result, our PAC approximations can be used for visualizing the reachability probability, finding counterexamples, and analyzing the properties of the original functions with a certain confidence.

Organization of the Paper. After giving in Sect. 2 some preliminaries, models, and logic we use in this paper, in Sect. 3 we present our PAC-based model checking approach; we evaluate it empirically in Sect. 4 before concluding the paper in Sect. 5 with some final remarks.

Due to space constraints, the proofs are provided in the technical report [49].

2 Preliminaries

In this section, we first recall DTMCs, a well-known probabilistic model (see, e.g., [6]), reward structures, the probabilistic logic PRCTL we adopt to express properties on them, and then consider their extension with parameters.

2.1 Probabilistic Models

Definition 1. Given a finite set of atomic propositions AP , a (labelled) discrete time Markov chain (DTMC) \mathcal{D} is a tuple $\mathcal{D} = (S, \bar{s}, \mathbf{P}, L)$ where S is a finite set of states; $\bar{s} \in S$ is the initial state; $\mathbf{P}: S \times S \rightarrow [0, 1]$ is a transition function such that for each $s \in S$, we have $\sum_{s' \in S} \mathbf{P}(s, s') = 1$; and $L: S \rightarrow 2^{AP}$ is a labelling function.

The *underlying graph* of a DTMC $\mathcal{D} = (S, \bar{s}, \mathbf{P}, L)$ is a directed graph (V, E) with $V = S$ as vertices and $E = \{(s, s') \in S \times S \mid \mathbf{P}(s, s') > 0\}$ as edges.

As an example of DTMC, consider the DTMC \mathcal{D} shown in Fig. 1. \mathcal{D} has 5 states (from s_0 to s_4), with s_0 being the initial one (marked with the gray background and the small incoming arrow); transitions with probability larger

than 0 are depicted as arrows, so for example we have $\mathbf{P}(s_0, s_1) = 0.8 > 0$, while the labels assigned to each state are shown on the top-right corner of the state itself, e.g., $L(s_2) = \{\checkmark\}$ while $L(s_0) = \emptyset$.

DTMCs can be equipped with reward structures that assign values to states and transitions; such reward structures can be used to count the number of transitions taken so far or to attach “costs” or “gains” to the DTMC.

Definition 2. A discrete time Markov reward model (DTMRM) \mathcal{R} is a pair $\mathcal{R} = (\mathcal{D}, \mathbf{r})$ where \mathcal{D} is a DTMC and $\mathbf{r}: S \cup (S \times S) \rightarrow \mathbb{R}_{\geq 0}$ is a reward function.

For example, the reward function \mathbf{c} defined as $\mathbf{c}(s) = 0$ and $\mathbf{c}(s, s') = 1$ for each $s, s' \in S$ allows us to “count” the number of steps taken by the DTMC.

Let \mathcal{D} be a DTMC; a *path* π of \mathcal{D} is a (possibly infinite) sequence of states $\pi = s_0 s_1 s_2 \cdots$ such that for each meaningful $i \in \mathbb{N}$, we have $\mathbf{P}(s_i, s_{i+1}) > 0$; we write π_i to indicate the state s_i . We let $\text{Paths}^*(\mathcal{D})$ ($\text{Paths}(\mathcal{D})$, resp.) denote the sets of all finite (infinite, resp.) paths of \mathcal{D} . Given a finite path $\pi = s_0 s_1 s_2 \cdots s_n$, we denote by $|\pi|$ the number of states $n + 1$ of π .

Given a DTMRM $\mathcal{R} = (\mathcal{D}, \mathbf{r})$, we can define the *expected cumulative reward* $\text{ExpRew}_s^{\mathcal{R}}$ as follows (cf. [6, 32, 43]): given set $T \subseteq S$ of states, $\text{ExpRew}_s^{\mathcal{R}}(T)$ is the expectation of the random variable $X^T: \text{Paths}(\mathcal{D}) \rightarrow \mathbb{R}_{\geq 0}$ with respect to the X^T defined as follows:

$$X^T(\pi) = \begin{cases} 0 & \text{if } \pi_0 \in T, \\ \infty & \text{if } \pi_i \notin T \text{ for each } i \in \mathbb{N}, \\ \sum_{i=0}^{\min\{n \in \mathbb{N} \mid \pi_n \in T\} - 1} \mathbf{r}(\pi_i) + \mathbf{r}(\pi_i, \pi_{i+1}) & \text{otherwise.} \end{cases}$$

2.2 Probabilistic Reward Logic PRCTL

To express properties about probabilistic models with rewards, we use formulas from PRCTL, the Probabilistic Reward CTL logic [1], that extends PCTL [10, 35] with rewards. Such formulas are constructed according to the following grammar, where φ is a *state formula* and ψ is a *path formula*:

$$\begin{aligned} \varphi &::= a \mid \neg\varphi \mid \varphi \wedge \varphi \mid \mathbf{P}_{\bowtie p}(\psi) \mid \mathbf{R}_{\bowtie r}(\mathbf{F}\varphi) \\ \psi &::= \mathbf{X}\varphi \mid \varphi \mathbf{U} \varphi \mid \varphi \mathbf{U}^{\leq k} \varphi \end{aligned}$$

where $a \in AP$, $\bowtie \in \{<, \leq, \geq, >\}$, $p \in [0, 1]$, $r \in \mathbb{R}_{\geq 0}$, and $k \in \mathbb{N}$. We use freely the usually derived operators, like $\varphi_1 \vee \varphi_2 = \neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\mathbf{tt} = a \vee \neg a$, and $\mathbf{F}\varphi = \mathbf{tt} \mathbf{U} \varphi$. The PCTL logic is just PRCTL without the $\mathbf{R}_{\bowtie r}(\mathbf{F}\varphi)$ operator.

The semantics of a state formula φ and of a path formula ψ is given with respect to a state s and a path π of a DTMRM $\mathcal{R} = (\mathcal{D}, \mathbf{r})$, respectively. The semantics is standard for all Boolean and temporal operators (see, e.g., [6, 20]); for the $\mathbf{P}_{\bowtie p}$ operator, it is defined as $s \models \mathbf{P}_{\bowtie p}(\psi)$ iff $\text{Pr}_s(\{\pi \in \text{Paths}(\mathcal{D}) \mid \pi \models \psi\}) \bowtie p$ and, similarly, $s \models \mathbf{R}_{\bowtie r}(\psi)$ iff $\text{ExpRew}_s(\{\pi \in \text{Paths}(\mathcal{D}) \mid \pi \models \psi\}) \bowtie r$.

With some abuse of notation, we write $\mathcal{R} \models \varphi$ if $\bar{s} \models \varphi$; we also consider $\mathbf{P}_{=?}(\psi)$ and $\mathbf{R}_{=?}(\psi)$ as PRCTL formulas, asking to compute the probability (resp.

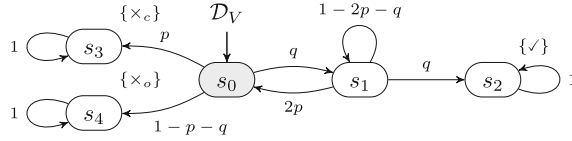


Fig. 2. An example of parametric discrete time Markov chain

expected reward) of satisfying ψ in the initial state \bar{s} of \mathcal{R} , i.e., to compute the value $Pr_{\bar{s}}(\{\pi \in Paths(\mathcal{D}) \mid \pi \models \psi\})$ (resp. $ExpRew_{\bar{s}}(\{\pi \in Paths(\mathcal{D}) \mid \pi \models \psi\})$).

Consider the DTMC \mathcal{D} shown in Fig. 1. As an example of PRCTL formula, there is $P_{=?}(\mathbf{F}\checkmark)$ that asks to compute the probability of eventually reaching a state labelled with \checkmark , for which we have $P_{=?}(\mathbf{F}\checkmark) \approx 0.78$.

2.3 Parametric Models

We now recall the definition of parametric models from [30,32]. Given a finite set of *variables*, or *parameters*, $V = \{v_1, \dots, v_n\}$, let $\mathbf{v} = (v_1, \dots, v_n)$ denote the vector of parameters and $range: V \rightarrow \mathbb{R}$ be the function assigning to each parameter $v \in V$ its closed interval $range(v) = [L_v, U_v] \subseteq \mathbb{R}$ of valid values. Given the ring \mathcal{P}_V of the polynomials with variables V over the field \mathbb{R} of real numbers, a *rational function* f is a fraction $f(\mathbf{v}) = \frac{g_1(\mathbf{v})}{g_2(\mathbf{v})}$ where $g_1, g_2 \in \mathcal{P}_V$; let \mathcal{F}_V denote the set of rational functions. An *evaluation* ν is a function $\nu: V \rightarrow \mathbb{R}$ such that for each $v \in V$, $\nu(v) \in range(v)$. Given $f = \frac{g_1}{g_2} \in \mathcal{F}_V$ and an evaluation ν , we denote by $f\langle\nu\rangle$ the rational number $f(\nu(\mathbf{v})) = f(\nu(v_1), \dots, \nu(v_n))$; we assume that $f\langle\nu\rangle$ is well defined for each evaluation ν , that is, we assume that $g_2\langle\nu\rangle \neq 0$ for each evaluation ν .

Definition 3. Given a finite set of parameters V , a parametric discrete time Markov chain (*pDTMC*) \mathcal{D}_V with parameters V is a tuple $\mathcal{D}_V = (S, \bar{s}, \mathbf{P}, L)$ where S , \bar{s} , and L are as in Definition 1, while $\mathbf{P}: S \times S \rightarrow \mathcal{F}_V$.

Definition 4. Given a pDTMC $\mathcal{D}_V = (S, \bar{s}, \mathbf{P}, L)$, an evaluation ν induces the DTMC $\mathcal{D}\langle\nu\rangle = (S, \bar{s}, \mathbf{P}_\nu, L)$, provided that $\mathbf{P}_\nu(s, s') = \mathbf{P}(s, s')\langle\nu\rangle$ for each $s, s' \in S$ satisfies the conditions given in Definition 1.

The extension to parametric DTMRMs (pDTMRMs) is trivial: a pDTMRM \mathcal{R}_V is just a pair $\mathcal{R}_V = (\mathcal{D}_V, \tau)$ where \mathcal{D}_V is a pDTMC and τ is a reward function.

To simplify the presentation and ensure that the underlying graph of \mathcal{D}_V does not depend on the actual evaluation, we make the following assumption:

Assumption 1 (cf. [30]). Given a pDTMC \mathcal{D}_V , for each pair of evaluations ν_1 and ν_2 , for the induced DTMCs $\mathcal{D}_V\langle\nu_1\rangle$ and $\mathcal{D}_V\langle\nu_2\rangle$ we have that for each $s, s' \in S$, it holds that $\mathbf{P}_{\nu_1}(s, s') = 0$ if and only if $\mathbf{P}_{\nu_2}(s, s') = 0$.

By this assumption, either a state s' has probability 0 to be reached from s (i.e., it is not reachable) independently of the evaluation, or it is always reachable, with possibly different probability values.

As an example of pDTMC, consider the model shown in Fig. 2: now, p and q are the parameters, with e.g. $\text{range}(p) = [0.01, 0.09]$ and $\text{range}(q) = [0.25, 0.8]$. One evaluation is $\nu(p) = 0.05$ and $\nu(q) = 0.8$, which gives us the DTMC shown in Fig. 1. The rational function corresponding to the PRCTL formula $P_{=?}(\mathbf{F}\checkmark)$ is $\frac{q^2}{q+2p-2pq} \approx 0.78$ when evaluated on ν , as one would expect.

3 Probably Approximately Correct Function Synthesis

In this section, we show how to approximate the exact functions for the properties of Markov models such as reachability probability with low-degree polynomials, while providing a statistical PAC guarantee on the closeness of the approximating polynomial with the approximated function.

3.1 Probably Approximately Correct Models

Our method provides a PAC approximation, with respect to the given significance level η and error rate ε . First, we define the PAC approximation of a generic function f as follows.

Definition 5. *Given a set of n variables $V = \{v_1, \dots, v_n\}$, their domain $X = \prod_{i=1}^n \text{range}(v_i)$, and a function $f: X \rightarrow \mathbb{R}$, let P be a probability measure over X , $\lambda \in \mathbb{R}_{\geq 0}$ be a margin to measure the approximation error, and $\varepsilon, \eta \in (0, 1]$ be an error rate and a significance level, respectively.*

We say that the polynomial $\tilde{f} \in \mathcal{P}_V$ is a PAC approximation of f with (ε, η) -guarantee if, with confidence $1 - \eta$, the following condition holds:

$$P(|\tilde{f}(\mathbf{v}) - f(\mathbf{v})| \leq \lambda) \geq 1 - \varepsilon.$$

In this work, we assume that P is the uniform distribution on the domain $X = \prod_{i=1}^n \text{range}(v_i)$ unless otherwise specified. Intuitively, our aim is to make the PAC approximation \tilde{f} as close as possible to f , so we introduce the margin λ to describe how close the two functions are. The two statistical parameters η and ε are the significance level and error rate, respectively; they are used to measure how often the difference between \tilde{f} and f respects the threshold λ . Specifically, the significance level η is a threshold set to describe the degree of risk of accepting an error while the error rate ε is used to describe the probability that the difference between the value $f(\mathbf{v})$ and $\tilde{f}(\mathbf{v})$ obtained by randomly sampled \mathbf{v} in the domain X exceeds λ , so we can adjust these parameters to change the quality of the approximation.

3.2 The Scenario Approach

PAC approximation is inspired by the scenario approach proposed in [16, 18], where the scenario approach was originally applied to robust convex programming problems [7, 8, 29]. Robust convex programming problems are a type of uncertain convex optimization problem, which has a general form as follows:

$$\begin{aligned} \min_{\theta \in \Theta \subseteq \mathbb{R}^m} \quad & \mathbf{a}^T \theta \\ \text{s.t.} \quad & f_\omega(\theta) \leq 0 \quad \forall \omega \in \Omega \end{aligned} \tag{1}$$

under the assumption that $f_\omega : \Theta \rightarrow \mathbb{R}$ is a convex function of $\theta \in \Theta$ for every uncertain parameter $\omega \in \Omega$. We also assume that the set Θ is convex and closed.

The set Ω has an infinite number of elements in general, the main obstacle on solving the optimization problem (1) is that it has infinitely many constraints. In most cases, robust convex optimization problems are NP-hard [7, 8]. Instead of solving the problem (1), based on the famous Helly theorem in convex analysis [54], the work [16] transform problem (1) to scenario optimization problem as formalized in Definition 6 by using finitely many constraints, while providing statistical guarantee on the error rate made with respect to the exact solution of (1).

Definition 6. *Let P be a probability measure over set Ω and $\omega_1, \dots, \omega_l$ be l independent identically distributed samples taken from Ω according to P . The scenario optimization problem corresponding to the problem (1) is defined as*

$$\begin{aligned} \min_{\theta \in \Theta \subseteq \mathbb{R}^m} \quad & \mathbf{a}^T \theta \\ \text{s.t.} \quad & \bigwedge_{i=1}^l f_{\omega_i}(\theta) \leq 0 \quad \omega_i \in \Omega \end{aligned} \tag{2}$$

The optimization problem (2) can be seen as the relaxation of the problem (1), since we do not require that the optimal solution θ_l^* of the problem (2) satisfies all constraints $f_\omega(\theta_l^*) \leq 0$ for each $\omega \in \Omega$, but only the constraints corresponding to the l samples from Ω according to P . The issue now is how to provide a strong enough guarantee that the optimal solution θ_l^* of (2) also satisfies the other constraints $f_\omega(\theta) \leq 0$ with $\omega \in \Omega \setminus \{\omega_i\}_{i=1}^l$ we have not considered.

To answer this question, an *error rate* ε is introduced to bound the probability that the solution θ_l^* violates the constraints of problem (1); we denote by η the *significance level* with respect to the random sampling solution algorithm. Statistics theory ensures that as the number of samples l increases, the probability that the optimal solution of the optimization problem (2) violates the other unseen constraints will tend to zero rapidly. The minimal number of sampled points l is related to the error rate $\varepsilon \in (0, 1]$ and significance level $\eta \in (0, 1]$ by:

Theorem 1 ([18]). *If the optimization problem (2) is feasible and has a unique optimal solution θ_l^* , then $P(f_\omega(\theta_l^*) > 0) < \varepsilon$, with confidence at least $1 - \eta$, provided that the number of constraints l satisfies*

$$l \geq \frac{2}{\varepsilon} \cdot \left(\ln \frac{1}{\eta} + m \right),$$

where m is the dimension of θ , that is, $\theta \in \Theta \subseteq \mathbb{R}^m$, ε and η are the given error rate and significance level, respectively.

Theorem 1 indicates that the number of sampling points can be flexibly changed according to the error rate, confidence level, and the number of parameters. We can observe that when the error rate ε is fixed, the number of sampling

points l is linearly related to the number of parameters m . In Theorem 1, we assume that the optimization problem (2) has a unique optimal solution θ_i^* . This is not a restriction in general, since for multiple optimal solutions we can just use the Tie-break rule [16] to get a unique optimal solution.

3.3 Synthesizing Parametric Functions

We now apply the above scenario approach to the synthesis of the parametric functions for pDTMRMs. Given a pDTMRM $\mathcal{R}_V = (\mathcal{D}_V, \mathbf{r})$ with $\mathcal{D}_V = (S, \bar{s}, \mathbf{P}, L)$, let \mathbf{v} denote the vector of parameters (v_1, \dots, v_n) of \mathcal{D}_V . For a PRCTL state formula φ , the analytic function $f_\varphi(\mathbf{v})$, representing the probability or the expected reward of the paths satisfying φ in the pDTMRM \mathcal{R}_V , can be a rational function with a very complicated form [31, 32] since the polynomials in these rational functions may have exponentially many terms. Our aim is to approximate the function $f_\varphi(\mathbf{v})$ with some low degree polynomial $\tilde{f}_\varphi(\mathbf{v})$.

The reason why we choose a polynomial $\tilde{f}_\varphi(\mathbf{v})$ with low degree to fit the rational function $f_\varphi(\mathbf{v})$ is that the graph of polynomials $\tilde{f}_\varphi(\mathbf{v})$ and original functions $f_\varphi(\mathbf{v})$ are both surfaces and the polynomial $\tilde{f}_\varphi(\mathbf{v})$ can approximate the rational function $f_\varphi(\mathbf{v})$ well if we synthesize appropriately the coefficients $\mathbf{c} = (\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2)$ of the polynomial by learning them.

It is worth mentioning that no matter how complicated the function $f_\varphi(\mathbf{v})$ is (it could also be any kind of function other than rational functions), we can still obtain an approximating polynomial $\tilde{f}_\varphi(\mathbf{v})$ of $f_\varphi(\mathbf{v})$ by solving an optimization problem, and utilize it to analyze various properties the original function $f_\varphi(\mathbf{v})$ may satisfy. In the remainder of this section, we show how we synthesize such coefficients \mathbf{c} , and thus the polynomial; we first introduce some notations.

Given the vector of parameters \mathbf{v} and a degree $d \in \mathbb{N}$, we denote by \mathbf{v}^d the vector of monomials $\mathbf{v}^d = (\mathbf{v}^\alpha)_{\|\alpha\|_1=d}$, where each monomial \mathbf{v}^α is defined as $\mathbf{v}^\alpha = v_1^{\alpha_1} v_2^{\alpha_2} \dots v_n^{\alpha_n}$, with $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{N}^n$ and $\|\alpha\|_1 = \sum_{i=1}^n \alpha_i$. Then, we associate a vector \mathbf{c}_i of coefficients to each of the monomials in the vector $(\mathbf{v}^i)_{i=0}^d$, obtaining the PAC approximation $\tilde{f}(\mathbf{v}) = \sum_{i=0}^d \mathbf{c}_i \cdot \mathbf{v}^i$. For example, if the pDTMC \mathcal{D}_V has two parameters v_1 and v_2 , then for $d = 2$ we get the quadratic polynomial $\tilde{f}(\mathbf{v}) = \mathbf{c}_0 + \mathbf{c}_1 \cdot \mathbf{v} + \mathbf{c}_2 \cdot \mathbf{v}^2 = c_0 + (c_{11} \cdot v_1 + c_{12} \cdot v_2) + (c_{21} \cdot v_1^2 + c_{22} \cdot v_1 \cdot v_2 + c_{23} \cdot v_2^2)$. In general, for n parameters and a polynomial of degree d , we need $\binom{n+d}{n}$ coefficients.

Given a PAC approximation schema $\tilde{f}(\mathbf{v}) = \sum_{i=0}^d \mathbf{c}_i \cdot \mathbf{v}^i = \mathbf{c} \cdot (1, \mathbf{v}, \dots, \mathbf{v}^d)^T$, we solve the following Linear Programming (LP) problem with infinitely many constraints to learn the coefficients $\mathbf{c} = (\mathbf{c}_i)_{i=0}^d$ of the polynomial $\tilde{f}(\mathbf{v})$:

$$\begin{aligned} \min_{\mathbf{c}, \lambda} \quad & \lambda \\ \text{s.t.} \quad & -\lambda \leq f(\mathbf{v}) - \mathbf{c} \cdot (1, \mathbf{v}, \dots, \mathbf{v}^d)^T \leq \lambda, \quad \forall \mathbf{v} \in X, \\ & \mathbf{c} \in \mathbb{R}^{\binom{n+d}{n}}, \lambda \geq 0 \end{aligned} \quad (3)$$

where $f(\mathbf{v})$ is the analytic function on the domain $X = \prod_{i=1}^n \text{range}(v_i)$, the domain $\mathbb{R}^{\binom{n+d}{n}}$ of vectors \mathbf{c} is convex and closed, and the constraints of (3) are

continuous convex functions with respect to the variable \mathbf{c} for any \mathbf{v} , which satisfies the condition of Theorem 1. Note that for pDTMRMs we do not need to compute the rational function f_φ used as f in problem (3) to get its value on \mathbf{v} , since we can first instantiate the pDTMRM with \mathbf{v} and then compute the value of φ in the instantiated DTMRM.

Given the error rate ε and the significance level η , by Theorem 1 we need only to independently and identically sample at least $l \geq \frac{2}{\varepsilon} \left(\ln \frac{1}{\eta} + \binom{n+d}{n} + 1 \right)$ points $\tilde{X} = \{\mathbf{v}_i\}_{i=1}^l$ to form the constraints used in the relaxed LP problem, as done in the problem (2). Concretely, we get the following LP problem:

$$\begin{aligned} \min_{\mathbf{c}, \lambda} \quad & \lambda \\ \text{s.t.} \quad & \bigwedge_{i=1}^l -\lambda \leq f(\mathbf{v}_i) - \mathbf{c} \cdot (1, \mathbf{v}_i, \dots, \mathbf{v}_i^d)^T \leq \lambda, \quad \forall \mathbf{v}_i \in \tilde{X}, \\ & \mathbf{c} \in \mathbb{R}^{\binom{n+d}{d}}, \lambda \geq 0. \end{aligned} \quad (4)$$

We solve the optimization problem (4) to get the coefficients \mathbf{c} , hence the PAC approximation \tilde{f} of the original function f , with the statistical guarantees given by Definition 5; in the context of a pDTMRM \mathcal{R}_V and a PRCTL state formula φ , we get the PAC approximation \tilde{f}_φ of the original function f_φ .

3.4 PRCTL Property Analysis

Given the probabilistic formula $\varphi = \mathbf{P}_{=?}(\psi)$ with path formula ψ , we can obviously use the PAC approximation \tilde{f}_φ to check whether the domain of parameters X is safe, with PAC guarantee. In this section, we introduce a direct PAC based approach for checking domain's safety, without having to learn the approximations first. Then, we consider linear approximations and discuss how counterexamples can be generated in this case before showing how the polynomial PAC approximation \tilde{f}_φ can be used to analyze global properties of f_φ over the whole parameter space X . Lastly, we present how to extend the approach to the reward formula $\varphi = \mathbf{R}_{=?}(\mathbf{F}\varphi')$.

Definition 7 (Safe Region). *Let $X = \prod_{i=1}^n \text{range}(v_i)$ be the domain of a set of n parameters V . Given a function $f: X \rightarrow \mathbb{R}_{\geq 0}$ and a safety level $\zeta \in \mathbb{R}_{\geq 0}$, we say that the point $\mathbf{v} \in V$ is safe if and only if $f(\mathbf{v}) < \zeta$; we call X safe if and only if each $\mathbf{v} \in V$ is safe.*

Intuitively, we hope that the probability of the pDTMRM \mathcal{R}_V to reach an unsafe state under any choice of the parameters will be less than the given safety level, to check whether the domain X of the parameters is safe, we can resort to solve the following optimization problem with respect to the given error rate ε and significance level η , and compare the obtained optimal solution λ^* with ζ :

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & f(\mathbf{v}) \leq \lambda \quad \forall \mathbf{v} \in \tilde{X}, \end{aligned} \quad (5)$$

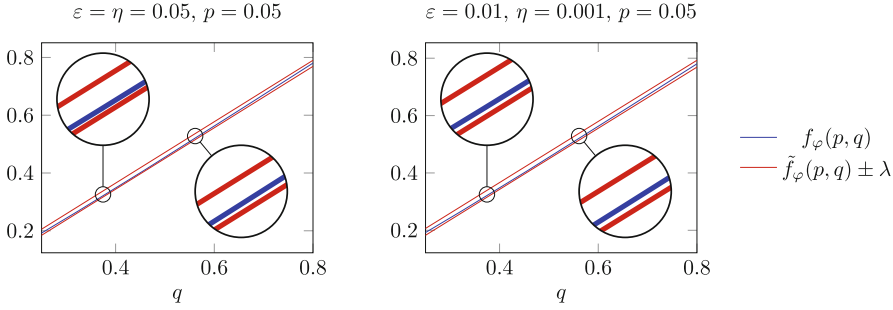


Fig. 3. The rational function $f_\varphi(p, q) = \frac{q^2}{q+2p-2pq}$ and its linear approximations $\tilde{f}_\varphi(p, q)$ with different choices of ε and η

where $\tilde{X} \subseteq X$ is a set of samples such that $|\tilde{X}| \geq \left\lceil \frac{2}{\varepsilon} \cdot (\ln \frac{1}{\eta} + 1) \right\rceil$. The optimization problem (5) can be solved in time $\mathcal{O}(|\tilde{X}|)$, since it only needs to compute the maximum value of $f_\varphi(\mathbf{v})$ for $\mathbf{v} \in \tilde{X}$ as the optimal solution λ^* . Although the calculation is very simple, polynomials with degree 0, i.e., constants, also have good probability and statistical meaning, so we have the following result as a direct consequence of the definitions:

Lemma 1. *Given the safety level ζ , if the optimal solution λ^* of the problem (5) satisfies $\lambda^* < \zeta$, then the domain X is safe with (ε, η) -guarantee. Otherwise, if $\lambda^* \geq \zeta$, then the parameter point $\mathbf{v}^* \in \tilde{X}$ corresponding to λ^* is unsafe.*

By Lemma 1, we can analyze with (ε, η) -guarantee whether the parameter space is safe or not. For example, consider the pDTMC \mathcal{D}_V shown in Fig. 2 and the safety property $\mathbf{P}_{<0.8}(\mathbf{F}(\times_c \vee \times_o))$. If we set $\varepsilon = \eta = 0.05$, by sampling in the region $X = [0.01, 0.09] \times [0.25, 0.8]$ at least 160 points and solving the resulting optimization problem (5), we get the optimal value $\lambda^* = 0.747$ by rounding to three decimals. Since $\lambda^* = 0.747 < 0.8$, by Lemma 1, the region X is safe with $(0.05, 0.05)$ -guarantee.

Linear PAC Approximation and Counterexamples. Since constants can approximate the maximum value of the function f with the given (ε, η) -PAC guarantee, linear functions can also be used to approximate f , which are more precise than constants. Also, we can check whether there is an unsafe region in the domain of parameters X with a given confidence, by the following Lemma 2, and further search counterexamples by linear PAC approximations.

Lemma 2. *Given the domain of parameters X , a function $f: X \rightarrow \mathbb{R}_{\geq 0}$, and a probability measure P over X , let \tilde{f} be a PAC approximation of f with (ε, η) -guarantee. Given the safety level $\zeta \in \mathbb{R}_{\geq 0}$, if for each $\mathbf{v} \in X$ we have $\tilde{f}(\mathbf{v}) + \lambda < \zeta$, then $P(f(\mathbf{v}) < \zeta) \geq 1 - \varepsilon$ holds with confidence $1 - \eta$. In turn, if $P(\tilde{f}(\mathbf{v}) - \lambda > \zeta) > \varepsilon$, then there exist $\mathbf{v} \in X$ such that $f(\mathbf{v}) > \zeta$ holds with confidence $1 - \eta$.*

The plots in Fig. 3 show the results of applying linear PAC approximation on the function $f_\varphi(p, q)$, with $\varphi = \mathbb{P}_{=?}(\mathbf{F}\checkmark)$, for the pDTMC \mathcal{D}_V shown in Fig. 2. We sampled 280 points for $\varepsilon = \eta = 0.05$ and 2182 points for $\varepsilon = 0.01$ and $\eta = 0.001$, respectively, according to Thm. 1. The plot on the left, where we fix the parameter $p = 0.05$, shows that even if we sample just 280 points, $f_\varphi(p, q)$ and $\tilde{f}_\varphi(p, q)$ are closer than the computed margin λ . For the case $\varepsilon = \eta = 0.05$, the linear approximation is $\tilde{f}_\varphi(p, q) = -0.035 + 1.063q - 0.718p$ with $\lambda = 0.011$ by rounding the coefficients to three decimals. We can easily check that for each $(p, q) \in X$ we have $\tilde{f}_\varphi(p, q) + \lambda < 0.85$ by linear programming, so $X = [0.01, 0.09] \times [0.25, 0.8]$ is a 0.85-safe region with respect to $f_\varphi(p, q)$ with (0.05, 0.05)-guarantee. However, if we set $\zeta = 0.6$, we can prove $P(\tilde{f}_\varphi - \lambda > \zeta) = 0.288 > \varepsilon = 0.05$, so by Lemma 2 we get that there exist an unsafe region such that $f(p, q) > \zeta$, with confidence 95%.

We can take advantage of the easy computation of linear programming with linear functions to further search for potential counterexamples that may exist. The maximum value of \tilde{f}_φ can be found at (0.01, 0.8), according to the linearity of \tilde{f}_φ , so we can instantiate the pDTMC \mathcal{D}_V in Fig. 2 with the parameter point (0.01, 0.8) to get that $f_\varphi(p, q) = 0.796$. Since $f_\varphi(p, q) > 0.6$ for the safety level $\zeta = 0.6$, we can claim that the *real counterexample* (0.01, 0.8) is found. In the case that the parameter point $\mathbf{v}_0 = (p, q)$ corresponding to maximum value of \tilde{f}_φ is a spurious counterexample for the pDTMC with respect to φ , we can learn a more precise approximation by adding \mathbf{v}_0 to \tilde{X} . One may also divide the domain X into several subdomains and analyze each of them separately.

As for the computational complexity, it is easy to find the maximum value of a linear function by linear programming; on the other hand, computing the maximum value of polynomials and rational functions is rather difficult if their degree is very high or the dimension of the parameter space is too large. So a linear function is a good alternative to compute the maximum value of f with PAC guarantee, while polynomials are suitable for analyzing more complicated properties, such as the global ones considered below.

Polynomial PAC Approximation. One advantage of polynomials over rational functions is that they make it easy to compute complex operations such as inner product and integral [55], as needed to evaluate e.g. the L_p norm $\|g\|_p = \sqrt[p]{\int_Z |g(z)|^p dz}$ of a function $g: Z \rightarrow \mathbb{R}$, with $p \geq 1$. This means that we can adopt polynomials to check some more complicated properties of a pDTMRM \mathcal{R}_V , such as whether the function f_φ is close to a given number β on the whole parameter space X . This is useful, for instance, to evaluate how much the behavior of \mathcal{R}_V with respect to the property φ is affected by the variations of the parameters. We can model this situation as follows:

Definition 8. *Given the domain X of a set of parameters, a function $f: X \rightarrow \mathbb{R}_{\geq 0}$, a safety level ζ , and $\beta \in \mathbb{R}_{\geq 0}$, we say that f is near β within the safety level ζ on X with respect to the L_p norm, if $\|f - \beta\|_p < \zeta$.*

To verify the above property, we can rely on the following result:

Lemma 3. *Given X , f , ζ , and β as in Definition 8, let M be an upper bound of $f(X)$ and \tilde{f} be a PAC approximation of f with (ε, η) -guarantee and margin λ ; let $|X| = \int_X 1 d\mathbf{v}$. For each $p \geq 1$, if \tilde{f} satisfies the condition*

$$\sqrt[p]{\left(\lambda \sqrt{(1-\varepsilon) \cdot |X|} + \|\tilde{f} - \beta\|_p\right)^p + \varepsilon \cdot |X| \cdot \max(|M - \beta|^p, \beta^p)} < \zeta \quad (6)$$

then $\|f - \beta\|_p < \zeta$ holds with confidence $1 - \eta$.

Consider again the pDTMC \mathcal{D}_V shown in Fig. 2 and $\varphi = \mathbf{P}_{=?}(\mathbf{F}\checkmark)$; since f_φ represents probabilities, we have the well-known upper bound $M = 1$. Here we consider the L_2 norm, which is widely used in describing the error between functions in the signal processing field (see, e.g., [12, 21]), as it can reflect the global approximation properties and is easy to compute. To simplify the notation, let UB denote the complex expression occurring in the formula (6), that is:

$$UB(\tilde{f}_\varphi, X, \beta) = \sqrt{\left(\lambda \sqrt{(1-\varepsilon) \cdot |X|} + \|\tilde{f}_\varphi - \beta\|_2\right)^2 + \varepsilon \cdot |X| \cdot \max(|1 - \beta|^2, \beta^2)}.$$

We want to know whether $f_\varphi(p, q) = \frac{q^2}{q+2p-2pq}$ is near 0.5 within 0.05, i.e., given the safety level $\zeta = 0.05$, we want to check $\|f_\varphi - 0.5\|_2 < 0.05$. According to Lemma 3, we first compute a PAC approximation \tilde{f}_φ of f_φ . By setting $\varepsilon = \eta = 0.05$, we get the quadratic polynomial $\tilde{f}_\varphi(p, q) = 0.013 + 0.925q - 1.442p + 0.953pq + 2.072p^2 + 0.085q^2$, by rounding to three decimals. In this case, we get $UB(\tilde{f}_\varphi, X, \beta) = 0.0432 < \zeta = 0.05$, so Lemma 3 applies. If, instead, we would have chosen $\zeta' = 0.04$, then we cannot prove $\|f_\varphi - 0.5\|_2 < 0.04$ by relying on Lemma 3. To do so, we need to consider the more conservative values $\varepsilon = 0.01$ and $\eta = 0.001$, which give us $UB(\tilde{f}_\varphi, X, \beta) = 0.0379 < \zeta' = 0.04$, so we can derive that $\|f_\varphi - 0.5\|_2 < 0.04$ holds with confidence 99.9%.

Extension to Reward Models. The extension of the constructions given above to reward properties is rather easy: for instance, we can approximate the rational function representing the state property $\varphi = \mathbf{R}_{=?}(\mathbf{F}\varphi')$, the reward counterpart of $\mathbf{P}_{=?}(\psi')$, by instantiating $f_\varphi(\mathbf{v}_i)$ in Problem (4) with the expected reward value computed on the pDTMC instantiated with \mathbf{v}_i . Similarly, we can compute linear and polynomial PAC approximations for safe regions, with the latter defined in terms of the value of the reward instead of the probability.

We can consider also the following case: given a pDTMRM \mathcal{R}_V , we want to verify whether the expected value of $\varphi = \mathbf{R}_{=?}(\mathbf{F}\varphi')$ over the parameters \mathbf{v} , denoted $f_\varphi(\mathbf{v})$, can reach a given reward level ρ . This model the scenarios where, to make a decision, we need to know whether the expectation of the rewards for a certain decision satisfies the given conditions. We formalize this case as follows:

Definition 9. *Given the domain X of a set of parameters, a function $f: X \rightarrow \mathbb{R}_{\geq 0}$, a reward level ρ , and a probability measure P over X , we say that the expectation of f on X with respect to P can reach the reward level ρ , if*

$$\int_X f(\mathbf{v}) dP(\mathbf{v}) > \rho. \quad (7)$$

We can resort to the following lemma to check condition (7):

Lemma 4. *Given X , f , P , and ρ as in Definition 9, let \tilde{f} be a PAC approximation of f with (ε, η) -guarantee and margin λ . If \tilde{f} satisfies the condition*

$$\int_X (\tilde{f}(\mathbf{v}) - \lambda) dP(\mathbf{v}) - \varepsilon \cdot |X| \cdot \max_{\mathbf{v} \in X} (\tilde{f}(\mathbf{v}) - \lambda) > \rho, \quad (8)$$

then Condition (7) holds with confidence $1 - \eta$.

4 Experimental Evaluation

We have implemented the PAC-based analysis approach proposed in Sect. 3 in a prototype tool named PACPMA¹, the PAC-based Parametric Model Analyzer, and evaluated it on several benchmarks: we considered the DTMCs from the PRISM benchmark suite [45], some of the resulting models are also available as examples in the STORM [36] repository². We replaced the probabilistic choices in them with parameters. The probabilistic choices in most of the models correspond to the flip of a fair coin, so we considered three possible ranges for the parameters, namely [0.01, 0.33], [0.33, 0.66], and [0.66, 0.99], to represent the fact that the coin is strongly unfair to head, rather fair, and strongly unfair to tail, respectively. For the remaining models, where the choice is managed by the uniform distribution over several outcomes, we split the outcomes into two groups (e.g., odd and even outcomes) and then used a parametric coin and five intervals to choose the group. By considering the reachability properties available for each DTMC and the choice of the constants controlling the size of the DTMCs, we get a total of 936 benchmarks for our evaluation for probabilistic properties and 620 benchmarks for expected rewards. We performed our experiments on a desktop machine with an i7-4790 CPU and 16 GB of memory running Ubuntu Server 20.04.4; we used BENCHEXEC [9] to trace and constrain the tools' executions: we allowed each benchmark to use 15 GB of memory and imposed a time limit of 10 min of wall-clock time.

PACPMA is written in JAVA and uses STORM [36] and MATLAB to get the value of the analyzed property and the solution of the LP problem, respectively. We also used STORM v1.7.0 and PRISM [44] v4.7 to compute the actual rational functions for the benchmarks, to check how well our PAC approximation works in practice. We were unable to compare with the fraction-free approach proposed in [5] since it is implemented as an extension of STORM v1.2.1 that fails to build on our system. To avoid to call repeatedly STORM for each sample as an external process, we wrote a C wrapper for STORM that parses the input model and formula and sets the model constants only once, and then repeatedly instantiates the obtained parametric model with the samples and computes the corresponding values of the property, similarly to the batch mode used in [3]. We

¹ <https://github.com/iscas-tis/PacPMA/>.

² <https://github.com/moves-rwth/storm/>.

Table 1. Overview of the outcomes of the experiments

	Outcome	PRISM	STORM	PACPMA ^d (1 thread/8 threads)				
				d = 1	d = 2	d = 3	d = 4	d = 5
P _{=?} [ψ]	Success	522	576	594/629	585/621	576/621	576/621	576/603
	Memoryout	18	63	0/306	0/306	0/306	0/306	0/306
	Timeout	396	297	342/1	351/9	360/9	360/9	360/27
R _{=?} [ψ]	Success	153	224	302/302	302/302	302/302	302/302	302/302
	Memoryout	0	0	0/282	0/282	0/282	0/282	0/282
	Timeout	467	396	318/36	318/36	318/36	318/36	318/36

also implemented a multi-threaded evaluation of the sampled points, by calling multiple instances of the wrapper in parallel on a partition of the samples.

4.1 Overall Evaluation

In Table 1 we show the outcome of the different tools on the 936 probabilistic (marked with P_{=?}[ψ]) and 620 reward (marked with R_{=?}[ψ]) benchmarks, namely whether they successfully produced a rational function or whether they failed by timeout or by running out of memory. Besides the results for PRISM and STORM computing the actual rational function, we report two values for each outcome of PACPMA^d, where the superscript d indicates the degree of the polynomial used as template: in e.g. the pair 594/629, the first value 594 is relative to the single-threaded PACPMA¹, while the value 629 is for the 8-threaded PACPMA¹, i.e., PACPMA with 8 instances of the STORM wrapper running in parallel. As parameters for PACPMA, we set $\varepsilon = \eta = 0.05$; for the benchmarks with two parameters, this results in sampling between 280 and 1000 points, for $d = 1$ to $d = 5$, respectively. To make the comparison between the different templates fairer, we set the same random seed for each run of PACPMA; this ensures that all samples used by e.g. PACPMA² are also used by PACPMA⁵.

As we can see from Table 1, PACPMA is able to compute polynomials with different degrees for more benchmarks than STORM and PRISM. By inspecting the single experiments, for the probabilistic properties we have that $\text{PRISM} \subseteq \text{STORM} \subseteq \text{PACPMA}_n^d \subseteq \text{PACPMA}_n^{d'}$ for each $d' < d$ degrees and n threads, as sets of successfully solved cases; we also have that $\text{PACPMA}_1^d \subseteq \text{PACPMA}_8^d$ for each d . For the reward properties we have that $\text{PACPMA}_n^d = \text{PACPMA}_{n'}^{d'}$ for each combination of $d, d' \in \{1, \dots, 5\}$ and $n, n' \in \{1, 8\}$ and that $\text{STORM}, \text{PRISM} \subseteq \text{PACPMA}_n^d$; however STORM and PRISM are incomparable, with cases solved by STORM but not by PRISM, and vice-versa. In the next section we will evaluate how the margin λ changes depending on the degree d and the statistical parameters ε and η through the induced number of samples.

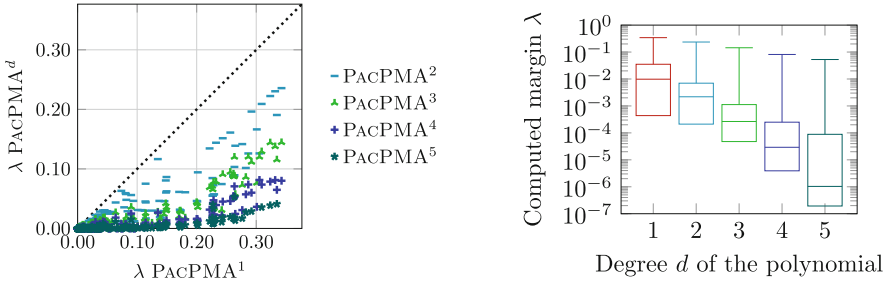


Fig. 4. Scatter plot for the margin λ for different PACPMA^d and box plots for the margin λ

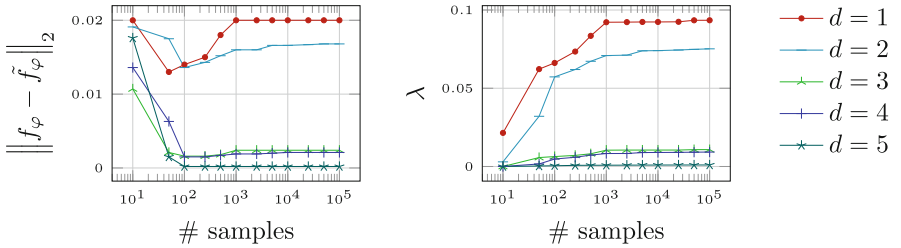


Fig. 5. Value of $\|f_\varphi - \tilde{f}_\varphi\|_2$ and of λ vs. degree of polynomials and number of samples

4.2 Relation of the Polynomial Degree d and the Number of Samples with the Margin λ and the Distance $\|f_\varphi - \tilde{f}_\varphi\|_2$

In Fig. 4 we present plots for PACPMA using polynomial templates with different degrees and how the computed λ changes. As we can see from the plots, by using a higher degree we get a lower value for the margin λ , as one would expect given that polynomials with higher degree can approximate better the shape of the actual rational function: from the box plots on the right side of the figure, we can see that using higher degree polynomials allows us to get values for λ that are much closer to 0. Note that in these box plots we removed the lower whiskers since they are 0 for all degrees, and we use a logarithmic y-axis. The scatter plot shown on the left side of Fig. 4, where we compare the values of λ produced by PACPMA¹ with those by PACPMA^d, for $d = 2, 3, 4, 5$, confirms that the higher the degree is, the closer to 0 the corresponding mark is, since the points for the same benchmark share the same x-axis value.

In Fig. 5 we show the value of $\|f_\varphi - \tilde{f}_\varphi\|_2$ for different degrees of the polynomial and the number of samples, as well as the corresponding values of the computed λ . The plots are relative to one benchmark such that the corresponding rational function (a polynomial having degree 96) computed by STORM can be managed by MATLAB without incurring obvious numerical errors while having the margin λ computed by PACPMA² reasonably large ($\lambda \approx 0.063$).

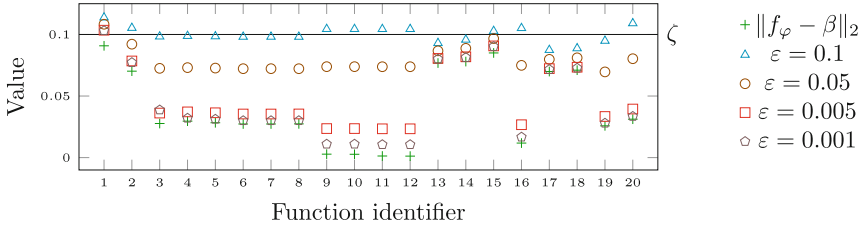


Fig. 6. Comparison of $\|f_\varphi - \beta\|_2$ with $UB(\tilde{f}_\varphi, X, \beta)$ for $\eta = 0.05$ and different ε

From the plots, we can see that we need at least 100 samples to get a rather stable value for $\|f_\varphi - \tilde{f}_\varphi\|_2$, so that the value of $\|f_\varphi - \tilde{f}_\varphi\|_2$ is smaller for higher degrees, which reflects the more accurate polynomial approximation of the original function, in line with the plots in Fig. 4. However, for the same degree, increasing the number of samples does not always lead to a decrease in $\|f_\varphi - \tilde{f}_\varphi\|_2$ value. This happens because with few points, the polynomial can fit them well, as indicated by the low value of λ ; however, such few points are likely to be not enough to represent accurately the shape of f_φ . By increasing the number of samples, the shape of f_φ can be known better, in particular where it changes more; this makes it more difficult for the polynomials to approximate f_φ , as indicated by the larger λ ; on the other hand, they get closer to f_φ , so $\|f_\varphi - \tilde{f}_\varphi\|_2$ stabilizes.

4.3 Relation of the Statistical Parameters ε and η with the Distances $\|f_\varphi - \beta\|_2$ and $UB(\tilde{f}_\varphi, X, \beta)$

We now consider the behavior of f_φ and whether it remains close to some number β within ζ , that is, we want to check whether $\|f_\varphi - \beta\|_2 < \zeta$ holds. Here we set the safety level ζ to be 0.1 and consider different β 's values for different functions f_φ . We consider 20 rational functions computed by STORM that MATLAB can work without incurring in obvious numerical errors, such as those outside the probability interval $[0, 1]$. For each of the function, we computed the corresponding value of β by sampling 20 points for the parameters and taking the average value, rounded to the first decimal, of the function on them. We rely on Lemma 3 to perform the analysis; the results are shown in Fig. 6.

In the figure, we plot the actual value of $\|f_\varphi - \beta\|_2$, the boundary ζ , and the value of $UB(\tilde{f}_\varphi, X, \beta)$ computed with respect to $\eta = 0.05$ and different choices of ε for the 20 functions. As we can see, the smaller ε , the higher the number of cases on which Lemma 3 ensures $\|f_\varphi - \beta\|_2 < \zeta$; this is expected, since a smaller ε increases the number of samples, so the approximating polynomial \tilde{f}_φ gets closer to the real shape of f_φ . Moreover, when $\|f_\varphi - \beta\|_2$ is already close to ζ , there is little space for \tilde{f}_φ to differ from f_φ , as happens for the e.g. the function 1. Thus it is more difficult for us to be able to rely on Lemma 3 to check whether $\|f_\varphi - \beta\|_2 < \zeta$ holds, even if this actually the case.

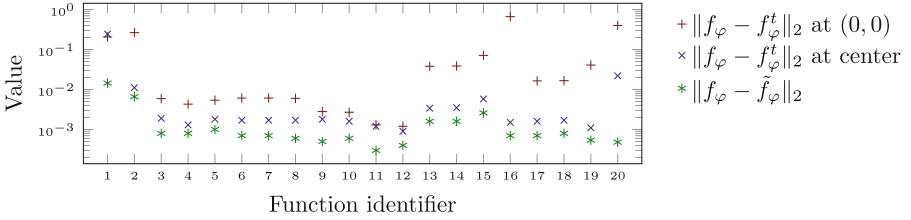


Fig. 7. Distance from f_φ of the Taylor expansion vs. the approximating polynomial

4.4 Comparison with the Taylor Expansion

We compare the accuracy of PAC approximation against that of the Taylor expansion on the same cases used for Fig. 6; the comparison is shown in Fig. 7. For the comparison with f_φ , we consider the degree 2 for both the Taylor expansion f_φ^t and the approximating polynomial \tilde{f}_φ computed with $\varepsilon = \eta = 0.05$. For the Taylor expansion f_φ^t , we considered two versions: the expansion at the origin, i.e., $(0, 0)$ for two parameters (marked as “ $\|f_\varphi - f_\varphi^t\|_2$ at $(0, 0)$ ” in Fig. 7), that is commonly used since it is cheaper to compute than the expansions at other points; and the expansion at the barycenter of the space of the parameters (marked as “ $\|f_\varphi - f_\varphi^t\|_2$ at center” in Fig. 7).

As we can see from the plot, that uses a logarithmic scale on the y-axis, the distance $\|f_\varphi - \tilde{f}_\varphi\|_2$ is between one and three orders of magnitude smaller than $\|f_\varphi - f_\varphi^t\|_2$ at the origin. If we consider $\|f_\varphi - f_\varphi^t\|_2$ at the barycenter, we get values much closer to $\|f_\varphi - \tilde{f}_\varphi\|_2$, but still larger up to one order of magnitude. One of the reasons for this is that the Taylor expansion reflects local properties of f_φ at the expansion point, while the PAC approximation provides a global approximation of f_φ , thus reducing the overall distance. Compared with the Taylor expansion, the PAC approximation has also other advantages: the PAC approximation can handle both white-box and black-box problems, i.e., we do not need to get the analytical form of f_φ ; this means that we can treat it as a black box and get a good approximation of it while the Taylor expansion can only be applied after computing the actual function f_φ . Moreover, the PAC approximation is able to generate polynomials with any given error rate and provide probabilistic guarantee, while Taylor expansion cannot.

4.5 Extension to Reward Models

In Fig. 8 we show how Eq. (8) applies to $\int_X f_\varphi(\mathbf{v}) dP(\mathbf{v})$ for a selection of 30 reward properties f_φ computed by STORM; as usual, we compute \tilde{f}_φ with $\varepsilon = \eta = 0.05$. In the figure, we report the actual value of $\int_X f_\varphi(\mathbf{v}) dP(\mathbf{v})$ as well as that of the expression in Eq. (8) computed for the polynomial PAC approximations \tilde{f}_φ at different degrees. As we can see from Fig. 8, the higher the degree of \tilde{f}_φ , the more accurate the estimation of the $\int_X f_\varphi(\mathbf{v}) dP(\mathbf{v})$'s lower bound is. In particular, the quadratic \tilde{f}_φ provides a very close lower bound for $\int_X f_\varphi(\mathbf{v}) dP(\mathbf{v})$; this

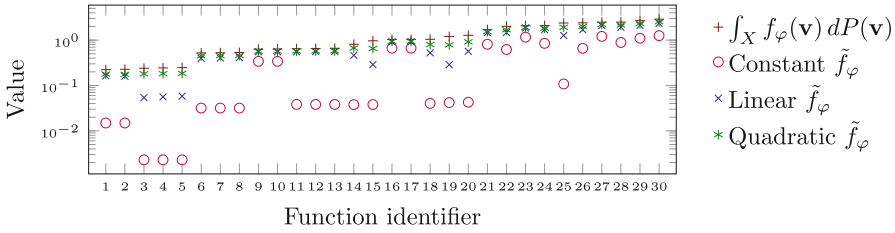


Fig. 8. Lower bound for Eq. (7) by PAC approximation with different degrees

is remarkable, since evaluating $\max(\tilde{f}(\mathbf{v}) - \lambda)$ in Eq. (8) is often an NP-hard non-convex optimization problem [51,56] and, for cubic or higher polynomials, it requires specialized theories and tools to solve [46,47,61].

5 Conclusion

In this paper, we presented a PAC-based approximation framework for studying several properties of parametric discrete time Markov chains. Within the framework, we can analyze the safety regions of the domain of the parameters, check whether the actual probability fluctuates around a reference value within a certain bound, and get a polynomial approximating the actual probability rational function with given (ϵ, η) -PAC guarantee. An extended experimental evaluation confirmed the efficacy of our framework in analyzing parametric models.

As future work, we plan to investigate the applicability of the scenario approach to other Markov models and properties, such as continuous time Markov chains and Markov decision processes with and without rewards, where parameters can also control the rewards structures. Moreover, we plan to explore the combination of the scenario approach with statistical model checking and black-box verification and model learning.

Acknowledgements. We thank the anonymous reviewers for their useful remarks that helped us improve the quality of the paper. Work supported in part by the CAS Project for Young Scientists in Basic Research under grant No. YSBR-040, NSFC under grant No. 61836005, the CAS Pioneer Hundred Talents Program, the ISCAS New Cultivation Project ISCAS-PYFX-202201, and the ERC Consolidator Grant 864075 (CAESAR).

This project is part of the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant no. 101008233.

Data Availability Statement. An environment with the tools and data used for the experimental evaluation presented in this work is available in the following Zenodo repository: <https://doi.org/10.5281/zenodo.8181117>.

References

1. Andova, S., Hermanns, H., Katoen, J.-P.: Discrete-time rewards model-checked. In: Larsen, K.G., Niebert, P. (eds.) FORMATS 2003. LNCS, vol. 2791, pp. 88–104. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-40903-8_8
2. Badings, T.S., Cubuktepe, M., Jansen, N., Junges, S., Katoen, J., Topcu, U.: Scenario-based verification of uncertain parametric MDPs. *Int. J. Softw. Tools Technol. Transf.* **24**(5), 803–819 (2022). <https://doi.org/10.1007/s10009-022-00673-z>
3. Badings, T.S., Jansen, N., Junges, S., Stoelinga, M., Volk, M.: Sampling-based verification of CTMCs with uncertain rates. In: Shoham, S., Vizel, Y. (eds.) CAV 2022 Part II. LNCS, vol. 13372, pp. 26–47. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-13188-2_2
4. Bai, H., Cai, S., Ye, N., Hsu, D., Lee, W.S.: Intention-aware online POMDP planning for autonomous driving in a crowd. In: IEEE International Conference on Robotics and Automation, ICRA 2015, Seattle, WA, USA, 26–30 May, 2015, pp. 454–460. IEEE (2015). <https://doi.org/10.1109/ICRA.2015.7139219>
5. Baier, C., Hensel, C., Hutschenreiter, L., Junges, S., Katoen, J., Klein, J.: Parametric Markov chains: PCTL complexity and fraction-free Gaussian elimination. *Inf. Comput.* **272**, 104504 (2020). <https://doi.org/10.1016/j.ic.2019.104504>
6. Baier, C., Katoen, J.: Principles of Model Checking. MIT Press, Cambridge (2008)
7. Ben-Tal, A., Nemirovski, A.: Robust convex optimization. *Math. Oper. Res.* **23**(4), 769–805 (1998). <https://doi.org/10.1287/moor.23.4.769>
8. Ben-Tal, A., Nemirovski, A.: Robust solutions of uncertain linear programs. *Oper. Res. Lett.* **25**(1), 1–13 (1999). [https://doi.org/10.1016/S0167-6377\(99\)00016-4](https://doi.org/10.1016/S0167-6377(99)00016-4)
9. Beyer, D., Löwe, S., Wendler, P.: Reliable benchmarking: requirements and solutions. *Int. J. Softw. Tools Technol. Transfer* **21**(1), 1–29 (2017). <https://doi.org/10.1007/s10009-017-0469-y>
10. Bianco, A., de Alfaro, L.: Model checking of probabilistic and nondeterministic systems. In: Thiagarajan, P.S. (ed.) FSTTCS 1995. LNCS, vol. 1026, pp. 499–513. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60692-0_70
11. Blackwell, D.: On the functional equation of dynamic programming. *J. Math. Anal. Appl.* **2**(2), 273–276 (1961)
12. Boggess, A., Narcowich, F.J.: A First Course in Wavelets with Fourier Analysis. Wiley, Hoboken (2015)
13. Bortolussi, L., Milios, D., Sanguinetti, G.: Smoothed model checking for uncertain continuous-time Markov chains. *Inf. Comput.* **247**, 235–253 (2016). <https://doi.org/10.1016/j.ic.2016.01.004>
14. Bortolussi, L., Silveti, S.: Bayesian statistical parameter synthesis for linear temporal properties of stochastic models. In: Beyer, D., Huisman, M. (eds.) TACAS 2018. LNCS, vol. 10806, pp. 396–413. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-89963-3_23
15. Calafiore, G.C., Campi, M.C.: Uncertain convex programs: randomized solutions and confidence levels. *Math. Program.* **102**(1), 25–46 (2005). <https://doi.org/10.1007/s10107-003-0499-y>
16. Calafiore, G.C., Campi, M.C.: The scenario approach to robust control design. *IEEE Trans. Autom. Control.* **51**(5), 742–753 (2006). <https://doi.org/10.1109/TAC.2006.875041>
17. Campi, M.C., Garatti, S.: A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *J. Optim. Theory Appl.* **148**(2), 257–280 (2011). <https://doi.org/10.1007/s10957-010-9754-6>

18. Campi, M.C., Garatti, S., Prandini, M.: The scenario approach for systems and control design. *Annu. Rev. Control.* **33**(2), 149–157 (2009). <https://doi.org/10.1016/j.arcontrol.2009.07.001>
19. Carè, A., Garatti, S., Campi, M.C.: Scenario min-max optimization and the risk of empirical costs. *SIAM J. Optim.* **25**(4), 2061–2080 (2015). <https://doi.org/10.1137/130928546>
20. Clarke, E.M., Henzinger, T.A., Veith, H., Bloem, R. (eds.): *Handbook of Model Checking*. Springer, Cham (2018). <https://doi.org/10.1007/978-3-319-10575-8>
21. Conway, J.B.: *A Course in Functional Analysis*, vol. 96. Springer, Cham (2019)
22. 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). https://doi.org/10.1007/978-3-540-31862-0_21
23. Dehnert, C., et al.: **PROPhESY**: A PRObabilistic ParamETER SYnthesis Tool. In: Kroening, D., Păsăreanu, C.S. (eds.) *CAV 2015*. LNCS, vol. 9206, pp. 214–231. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21690-4_13
24. Dehnert, C., Junges, S., Katoen, J., Volk, M.: A storm is coming: A modern probabilistic model checker. In: Majumdar, R., Kuncak, V. (eds.) *Computer Aided Verification. CAV 2017*. LNCS vol. 10427, pp. 592–600. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63390-9_31
25. Dubins, L.E., Savage, L.: *How to Gamble If You Must*. McGraw-Hill (1965)
26. Fang, X., Calinescu, R., Gerasimou, S., Alhwickem, F.: Fast parametric model checking through model fragmentation. In: *43rd IEEE/ACM International Conference on Software Engineering, ICSE 2021, Madrid, Spain, 22–30 May 2021*, pp. 835–846. IEEE (2021). <https://doi.org/10.1109/ICSE43902.2021.00081>
27. Gainer, P., Hahn, E.M., Schewe, S.: Accelerated model checking of parametric Markov chains. In: Lahiri, S.K., Wang, C. (eds.) *ATVA 2018*. LNCS, vol. 11138, pp. 300–316. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01090-4_18
28. Garavel, H., Lang, F., Mateescu, R., Serwe, W.: CADP 2011: a toolbox for the construction and analysis of distributed processes. *Int. J. Softw. Tools Technol. Transf.* **15**(2), 89–107 (2013). <https://doi.org/10.1007/s10009-012-0244-z>
29. Ghaoui, L.E., Oustry, F., Lebret, H.: Robust solutions to uncertain semidefinite programs. *SIAM J. Optim.* **9**(1), 33–52 (1998). <https://doi.org/10.1137/S1052623496305717>
30. 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). https://doi.org/10.1007/978-3-642-20398-5_12
31. Hahn, E.M., Hermanns, H., Wachter, B., Zhang, L.: **PARAM**: a model checker for parametric Markov models. In: Touili, T., Cook, B., Jackson, P. (eds.) *CAV 2010*. LNCS, vol. 6174, pp. 660–664. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14295-6_56
32. Hahn, E.M., Hermanns, H., Zhang, L.: Probabilistic reachability for parametric Markov models. *Int. J. Softw. Tools Technol. Transf.* **13**(1), 3–19 (2011). <https://doi.org/10.1007/s10009-010-0146-x>
33. Hahn, E.M., Li, Y., Schewe, S., Turrini, A., Zhang, L.: **ISCASMC**: a web-based probabilistic model checker. In: Jones, C., Pihlajasaari, P., Sun, J. (eds.) *FM 2014*. LNCS, vol. 8442, pp. 312–317. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-06410-9_22
34. Han, T.: Diagnosis, synthesis and analysis of probabilistic models. In: Bernstein, A., et al. (eds.) *Ausgezeichnete Informatikdissertationen 2009, LNI*, vol. D-10, pp. 81–90. GI (2009). <https://dl.gi.de/handle/20.500.12116/33657>

35. Hansson, H., Jonsson, B.: A logic for reasoning about time and reliability. *Formal Aspects Comput.* **6**(5), 512–535 (1994). <https://doi.org/10.1007/BF01211866>
36. Hensel, C., Junges, S., Katoen, J., Quatmann, T., Volk, M.: The probabilistic model checker storm. *Int. J. Softw. Tools Technol. Transf.* **24**(4), 589–610 (2022). <https://doi.org/10.1007/s10009-021-00633-z>
37. von Hilgers, P., Langville, A.N.: The five greatest applications of Markov chains. In: *Proceedings of the Markov Anniversary Meeting*, pp. 155–168 (2006)
38. Hopcroft, J.E., Motwani, R., Ullman, J.D.: *Introduction to Automata Theory, Languages, and Computation*, 3rd edn. Addison-Wesley, Pearson international edition (2007)
39. Jansen, N., et al.: Accelerating parametric probabilistic verification. In: Norman, G., Sanders, W. (eds.) *QEST 2014*. LNCS, vol. 8657, pp. 404–420. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10696-0_31
40. Jansen, N., Junges, S., Katoen, J.: Parameter synthesis in Markov models: a gentle survey. In: Raskin, J., Chatterjee, K., Doyen, L., Majumdar, R. (eds.) *Principles of Systems Design*. LNCS, vol. 13660, pp. 407–437. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22337-2_20
41. Junges, S., Katoen, J., Pérez, G.A., Winkler, T.: The complexity of reachability in parametric Markov decision processes. *J. Comput. Syst. Sci.* **119**, 183–210 (2021). <https://doi.org/10.1016/j.jcss.2021.02.006>
42. Katoen, J., Zapreev, I.S., Hahn, E.M., Hermanns, H., Jansen, D.N.: The ins and outs of the probabilistic model checker MRMC. *Perform. Evaluation* **68**(2), 90–104 (2011). <https://doi.org/10.1016/j.peva.2010.04.001>
43. Kwiatkowska, M., Norman, G., Parker, D.: Stochastic model checking. In: Bernardo, M., Hillston, J. (eds.) *SFM 2007*. LNCS, vol. 4486, pp. 220–270. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-72522-0_6
44. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) *CAV 2011*. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_47
45. Kwiatkowska, M.Z., Norman, G., Parker, D.: The PRISM benchmark suite. In: *Ninth International Conference on Quantitative Evaluation of Systems, QEST 2012, London, United Kingdom, September 17–20, 2012*, pp. 203–204. IEEE Computer Society (2012). <https://doi.org/10.1109/QEST.2012.14>
46. Lasserre, J.B.: A semidefinite programming approach to the generalized problem of moments. *Math. Program.* **112**(1), 65–92 (2008). <https://doi.org/10.1007/s10107-006-0085-1>
47. Lasserre, J.B.: *Moments, Positive Polynomials and their Applications*, vol. 1. World Scientific, Singapore (2009)
48. Li, R., Yang, P., Huang, C., Sun, Y., Xue, B., Zhang, L.: Towards practical robustness analysis for DNNs based on PAC-model learning. In: *44th IEEE/ACM 44th International Conference on Software Engineering, ICSE 2022, Pittsburgh, PA, USA, May 25–27, 2022*, pp. 2189–2201. ACM (2022). <https://doi.org/10.1145/3510003.3510143>
49. Liu, Y., Turrini, A., Hahn, E.M., Xue, B., Zhang, L.: Scenario approach for parametric Markov models. *CoRR abs/2304.08330* (2023). <https://doi.org/10.48550/arXiv.2304.08330>
50. Margellos, K., Goulart, P., Lygeros, J.: On the road between robust optimization and the scenario approach for chance constrained optimization problems. *IEEE Trans. Autom. Control.* **59**(8), 2258–2263 (2014). <https://doi.org/10.1109/TAC.2014.2303232>

51. Pardalos, P.M., Ye, Y., Han, C.G.: Algorithms for the solution of quadratic knapsack problems. *Linear Algebra Appl.* **152**, 69–91 (1991)
52. Puterman, M.L.: *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics, Wiley, Hoboken (1994). <https://doi.org/10.1002/9780470316887>
53. Quatmann, T., Dehnert, C., Jansen, N., Junges, S., Katoen, J.-P.: Parameter synthesis for markov models: faster than ever. In: Artho, C., Legay, A., Peled, D. (eds.) *ATVA 2016*. LNCS, vol. 9938, pp. 50–67. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46520-3_4
54. Rockafellar, R.T.: *Convex Analysis*, vol. 11. Princeton University Press, Princeton (1997)
55. Rudin, W.: *Principles of Mathematical Analysis*, vol. 3. McGraw-Hill, New York (1976)
56. Sahni, S.: Computationally related problems. *SIAM J. Comput.* **3**(4), 262–279 (1974). <https://doi.org/10.1137/0203021>
57. Spel, J., Junges, S., Katoen, J.-P.: Finding provably optimal Markov chains. In: *TACAS 2021*. LNCS, vol. 12651, pp. 173–190. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-72016-2_10
58. Vayanos, P., Kuhn, D., Rustem, B.: A constraint sampling approach for multi-stage robust optimization. *Automatica* **48**(3), 459–471 (2012). <https://doi.org/10.1016/j.automatica.2011.12.002>
59. Watkins, C.J.C.H., Dayan, P.: Q-learning. *Mach. Learn.* **8**, 279–292 (1992). <https://doi.org/10.1007/BF00992698>
60. Xue, B., Zhang, M., Easwaran, A., Li, Q.: PAC model checking of black-box continuous-time dynamical systems. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **39**(11), 3944–3955 (2020). <https://doi.org/10.1109/TCAD.2020.3012251>
61. Yang, J., Ye, K., Zhi, L.: Computing sparse Fourier sum of squares on finite Abelian groups in quasi-linear time. *CoRR abs/2201.03912* (2022)