Fine-Tuning the Odds in Bayesian Networks

Bahare Salmani[(✉)] and Joost-Pieter Katoen[(✉)]

RWTH Aachen University, Aachen, Germany
`{salmani,katoen}@cs.rwth-aachen.de`

Abstract. This paper proposes new analysis techniques for Bayes networks in which conditional probability tables (CPTs) may contain symbolic variables. The key idea is to exploit scalable and powerful techniques for synthesis problems in parametric Markov chains. Our techniques are applicable to arbitrarily many, possibly dependent, parameters that may occur in multiple CPTs. This lifts the severe restrictions on parameters, e.g., by restricting the number of parametrized CPTs to one or two, or by avoiding parameter dependencies between several CPTs, in existing works for parametric Bayes networks (pBNs). We describe how our techniques can be used for various pBN synthesis problems studied in the literature such as computing sensitivity functions (and values), simple and difference parameter tuning, ratio parameter tuning, and minimal change tuning. Experiments on several benchmarks show that our prototypical tool built on top of the probabilistic model checker Storm can handle several hundreds of parameters.

1 Introduction

Parametric Bayesian Networks. We consider Bayesian networks (BNs) whose conditional probability tables (CPTs) contain symbolic parameters such as x_1, $2 \cdot x_1^2$, and $x_1 + x_2$ with $0 < x_1, x_2 < 1$. Parametric probabilistic graphical models received a lot of attention, see e.g., [7,9–11,13,14,17–19,26,34,37,40,42]. Sensitivity analysis determines the effect of the parameter values in the CPTs on the decisions drawn from the parametric BN (pBN), e.g., whether $\Pr(H{=}h \mid E{=}e) > q$ for a given $q \in \mathbb{Q} \cap [0,1]$. It amounts to establishing a function expressing an output probability in terms of the x_i parameters under study. Parameter synthesis on pBNs deals with instantiating or altering the parameters such that the resulting BN satisfies some constraint of interest. For pBNs, synthesis mostly amounts to parameter tuning: find the minimal change on the parameters such that some constraint, e.g., $\Pr(H{=}h \mid E{=}e) > q$ holds [15,40]. As sensitivity analysis and parameter synthesis are computationally hard in general [37,38,40], many techniques restrict the number of parameters per CPT (n-way for small n [13,17,29]), permit parameter dependencies in several CPTs (single CPT [14]), or consider specific structures such as join trees [37] and require all parameters to occur in the same clique of the junction tree.

Parametric Markov Chains. Quite independently, analysis techniques for Markov chains (MCs) have been developed in the formal verification community in the last two decades [21,23,24,27,30,31,39,41]; for a recent overview see [35]. Parametric MCs (pMCs) are MCs in which transitions are labelled with multivariate polynomials over a fixed set of parameters. Substitution of these variables by concrete values induces a probability distribution over the state space of the MC. Whereas early works focused on computing a rational function over the parameters expressing the reachability probability of a given target state, in the last years significant progress has been made to check whether there exists a parameter valuation inducing a MC that satisfies a given objective, or to partition the parameter space—the space of all possible parameter values—into "good" and "bad" w.r.t. a given objective, e.g., is the probability to reach some states below (or above) a given threshold q? The complexity of various pMC synthesis problems is studied in [1,36].

This paper aims to extend the spectrum of parameter synthesis techniques for parametric BNs, i.e., BNs in which arbitrary many CPTs may contain symbolic probabilities, with state-of-the-art and recently developed techniques for parametric MCs. Consider the BN adapted from [22] depicted below. The probability of a cow being pregnant given both tests are negative is about 0.45. Assume the farmer wants to replace both tests such that this false-positive error is below 0.2. Figure 1 (left) indicates the corresponding pBN while (right) shows the synthesized values of the parameters p and q (the false-negative probabilities for the new urine and blood tests) using pMC techniques [41] such that the farmer's constraint is satisfied (green) or not (red).
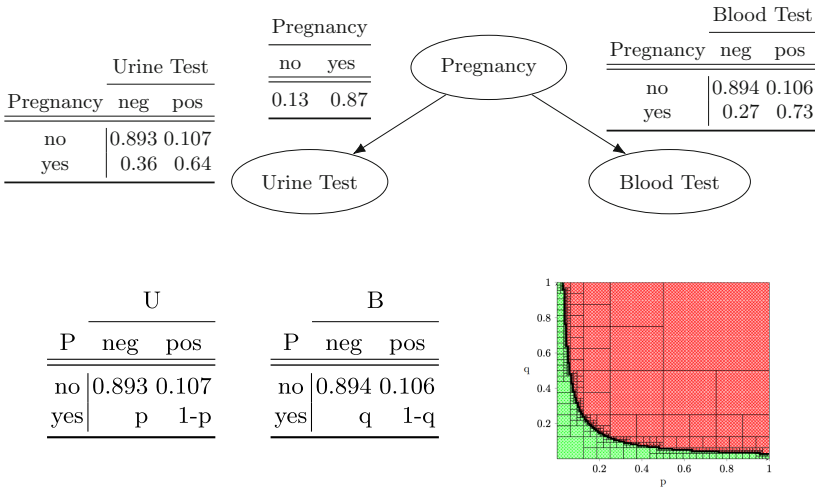

Urine Test

Pregnancy	neg	pos
no	0.893	0.107
yes	0.36	0.64

Pregnancy

no	yes
0.13	0.87

Blood Test

Pregnancy	neg	pos
no	0.894	0.106
yes	0.27	0.73

U

P	neg	pos
no	0.893	0.107
yes	p	1-p

B

P	neg	pos
no	0.894	0.106
yes	q	1-q

Fig. 1. (left) Parametric CPTs and (right) the parameter space split into safe and unsafe regions for the constraint $\Pr(P = yes \mid U = neg$ and $B = neg) \leq 0.2$. (Color figure online)

Let us highlight a few issues: we can treat parameter space partitionings that go beyond rectangular shapes, and we support multiple, possibly dependent, parameters (not illustrated in our example). Thanks to approximate techniques such as parameter lifting [41], the entire parameter space can be split into safe, unsafe, and—by some approximation factor—unknown regions. This provides useful information: if it is not possible to find urine and blood tests of a certain quality, are alternatives fine too? Parameter tuning [13] for pBNs finds parameter values that are at a minimal distance to the original values in the BN. For our example, the BN sensitivity tool SamIam suggests changing the false-negative probability of the urine test (p) from 0.36 to 0.110456; or changing the false-negative probability of the blood test (q) from 0.27 to 0.082842.[1] Interestingly, if the parameters occur in multiple CPTs, the constraint can be satisfied with a smaller deviation from the original parameter values. Other recent work [5] focuses on obtaining symbolic functions for pBN-related problems such as sensitivity analysis. Note that pBNs are similar to constrained BNs [6] that focus on logical semantics rather than synthesis algorithms and tools as we do.

We treat the theoretical foundations of exploiting pMC techniques for various synthesis questions on pBNs, present a prototypical tool that is built on top of the probabilistic model checker Storm [25] and the pMC analysis tool Prophesy [24], and provide experimental results that reveal:

- pMC techniques are competitive to most common functions with the pBN tools SamIam and Bayesserver[2].
- pMC techniques are well-applicable to general pBNs, in particular by allowing parameter dependencies.
- Approximate parameter space partitioning is effective for parameter tuning e.g., ratio, difference, and minimal change problems.

Further proofs and details of this paper can be found in [44].

2   Parametric Bayesian Networks

This section defines *parametric* Bayesian networks (BNs)[3] and defines the sensitivity analysis and parameter tuning tasks from the literature that we consider.

Parameters. Let $X = \{x_1, \ldots, x_n\}$ be a finite set of real-valued *parameters* and $\mathbb{Q}(X)$ denote the set of multivariate polynomials over X with rational coefficients. A *parameter instantiation* is a function $u \colon X \to \mathbb{R}$. A polynomial f can be interpreted as a function $f \colon \mathbb{R}^n \to \mathbb{R}$ where $f(u)$ is obtained by substitution, i.e., in $f(u)$ each occurrence of x_i in f is replaced by $u(x_i)$. To make clear where substitution occurs, we write $f[u]$ instead of $f(u)$ from now on. We assume that

---

[1] When it comes to *multiple parameter suggestions*, SamIam suggests $p = 0.120097$ or $q = 0.089892$.

[2] We do not consider tools such as [48] for sensitivity analysis of Credal networks.

[3] Our notion of parametric BN should not be confused with existing parametric notions that consider variations of the structure, i.e., the topology of the BN.

all parameters are bounded, i.e., $lb_i \leq u(x_i) \leq ub_i$ for each parameter x_i. The *parameter space* of X is the set of all possible parameter values, the hyper-rectangle spanned by the intervals $[lb_i, ub_i]$. A subset R of the parameter space is called a *region*.

Parametric Bayes Networks. A parametric BN is a BN in which entries in the conditional probability tables (CPTs) are polynomials over the parameters in X.

Definition 1. *A parametric BN (pBN) $\mathcal{B}$ is a tuple (V, E, X, Θ) with:*

- $V = \{v_1, \ldots, v_m\}$ *is a set of discrete* random variables *with* $dom(v_i) = D_{v_i}$
- $G = (V, E)$ *with* $E \subseteq V \times V$ *is a directed acyclic graph on* V
- $X = \{x_1, \ldots, x_n\}$ *is a finite set of real-valued parameters*
- Θ *is a set of parametric conditional probability tables* $\Theta = \{ \Theta_{v_i} \mid v_i \in V \}$

$$\Theta_{v_i} : \left(\prod_{p \in parents(v_i)} D_p \right) \rightarrow \mathbb{Q}(X)^{|D_{v_i}|}.$$

Let $\mathcal{B}[u]$ be obtained by replacing every parameter x_i in $\mathcal{B}$ by its value $u(x_i)$. A parameter instantiation u is *well-formed* for the pBN $\mathcal{B}$ if $\mathcal{B}[u]$ is a BN, i.e., for every $v_i \in V$ and parent evaluation $\overline{par}$, $\Theta_{v_i}(\overline{par})[u]$ yields a probability distribution over D_{v_i}. In the sequel, we assume u to be well-formed. A pBN defines a *parametric* joint probability distribution over V.

pBN Subclasses. We define some sub-classes of pBNs that are used in existing sensitivity analysis techniques and tools. They constrain the number of parameters, the number of CPTs (and the number of rows in a CPT) containing parameters. Let $\mathcal{B} = (V, E, X, \Theta)$ be a pBN, $c(x_i)$ the number of CPTs in $\mathcal{B}$ in which x_i occurs and $r(x_i)$ the number of CPT rows in which x_i occurs.

- $\mathcal{B} \in p_1 c_1 r_1$ iff $\mathcal{B}$ contains one parameter x_1 and x_1 only occurs in a single row of a single CPT, i.e., $X = \{x_1\}$, $c(x_1) = r(x_1) = 1$.
- $\mathcal{B} \in p_2 c_{\leq 2} r_1$ iff $\mathcal{B}$ involves two parameters occurring only in two rows of two (or one) CPTs, i.e., $X = \{x_1, x_2\}$, $c(x_i) \in \{1, 2\}$, $r(x_i) = 1$ for $i = 1, 2$.
- $\mathcal{B} \in p_* c_1 r_1$ iff $\mathcal{B}$ allows multiple distinct parameters, provided each parameter occurs in a single row of a single CPT, i.e., $r(x_i) = 1$, $c(x_i) = 1$ for each x_i and all the parameters occur in the same CPT.

The class $p_1 c_1 r_1$ is used in one-way, $p_2 c_{\leq 2} r_1$ in two-way sensitivity analysis [13,17,29] and $p_* c_1 r_1$ in single CPT [14].

Parameter Synthesis Problems in pBN. We define some synthesis problems for pBNs by their corresponding decision problems [38]. Let Pr denote the parametric joint distribution function induced by pBN $\mathcal{B} = (V, E, X, \Theta)$ and $\Pr[u]$ the joint probability distribution of $\mathcal{B}[u]$ at well-formed instantiation u. Let $E \subseteq V$ be the evidence, $H \subseteq V$ the hypothesis and $q \in \mathbb{Q} \cap [0, 1]$ a threshold.

Parameter Tuning. Find an instantiation u s.t. $\Pr[u](H = h \mid E = e) \geq q$.

Hypothesis Ratio Parameter Tuning. Find an instantiation u s.t.

$$\frac{\Pr[u](H = h' \mid E = e)}{\Pr[u](H = h \mid E = e)} \geq q \quad \text{i.e.,} \quad \frac{\Pr[u](H = h', E = e)}{\Pr[u](H = h, E = e)} \geq q,$$

where h and h' are joint variable evaluations for the hypothesis H.

Hypothesis Difference Parameter Tuning. Find an instantiation u s.t.

$$\Pr[u](H = h \mid E = e) - \Pr[u](H = h' \mid E = e) \geq q,$$

where h and h' are joint variable evaluations for the hypothesis H.

Minimal Change Parameter Tuning. For a given parameter instantiation u_0 and $\epsilon \in \mathbb{Q}_{>0}$, find an instantiation u s.t.

$$d(\Pr[u], \Pr[u_0]) \leq \epsilon,$$

where d is a distance notion on probability distributions, see [15].

Computing Sensitivity Function and Sensitivity Value. For the evidence $E = e$ and the hypothesis $H = h$, compute the sensitivity function:

$$f_{\Pr(H=h \mid E=e)} = \Pr(H = h \mid E = e).$$

This is a rational function over X, i.e., a fraction g/h with $g, h \in \mathbb{Q}(X)$.

The difference and ratio problems can analogously be defined for evidences. The evidence tuning problems are defined for values e and e' for E, given a fixed value h for H.

3   Parametric Markov Chains

A parametric Markov chain is a Markov chain in which the transitions are labelled with polynomials over the set X of parameters. These polynomials are intended to describe a parametric probability distribution over the pMC states.

Definition 2. *A parametric Markov chain (pMC) $\mathcal{M}$ is a tuple (Σ, σ_I, X, P) where Σ is a finite set of states with initial state $\sigma_I \in \Sigma$, X is as before, and $P : \Sigma \times \Sigma \to \mathbb{Q}(X)$ is the transition probability function.*

For pMC $\mathcal{M} = (\Sigma, \sigma_I, X, P)$ and well-formed parameter instantiation u on X, $\mathcal{M}[u]$ is the discrete-time Markov chain $(\Sigma, \sigma_I, X, P[u])$ where $P[u]$ is a probability distribution over Σ. We only consider well-formed parameter instantiations. *Reachability Probabilities.* Let $\mathcal{D}$ be an MC. Let $Paths(\sigma)$ denote the set of all infinite paths in $\mathcal{D}$ starting from σ, i.e., infinite sequences of the form $\sigma_1\sigma_2\sigma_3\ldots$ with $\sigma_1 = \sigma$ and $P(\sigma_i, \sigma_{i+1}) > 0$. A probability measure $\Pr_{\mathcal{D}}$ is defined on measurable sets of infinite paths using a standard cylinder construction; for details, see, e.g., [2, Ch. 10]. For $T \subseteq \Sigma$ and $\sigma \in \Sigma$, let

$$\Pr_{\mathcal{D}}(\Diamond T) = \Pr_{\mathcal{D}}\{ \sigma_1\sigma_2\sigma_3\ldots \in Paths(\sigma) \mid \exists i. \, \sigma_i \in T \} \tag{1}$$

denote the probability to eventually reach some state in T from σ. For pMC $\mathcal{M}$, $\Pr_{\mathcal{M}}(\lozenge T)$ is a function with $\Pr_{\mathcal{M}}(\lozenge T)[u] = \Pr_{\mathcal{D}}(\lozenge T)$ where $\mathcal{D} = \mathcal{M}[u]$, see [23]. *Parameter Synthesis Problems on pMCs.* We consider the following synthesis problems on pMCs. Let $\mathcal{M} = (\Sigma, \sigma_I, X, P)$ be a pMC and $\lambda \in \mathbb{Q} \cap [0, 1]$ a threshold, $\sim$ a binary comparison operator, e.g., $<$ or $\geq$, and $T \subseteq \Sigma$.

Feasibility Problem. Find a parameter instantiation u s.t. $\Pr_{\mathcal{M}[u]}(\lozenge T) \sim \lambda$.
Synthesis Problem. Partition a region R into R_a and R_r s.t.

$$\Pr_{\mathcal{M}[u]}(\lozenge T) \sim \lambda \text{ for all } u \in R_a \quad \text{and} \quad \Pr_{\mathcal{M}[u]}(\lozenge T) \not\sim \lambda \text{ for all } u \in R_r.$$

  R_a is called an accepting region and R_r a rejecting region.
Approximate Synthesis Problem. Partition a region R into an accepting region R_a, rejecting region R_r, and unknown region R_u, such that $R_a \cup R_r$ covers at least $c\%$ of R. Additionally, R_a, R_r, and R_u should be finite unions of rectangular regions.
Verification Problem. Check whether region R is accepting, rejecting, or inconsistent, i.e., neither accepting nor rejecting.
Computing Reachability Functions. Compute the rational function $\Pr_{\mathcal{M}}(\lozenge T)$.

Algorithms for pMC Synthesis Problems. Several approaches have been developed to *compute the reachability function* $\Pr_{\mathcal{M}}(\lozenge T)$. This includes state elimination [23], fraction-free Gaussian elimination [1] and decomposition [28,33]. The reachability function can grow exponentially in the number of parameters, even for acyclic pMCs [1]. *Feasibility* is a computationally hard problem: finding parameter values for a pMC that satisfy a reachability objective is ETR-complete[4] [36]. Feasibility has been tackled using sampling search methods such as PSO[5] and Markov Chain Monte Carlo [16] and solving a non-linear optimization problem [4]. State-of-the-art approaches [20,21] iteratively simplify the NLP[6] encoding around a point to guide the search. The *approximate synthesis problem* checking is best tackled with parameter lifting [41]. The parameter lifting algorithm (PLA) first drops all dependencies between parameters in a pMC. It then transforms the pMC into a Markov decision process to get upper and lower bounds for the given objective.

4   Analysing Parametric BNs Using pMC Techniques

The key of our approach to tackle various synthesis problems on pBNs is to exploit pMC techniques. To that end, we transform a pBN into a pMC. We first present a recipe that is applicable to all inference queries on pBNs, and then detail a transformation that is tailored to the evidence in an inference query.

---

[4] Existential Theory of the Reals. ETR problems are between NP and PSPACE, and ETR-hard problems are as hard as finding the roots of a multi-variate polynomial.
[5] Particle swarm optimization.
[6] Nonlinear programming.

A pBN2pMC Transformation. This is inspired by our mapping of BNs onto tree-shaped MCs [43]. Here, we propose an alternative transformation that yields more succinct (p)MCs, as it only keeps track of a subset of (p)BN variables at each "level" of the (p)MC. These are the so-called *open variables* whose valuations are necessary to determine the subsequent transitions. Intuitively, the variable v_i is open if it has already been processed and at least one of its children has not. Below, we use $*$ if the value of variable $v \in V$ is *don't care*, i.e., either not yet determined or not needed any more.

Definition 3. *Let $G = (V, E)$ be a DAG with topological order $\varrho = (v_1, \ldots, v_m)$ on V. Let $v_i \in V$ be* open *at level[7] j iff $i <_\varrho j$ and $\exists v_k \in children(v_i). k >_\varrho j$. Let $open_\varrho(j)$ denote the set of open variables at level j under ϱ.*

Definition 4. *For pBN $\mathcal{B} = (V, E, X, \Theta)$ with topological order $\varrho = (v_1, \ldots, v_m)$ on V and $dom(v_i) = D_{v_i}$, let the pMC $\mathcal{M}_\mathcal{B}^\varrho = (\Sigma, \sigma_I, X, P)$ where:*

- $\Sigma = \bigcup_{j=1}^{m} \prod_{i=1}^{m} \{v_i\} \times T_j(D_{v_i})$ *with* $T_j(D_{v_i}) = \begin{cases} D_{v_i} & \text{if } i = j \text{ or } v_i \in open_\varrho(j), \\ \{*\} & \text{otherwise} \end{cases}$
- $\sigma_I = V \times \{*\}$ *is the initial state, and*
- P *is the transition probability function defined for $f \in \mathbb{Q}(X)$ and function*

$$t_j(d_i) = \begin{cases} d_i & \text{if } v_i \in open_\varrho(j) \\ * & \text{otherwise} \end{cases}$$

by the following inference rules:

$$\frac{\Theta_{v_1}(d_1) = f}{\sigma_I \xrightarrow{f} \big((v_1, d_1), (v_2, *), \ldots, (v_m, *)\big)} \tag{2}$$

$$\frac{\sigma = \big((v_1, t_{i-1}(d_1)), \ldots, (v_{i-2}, t_{i-1}(d_{i-2})), (v_{i-1}, d_{i-1}), (v_i, *), \ldots, (v_m, *)\big),}{\sigma \models \underline{parents(v_i)}, \quad \Theta_{v_i}(\underline{parents(v_i)})(d_i) = f}{\sigma \xrightarrow{f} \sigma' = \big((v_1, t_i(d_1)), \ldots, (v_{i-1}, t_i(d_{i-1})), (v_i, d_i), (v_{i+1}, *), \ldots, (v_m, *)\big)} \tag{3}$$

$$\sigma_F = \big((v_1, *), \ldots, (v_{m-1}, *), (v_m, d_m)\big) \xrightarrow{1} \sigma_F \tag{4}$$

The states in pMC $\mathcal{M}_\mathcal{B}^\varrho$ are tuples of pairs (v_i, d_i) where v_i is a random variable in $\mathcal{B}$ and $d_i \in dom(v_i) \cup \{*\}$ is the current value of v_i. Note that (v_i, d_i) encodes $v_i = d_i$. In the initial state all variables are *don't care*. The (parametric) function P specifies the probability of evolving between pMC states, which is determined by *parametric* CPT entries of the pBN $\mathcal{B}$. Rule (2) defines the outgoing transitions of the initial state σ_I. Let $d_1 \in D_{v_1}$ and $f = \Theta_{v_1}(d_1)$ be v_1's CPT entry for d_1. σ_I evolves—with the parametric probability function f—to a state in which all variables are $*$ except that $v_1 = d_1$. Let $\sigma_F = ((v_1, *), \cdots, (v_m, d_m))$

---

[7] Levels are definable on $\mathcal{M}_\mathcal{B}^\varrho$ as Definition 4 does not impose any backward transition.

be the final state of $\mathcal{M}_{\mathcal{B}}^{\varrho}$. Since v_m has no children, $open_{\varrho}(m)=\emptyset$. It also holds that $open_{\varrho}(m-1)-\{v_m\}=\emptyset$ as v_{m-1} can not have any other unprocessed children than v_m. Thus all variables, except for v_m, have been reset to $*$ in σ_F, see rule (4). Rule (3) specifies the transitions from the states that are neither initial nor final. Let σ be the state in which $v_1, \cdots, v_{i-1}$ have been already processed, i.e., have been assigned a value in the path from σ_I to σ; some may have been reset to $*$ afterwards. Functions $t_j(d_i)$ ensures that only variables with no unprocessed children are reset. Let $\underline{parents(v_i)}$ be a joint variable evaluation for v_i's parents. The second premise in rule (3) confirms that σ is consistent with $\underline{parents(v_i)}$. Let $f=\Theta(\underline{parents(v_i)})(d_i)$ be v_i's CPT entry for $\underline{parents(v_i)}$ and $v_i=\underline{d_i}$. Function f accordingly determines the (parametric) transition probability from σ to σ', i.e. a state with $v_i=d_i$ and the other variables determined by $t_i(d_j)$. Intuitively, for $j < i$, σ' forgets v_j if v_i has been v_j's only unprocessed child.

Example. Figure 2 (left) indicates the pMC for the pregnancy test pBN and the topological ordering (P, U, B) by definition 4. The node names are abbreviated by the first letter and the "don't care" evaluations are omitted.
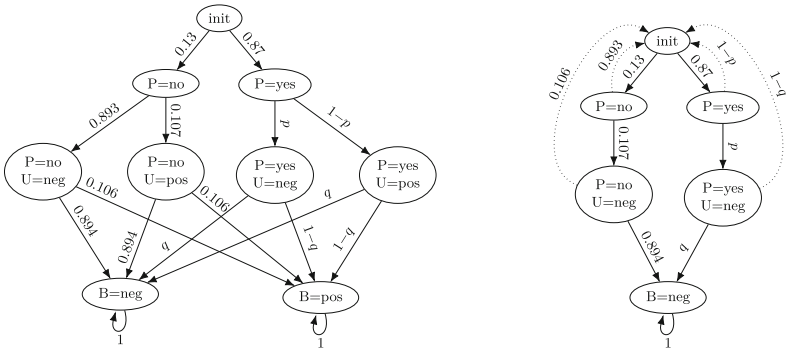

Fig. 2. The generated pMCs for the pregnancy test example based on (left) pBN2pMC and (right) evidence-guided pBN2pMC transformation.

The following result relates (conditional) inference in pBNs to (conditional) reachability objectives in pMCs.

Proposition 1. *Let $\mathcal{B}$ be a pBN. Let $E, H \subseteq V$ and ϱ be a topological ordering on V. Then, for the pMC $\mathcal{M}_{\mathcal{B}}^{\varrho}$ we have:*

$$\Pr_{\mathcal{B}}(E) \;=\; 1 - \Pr_{\mathcal{M}_{\mathcal{B}}^{\varrho}}(\Diamond \neg E) \quad and \quad \Pr_{\mathcal{B}}(H \mid E) \;=\; \frac{1 - \Pr_{\mathcal{M}_{\mathcal{B}}^{\varrho}}(\Diamond (\neg H \vee \neg E))}{1 - \Pr_{\mathcal{M}_{\mathcal{B}}^{\varrho}}(\Diamond \neg E)},$$

where the latter equality requires $\Pr_{\mathcal{B}}(\neg E) < 1$.

This result directly enables to use techniques for *feasibility checking* on pMCs to pBNs, and the use of techniques to compute reachability functions on pMCs to *computing sensitivity functions* on pBNs.

An Evidence-Tailored pBN2pMC Transformation. The above transformation is agnostic from the inference query. We now construct a pMC $\mathcal{M}_{\mathcal{B},E}^{\varrho}$ tailored to a given evidence $E \subseteq V$. This transformation is inspired by a transformation on MCs [3] for computing conditional reachability probabilities. Let $\mathcal{B} = (V, E, X, \Theta)$ be a pBN and ϱ be a topological order on V. Let evidence $E = (v_1 = d_1) \wedge \cdots \wedge (v_k = d_k)$, such that $v_1 <_\varrho \ldots <_\varrho v_k$. We construct pMC $\mathcal{M}_{\mathcal{B},E}^{\varrho}$ by the following two amendments on the pMC $\mathcal{M}_{\mathcal{B}}^{\varrho}$ as defined in Definition 4:

1. Let $\Sigma_{\neg E} = \{\sigma \in E \mid \exists i. \sigma \models (v_i = \neg d_i)\}$ be the set of states violating E. We redirect all incoming transitions of $\sigma \in \Sigma_{\neg E}$ to the initial state σ_I.
2. For $v_j \notin E$ with $v_j <_\varrho v_k$, we propagate the values of v_j until the level k. In other words, we pretend v_{k+1} is the child of v_j, so we keep v_j open.

Let us formalize this. Let Σ' and P' be defined analogously to Σ and P in Definition 4, except that the definition of *open set* changes as described above, affecting the definitions of $T_j(v_i)$ and $t_j(d_i)$. Then, let $\mathcal{M}_{\mathcal{B},E}^{\varrho} = (\Sigma_E, \sigma_I, X, P_E)$, where

$$
\Sigma_E = \Sigma' \setminus \Sigma_{\neg E} \quad \text{and} \quad P_E(\sigma, \sigma') = \begin{cases} \displaystyle\sum_{\sigma' \not\models E} P'(\sigma, \sigma') & \text{if } \sigma' \in \Sigma_{\neg E} \\ P'(\sigma, \sigma') & \text{otherwise.} \end{cases}
$$

Example. Fig. 2 (right) indicates the evidence-guided pMC generated for our running example, the ordering (P, U, B), and the evidence $U = neg \wedge B = neg$.

Proposition 2. *For the evidence-tailored MC $\mathcal{M}_{\mathcal{B},E}^{\varrho}$ of pBN $\mathcal{B}$, we have:*

$$
\Pr_{\mathcal{B}}(H \mid E) = 1 - \Pr_{\mathcal{M}_{\mathcal{B},E}^{\varrho}} (\Diamond \neg H). \tag{5}
$$

This result facilitates using pMC techniques for pBN *parameter tuning*.

Ratio and Difference Parameter Tuning by Parameter Lifting. The ratio problem on pBN $\mathcal{B}$ corresponds to finding an instantiation u in the pMC $\mathcal{M}_{\mathcal{B},E}^{\varrho}$ s.t.

$$
\Pr_{\mathcal{M}_{\mathcal{B},E}^{\varrho}} [u](\Diamond T) \geq q \cdot \Pr_{\mathcal{M}_{\mathcal{B},E}^{\varrho}} [u](\Diamond G), \tag{6}
$$

where $\Pr(\Diamond T)$ stands for $1 - \Pr_{\mathcal{M}_{\mathcal{B},E}^{\varrho}}[u](\Diamond(H = \neg h' \vee E = \neg e))$ and $\Pr(\Diamond G)$ abbreviates $1 - \Pr_{\mathcal{M}_{\mathcal{B},E}^{\varrho}}[u](\Diamond(H = \neg h \vee E = \neg e))$. This problem can be solved by using PLA: let region $R \subseteq \mathbb{R}_{\geq 0}^n$. We perform PLA for reaching G and reaching T on $\mathcal{M}_{\mathcal{B}}$, respectively. This gives upper (UB_T, UB_G) and lower bounds (LB_T, LB_G) for the probabilities on the left- and the right-hand side of (6). Then:

– If $LB_T \geq q \cdot UB_G$, the region R is accepting for the ratio property.

– If $LB_T \leq q \cdot UB_G$, the region R is rejecting for the ratio property.
– Otherwise, refine the region R.

For difference parameter tuning, we adopt the above recipe by replacing (6) by:

$$\Pr_{\mathcal{M}_{\mathcal{B},E}^\varrho} [u](\Diamond T) \ \geq \ q + \Pr_{\mathcal{M}_{\mathcal{B},E}^\varrho} [u](\Diamond G). \tag{7}$$

5   Experiments

Our pBN Analysis Tool. We developed a prototypical tool on top of the tools Storm [25] and Prophesy [24], see Fig. 3. Storm is a probabilistic model checker that dominated the last (and only) two model-checking competitions, see https://qcomp.org/; Prophesy is an efficient tool for pMC synthesis. Our tool deploys pMC parameter synthesis techniques to analyze pBNs. It includes both pBN2pMC transformations where pBNs are provided in an extended `bif` format. The pMCs are either encoded in Jani [8] or in the explicit `drn` format. It is also possible to transform non-parametric BNs into MCs and parameterize the MC. Storm is used to compute the sensitivity function and for parameter tuning using PLA. Prophesy is exploited for feasibility checking: find a parameter instance satisfying an inference query. Our tool-chain supports $p_* c_* r_*$, the general pBNs class. As baseline we used two synthesis tools for parametric BNs: SamIam and Bayesserver.
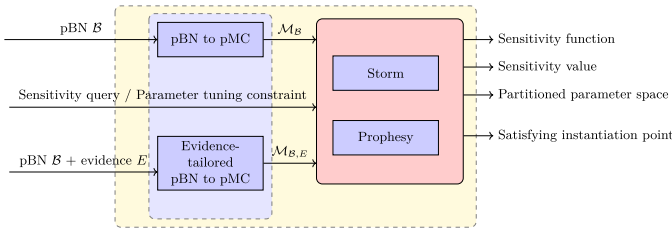

Fig. 3. Our prototypical tool-chain for synthesis problems on pBNs

SamIam. SamIam[8] is a commonly used tool for the sensitivity analysis for pBNs, developed at Darwiche's group at UCLA. It allows the specification of conditional, hypothesis ratio, and hypothesis difference constraints on pBNs. SamIam then attempts to identify minimal parameter changes that are necessary to satisfy these constraints. It supports the pBN classes $p_1 c_1 r_1$ and $p_* c_1 r_1$.

Bayesserver. Bayesserver[9] is a commercial tool that offers sensitivity analysis and parameter tuning of pBNs. For sensitivity analysis, it computes the sensitivity

---

[8] http://reasoning.cs.ucla.edu/samiam.
[9] https://www.bayesserver.com.

function and sensitivity value. It also performs minimal-change parameter tuning for conditional, hypothesis ratio, and hypothesis difference constraints. It supports the classes $p_1c_1r_1$ and $p_2c_{\leq 2}r_1$ for sensitivity analysis and the class $p_1c_1r_1$ for parameter tuning. Table 1 lists the functionalities of all tools.

Table 1. Overview of the capabilities of the pBN synthesis tools considered.

	SamIam	Bayesserver	Storm-prophesy
Computing sensitivity function	✗	$p_{\leq 2}c_{\leq 2}r_1$	$p_*c_*r_*$
Computing sensitivity value	✗	$p_{\leq 2}c_{\leq 2}r_1$	$p_1c_*r_*$
Simple parameter tuning	$p_*c_1r_1$	$p_1c_1r_1$	$p_*c_*r_*$
Difference parameter tuning	$p_*c_1r_1$	$p_1c_1r_1$	$p_*c_*r_*$
Ratio parameter tuning	$p_*c_1r_1$	$p_1c_1r_1$	$p_*c_*r_*$
Minimal change tuning	$p_*c_1r_1$	$p_1c_1r_1$	$p_*c_*r_*$

Experimental Set-up. We took benchmarks from [45] and conducted all our experiments on a 2.3 GHz Intel Core i5 processor with 16 GB of RAM. We focused on questions such as:

1. What is the scalability for computing sensitivity functions on pBNs?
2. What is the practical scalability for feasibility checking?
3. To what extent is PLA applicable to parameter tuning for pBNs?

Computing Sensitivity Function. We performed a series of experiments for computing pBN sensitivity functions using our tool-chain for the $p_*c_*r_1$ class. Figure 4 summarizes the results. The $x-$axis (log scale) indicates the pBN benchmarks and the $y-$axis denotes the timing in seconds. The numbers on the bars indicate the number of parameters in the solution functions, which is related to the number of relevant parameters identified for the given query. We observe that
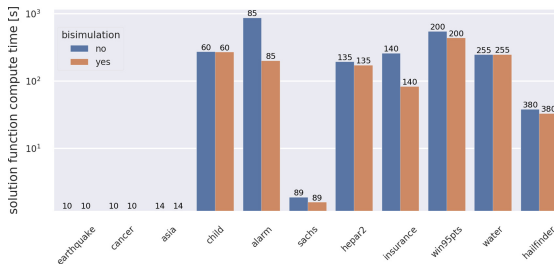

Fig. 4. Storm's performance for calculating prior sensitivity functions of pBNs. (Unfortunately, a comparison with the other tools was not possible, as SamIam does not explicitly offers sensitivity function computation and Bayesserver sensitivity analysis is limited to 1 or 2 parameters, see Table 1.)

Storm scales up to 380 parameters for very large networks such as `hailfinder`. The blue bars represent regular computations, while the orange bars indicate the impact of bisimulation minimization, a built-in reduction technique in Storm.

Feasibility Checking. Our tool exploits Prophesy to find a parameter instantiation u of pBN $\mathcal{B}$ such that the BN $\mathcal{B}[u]$ satisfies the given inference query. We have performed a set of experiments for the class $p_* c_* r_1$. Figure 5 (log-log scale) illustrates the results; the x-axis indicates the number of parameters in the pBN and the y-axis the time for feasibility checking (in seconds). Each line corresponds to a pBN and the points on the lines represent single experiments. We inserted the parameters in the predecessors of the query nodes (i.e., in H) to maximize their relevance. We also imposed queries over multiple nodes at once to push the boundaries. We used convex optimization (QCQP[10]) (left plot) and PSO (right plot). Prophesy was able to handle up to 853 parameters.
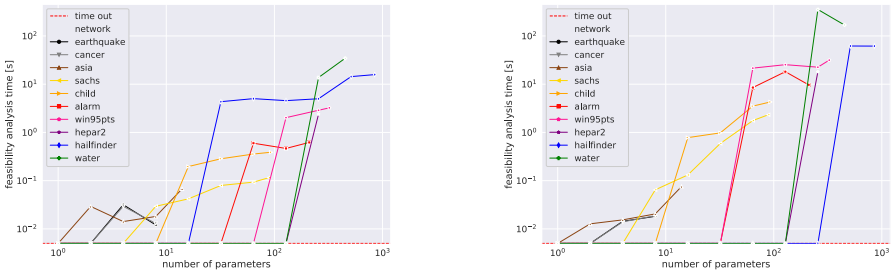

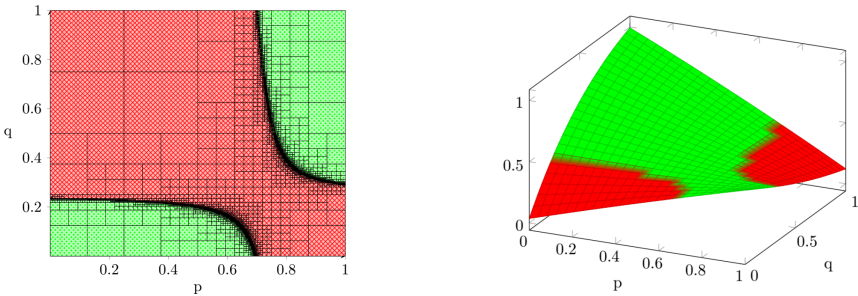**Fig. 5.** Feasibility checking on pBN benchmarks by (left) QCQP and (right) PSO.

Fig. 6. PLA results on the `alarm` pBN ($p_2 c_3 r_{26}$) for the constraint $\Pr(venttube = 0 \mid ventlung = 0) > 0.6$ with a 99% parameter space coverage. (Color figure online)

Approximate Parameter Synthesis on pBNs: Tuning the Parameters and More. Experiments on the pBN benchmarks using PLA aimed at (a) the classes $p_1 c_1 r_1$

---

[10] Quadratically-constrained quadratic programming.

and $p_* c_1 r_1$ to validate them against SamIam and Bayesserver, and (b) the class $p_* c_* r_*$ to investigate the potential of PLA for general pBNs. Figure 6 visualizes the results for the `alarm` pBN with 2 parameters occurring in 26 rows of 3 CPTs, i.e., a pBN with parameter dependencies. The parameter x was used in the CPT entries $e_1, \cdots, e_k$ only when the probability values of those entries coincided in the original BN. As seen in the figure, PLA can partition the entire n-way parameter space. The minimal-change parameter values can be extracted from the PLA results, where the precision depends on the PLA approximation factor.

6   Conclusion

This paper exploited tools and techniques for parameter synthesis on Markov chains to synthesis problems on parametric Bayesian networks. Prototypical tool support for pBN analysis on top of existing pMC synthesis tools has been realized. Our experiments indicate that pMC techniques can scale sensitivity analysis and parameter tuning tasks on pBNs. The experiments reveal the potential of parameter lifting [41] for partitioning the parameter space of pBNs. Most importantly, the proposed techniques are applicable to *general* pBNs—no restrictions are imposed on the number or occurrence of parameters—and may involve parameter dependencies. Future work include finding optimal parameter settings [47], exploiting monotonicity checking [46] and to extend the current work to (parametric) dynamic, Gaussian [12], and recursive BNs [32].

References

1. 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)
2. Baier, C., Katoen, J.: Principles of Model Checking. MIT Press (2008)
3. Baier, C., Klein, J., Klüppelholz, S., Märcker, S.: Computing conditional probabilities in Markovian models efficiently. In: Ábrahám, E., Havelund, K. (eds.) TACAS 2014. LNCS, vol. 8413, pp. 515–530. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54862-8_43
4. Bartocci, E., Grosu, R., Katsaros, P., Ramakrishnan, C.R., Smolka, S.A.: Model repair for probabilistic systems. In: Abdulla, P.A., Leino, K.R.M. (eds.) TACAS 2011. LNCS, vol. 6605, pp. 326–340. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19835-9_30
5. Bartocci, E., Kovács, L., Stanković, M.: Analysis of Bayesian networks via probsolvable loops. In: Pun, V.K.I., Stolz, V., Simao, A. (eds.) ICTAC 2020. LNCS, vol. 12545, pp. 221–241. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-64276-1_12

6. Beaumont, P., Huth, M.: Constrained Bayesian Networks: Theory, Optimization, and Applications. CoRR https://arxiv.org/abs/1705.05326 (2017)

7. Bolt, J.H., van der Gaag, L.C.: Balanced tuning of multi-dimensional Bayesian network classifiers. In: Destercke, S., Denoeux, T. (eds.) ECSQARU 2015. LNCS (LNAI), vol. 9161, pp. 210–220. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-20807-7_19

8. Budde, C.E., Dehnert, C., Hahn, E.M., Hartmanns, A., Junges, S., Turrini, A.: JANI: quantitative model and tool interaction. In: Legay, A., Margaria, T. (eds.) TACAS 2017. LNCS, vol. 10206, pp. 151–168. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54580-5_9

9. Castillo, E., Gutiérrez, J.M., Hadi, A.S.: Parametric structure of probabilities in Bayesian networks. In: Froidevaux, C., Kohlas, J. (eds.) ECSQARU 1995. LNCS, vol. 946, pp. 89–98. Springer, Heidelberg (1995). https://doi.org/10.1007/3-540-60112-0_11

10. Castillo, E.F., Gutiérrez, J.M., Hadi, A.S.: Goal oriented symbolic propagation in Bayesian networks. In: AAAI/IAAI, vol. 2. pp. 1263–1268. AAAI Press/The MIT Press (1996)

11. Castillo, E.F., Gutiérrez, J.M., Hadi, A.S.: Sensitivity analysis in discrete Bayesian networks. IEEE Trans. Syst. Man Cybern. Part A **27**(4), 412–423 (1997)

12. Castillo, E.F., Gutiérrez, J.M., Hadi, A.S., Solares, C.: Symbolic propagation and sensitivity analysis in Gaussian Bayesian networks with application to damage assessment. Artif. Intell. Eng. **11**(2), 173–181 (1997)

13. Chan, H., Darwiche, A.: When do numbers really matter? J. Artif. Intell. Res. **17**, 265–287 (2002)

14. Chan, H., Darwiche, A.: Sensitivity analysis in Bayesian networks: from single to multiple parameters. In: UAI, pp. 67–75. AUAI Press (2004)

15. Chan, H., Darwiche, A.: A distance measure for bounding probabilistic belief change. Int. J. Approx. Reason. **38**(2), 149–174 (2005)

16. Chen, T., Hahn, E.M., Han, T., Kwiatkowska, M.Z., Qu, H., Zhang, L.: Model repair for Markov decision processes. In: TASE. IEEE (2013)

17. Coupé, V.M.H., van der Gaag, L.C.: Properties of sensitivity analysis of Bayesian belief networks. Ann. Math. Artif. Intell. **36**(4), 323–356 (2002)

18. Coupe, V.M., van der Gaag, L.C.: Practicable sensitivity analysis of Bayesian belief networks, vol. 1998. Utrecht University, Information and Computing Sciences (1998)

19. Coupé, V.M., van der Gaag, L.C., Habbema, J.D.F.: Sensitivity analysis: an aid for belief-network quantification. Knowl. Eng. Rev. **15**(3), 215–232 (2000)

20. Cubuktepe, M., et al.: Sequential convex programming for the efficient verification of parametric MDPs. In: Legay, A., Margaria, T. (eds.) TACAS 2017. LNCS, vol. 10206, pp. 133–150. Springer, Heidelberg (2017). https://doi.org/10.1007/978-3-662-54580-5_8

21. Cubuktepe, M., Jansen, N., Junges, S., Katoen, J.-P., Topcu, U.: Synthesis in pMDPs: a tale of 1001 parameters. In: Lahiri, S.K., Wang, C. (eds.) ATVA 2018. LNCS, vol. 11138, pp. 160–176. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01090-4_10

22. Darwiche, A.: Modeling and Reasoning with Bayesian Networks. Cambridge University Press (2009)

23. 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

24. 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

25. Dehnert, C., Junges, S., Katoen, J.-P., Volk, M.: A storm is coming: a modern probabilistic model checker. In: Majumdar, R., Kunčak, V. (eds.) CAV 2017. LNCS, vol. 10427, pp. 592–600. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63390-9_31

26. Druzdzel, M.J., van der Gaag, L.C.: Building probabilistic networks: "where do the numbers come from?". IEEE Trans. Knowl. Data Eng. **12**(4), 481–486 (2000)

27. Fang, X., Calinescu, R., Gerasimou, S., Alhwikem, F.: Fast parametric model checking through model fragmentation. In: ICSE, pp. 835–846. IEEE (2021)

28. Fang, X., Calinescu, R., Gerasimou, S., Alhwikem, F.: Fast parametric model checking through model fragmentation. CoRR https://arxiv.org/abs/2102.01490 (2021)

29. van der Gaag, L.C., Renooij, S., Coupé, V.M.H.: Sensitivity analysis of probabilistic networks. In: Lucas, P., Gámez, J.A., Salmerón, A. (eds.) Advances in Probabilistic Graphical Models. SFSC, vol. 213, pp. 103–124. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-68996-6_5

30. 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

31. Hahn, E.M., Hermanns, H., Zhang, L.: Probabilistic reachability for parametric Markov models. Int. J. Softw. Tools Technol. Transf. **13**(1), 3–19 (2011)

32. Jaeger, M.: Complex probabilistic modeling with recursive relational Bayesian networks. Ann. Math. Artif. Intell. **32**(1–4), 179–220 (2001)

33. 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

34. Jensen, F.V.: Gradient descent training of Bayesian networks. In: Hunter, A., Parsons, S. (eds.) ECSQARU 1999. LNCS (LNAI), vol. 1638, pp. 190–200. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48747-6_18

35. Junges, S., et al.: Parameter synthesis for Markov models. CoRR https://arxiv.org/abs/1903.07993 (2019)

36. Junges, S., Katoen, J.P., Pérez, G.A., Winkler, T.: The complexity of reachability in parametric Markov decision processes. J. Comput. Syst. Sci. **119**, 183–210 (2021)

37. Kjærulff, U., van der Gaag, L.C.: Making sensitivity analysis computationally efficient. In: UAI, pp. 317–325. Morgan Kaufmann (2000)

38. Kwisthout, J., van der Gaag, L.C.: The computational complexity of sensitivity analysis and parameter tuning. In: UAI, pp. 349–356. AUAI Press (2008)

39. Lanotte, R., Maggiolo-Schettini, A., Troina, A.: Parametric probabilistic transition systems for system design and analysis. Formal Aspects Comput. **19**(1), 93–109 (2007)

40. Laskey, K.B.: Sensitivity analysis for probability assessments in Bayesian networks. IEEE Trans. Syst. Man Cybern. **25**(6), 901–909 (1995)

41. 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

42. Renooij, S.: Co-variation for sensitivity analysis in Bayesian networks: properties, consequences and alternatives. Int. J. Approx. Reason. **55**(4), 1022–1042 (2014)

43. Salmani, B., Katoen, J.-P.: Bayesian inference by symbolic model checking. In: Gribaudo, M., Jansen, D.N., Remke, A. (eds.) QEST 2020. LNCS, vol. 12289, pp. 115–133. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-59854-9_9
44. Salmani, B., Katoen, J.: Fine-tuning the odds in Bayesian networks. CoRR https://arxiv.org/abs/2105.14371 (2021)
45. Scutari, M.: Bayesian network repository. https://www.bnlearn.com. Accessed 2019
46. Spel, J., Junges, S., Katoen, J.-P.: Are parametric Markov chains monotonic? In: Chen, Y.-F., Cheng, C.-H., Esparza, J. (eds.) ATVA 2019. LNCS, vol. 11781, pp. 479–496. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-31784-3_28
47. 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
48. Tolo, S., Patelli, E., Beer, M.: An open toolbox for the reduction, inference computation and sensitivity analysis of Credal networks. Adv. Eng. Softw. **115**, 126–148 (2018)