Finding an ϵ-Close Minimal Variation of Parameters in Bayesian Networks

Bahare Salmani and **Joost-Pieter Katoen**
RWTH Aachen University
{salmani, katoen}@cs.rwth-aachen.de

Abstract

This paper addresses the ϵ-close parameter tuning problem for Bayesian networks (BNs): find a minimal ϵ-close amendment of probability entries in a given set of (rows in) conditional probability tables that make a given quantitative constraint on the BN valid. Based on the state-of-the-art "region verification" techniques for parametric Markov chains, we propose an algorithm whose capabilities go beyond any existing techniques. Our experiments show that ϵ-close tuning of large BN benchmarks with up to eight parameters is feasible. In particular, by allowing (i) varied parameters in multiple CPTs and (ii) inter-CPT parameter dependencies, we treat subclasses of parametric BNs that have received scant attention so far.

1 Introduction

Bayesian networks. Bayesian networks (BNs) [Pearl, 1988] are probabilistic graphical models that enable succinct knowledge representation and facilitate probabilistic reasoning [Darwiche, 2009]. Parametric Bayesian networks (pBNs) [Castillo *et al.*, 1997] extend BNs by allowing polynomials in conditional probability tables (CPTs) rather than constants.

Parameter synthesis on Markov models. Parameter synthesis is to find the right values for the unknown parameters with respect to a given constraint. Various synthesis techniques have been developed for parametric Markov chains (pMCs) ranging over e.g., the gradient-based methods [Heck *et al.*, 2022], convex optimization [Cubuktepe *et al.*, 2018; Cubuktepe *et al.*, 2022], and region verification [Quatmann *et al.*, 2016]. Recently, Salmani and Katoen [2021a] have proposed a translation from pBNs to pMCs that facilitates using pMC algorithms to analyze pBNs. Proceeding from this study, *we tackle a different problem* [Kwisthout and van der Gaag, 2008] for Bayesian networks.

Minimal-change parameter tuning. Given a Bayesian network B, a hypothesis H, evidence E, $\lambda \in [0,1]$, and a constraint of the form $\Pr(H|E) \leq \lambda$ (or $\geq \lambda$),

> what is a minimal change—with respect to a given measure of distance—in the probability values of (a subset of) CPT rows, such that the constraint holds?

We illustrate the problem with an example of testing COVID-19 that is adopted from several medical studies [Barreiro *et al.*, 2021; Nishiura *et al.*, 2020; Dinnes *et al.*, 2022]. The outcome of PCR tests only depends on whether the person is infected, while the antigen tests are less likely to correctly identify COVID-19 if the infection is asymptomatic (or presymptomatic). Figure 1a depicts a Bayesian network that models such probabilistic dependencies. In the original network, the probability of no COVID-19 given that both tests are positive, is 0.011089. Assume that in an application domain, the result of such a query is now required not to exceed 0.009: the aim is to make this constraint hold while imposing the least change in the original network with respect to a distance measure. This is an instance of the *minimal-change parameter tuning* problem. We consider the $\epsilon-$bounded variant of the problem: for a subset of *modifiable* CPT rows, are there new values within the distance of ϵ from the original probability values that make the constraint hold?

Main contributions. Based on existing *region verification* and *region partitioning* techniques for pMCs,

- we propose a practical algorithm for ϵ-bounded tuning. More precisely, we find instantiations that (i) satisfy the constraint (if the constraint is satisfiable) and (ii) are ϵ-close and (iii) lean towards the minimum distance instantiation depending on a coverage factor $0 \leq \eta \leq 1$.
- We propose two region expansion schemes to realize ϵ-closeness of the results both for Euclidean distance and for CD distance [Chan and Darwiche, 2005].
- Contrary to the existing techniques that restrict to hyperplane solution spaces, we handle pBNs with multiple parameters in multiple distributions and multiple CPTs.

Our experiments on our prototypical implementation [1] indicate that ϵ-bounded tuning of up to 8 parameters for large networks with 100 variables is feasible.

Paper organization. Section 2 includes the basic notations and Sec. 3 the preliminaries on parametric Bayesian networks. Section 4 introduces parametric Markov chains and the region verification techniques thereof. Section 5 details our main contributions and Sec. 6 our experimental results. Section 7 concludes the paper with an overview of the related studies that were not mentioned before.

[1] https://github.com/baharslmn/pbn-epsilon-tuning

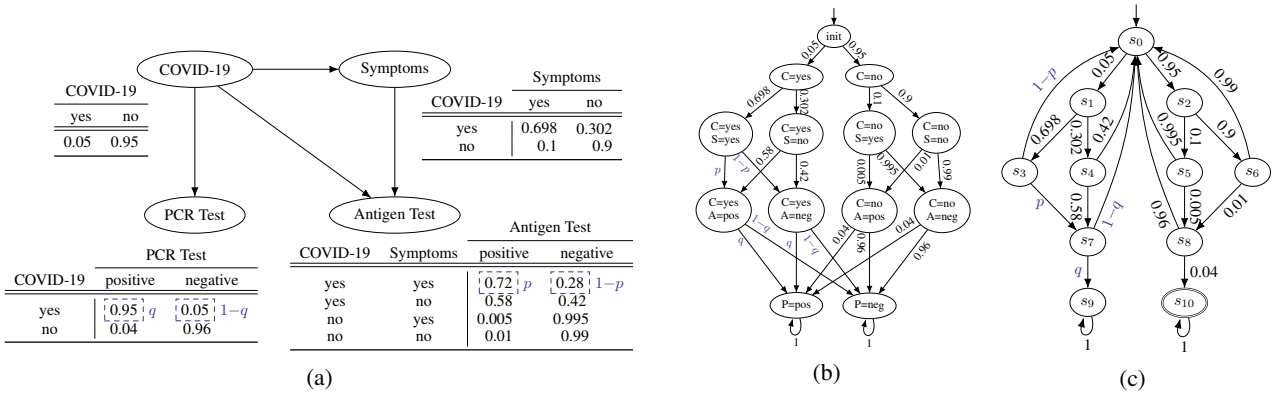


Figure 1: Our running example: (a) the parametric BN `COVID-19`, (b) the parametric MC of `COVID-19`, for the topological order *C;S;A;P*, (c) the parametric MC of `COVID-19` tailored to the evidence *Antigen Test* = *pos* and *PCR Test* = *pos*, abstracted from variable valuations.

2 Background

Variables. Let V be a set of m random variables $v_1, \ldots, v_m$ and D_{v_i} the domain of variable v_i. For $A \subseteq V$, $\mathit{Eval}(A)$ denotes the set of joint configurations for the variables in A.

Parameters. Let $X = \{x_1, \ldots, x_n\}$ be a set of n real-valued parameters. A *parameter instantiation* is a function $u : X \to \mathbb{R}$ that maps each parameter to a value. All parameters are bounded; i.e., $lb_{x_i} \leq u(x_i) \leq ub_{x_i}$ for x_i. Let $I_i = [lb_{x_i}, ub_{x_i}]$. The *parameter space* $\mathcal{U} \subseteq \mathbb{R}^n$ of X is the set of all possible values of X, i.e., the hyper-rectangle spanned by the intervals I_i for all i.

Substitution. Polynomials f over X are functions $f : \mathbb{R}^n \to \mathbb{R}$ where $f(u)$ is obtained by replacing each occurrence of x_i in f by $u(x_i)$; e.g., for $f = 2x_1^2 + x_2$, $u(x_1) = 3$ and $u(x_2) = 2$, $f(u) = 20$. Let $f[u]$ denote such substitution.

Parametric distributions. Let $\mathit{Distr}(D)$ denote the set of probability distributions over D. Let $\mathbb{Q}[X]$ be the set of multivariate polynomials with rational coefficients over X. A *parametric probability distribution* is the function $\mu : D \to \mathbb{Q}[X]$ with $\sum_{d \in D} \mu(d) = 1$. Let $\mathit{pDistr}(D)$ denote the set of *parametric* probability distributions over D with parameters in X. Instantiation u is *well-formed* for $\mu \in \mathit{pDistr}(D)$ iff $0 \leq \mu(d)[u] \leq 1$ and $\Sigma_{d \in D}\, \mu(d)[u] = 1$.

Distance measure. The function $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}^{\geq 0}$ is a *distance measure* if for all $u, u', u'' \in \mathcal{U}$, it satisfies: (I) positiveness: $d(u, u') \geq 0$, (II) symmetry: $d(u, u') = d(u', u)$, and (III) triangle inequality: $d(u, u'') \leq d(u, u') + d(u', u'')$.

3 Parametric Bayesian Networks

A *parametric Bayesian network* (pBN) is a BN in which a subset of entries in the conditional probability tables (CPTs) are polynomials over the parameters in X. Let par_{v_i} denote the set of parents for the node v_i in the graph G.

Definition 1. *The tuple $\mathcal{B}{=}(G, X, \Theta)$ is a* parametric Bayesian network *(pBN) with directed acyclic graph $G{=}(V, W)$ over random variables $V{=}\{v_1, \ldots, v_m\}$, set of parameters $X{=}\{x_1, \ldots, x_n\}$, and parametric CPTs $\Theta{=}\{\, \Theta_{v_i} \mid v_i{\in}V \,\}$ where Θ_{v_i}: $\mathit{Eval}(par_{v_i}) \to \mathit{pDistr}(D_{v_i})$.*

The CPT row $\Theta_v(\overline{par})$ is a parametric distribution over D_v given the parent evaluation $\overline{par}$. The *CPT entry* $\Theta_v(\overline{par})(d)$, short $\theta_{(v,d,\overline{par})}$, is the probability that $v{=}d$ given $\overline{par}$. A pBN without parameters, i.e., $X = \emptyset$, is an ordinary BN. A pBN $\mathcal{B}$ defines the *parametric* distribution function $\Pr_{\mathcal{B}}$ over $\mathrm{Eval}(V)$. For well-formed instantiation u, BN $\mathcal{B}[u] = (G, \Theta[u])$ is obtained by replacing the parameter x_i in the parametric functions in the CPTs of $\mathcal{B}$ by $u(x_i)$.

Example 1. *Fig. 1a shows a pBN over variables $V = \{C, S, A, P\}$ (initial letters of node names) and parameters $X = \{p, q\}$. Instantiating with $u_0(p) = 0.72$ and $u_0(q) = 0.95$ yields the BN $\mathcal{B}[u_0]$ as indicated using dashed boxes.*

pBN constraints. Constraints $\Pr_B(H|E) \sim \lambda$ involve a hypothesis H, an evidence E, $\sim \in \{\leq, \geq\}$ and threshold $0 \leq \lambda \leq 1$. For the instantiation $u : X \to \mathbb{R}$,

$$\mathcal{B}[u] \models \phi \quad \text{if and only if} \quad \Pr_{\mathcal{B}[u]}(H \mid E) \sim \lambda.$$

Sensitivity functions. Sensitivity functions are rational functions that relate the result of a pBN query to the parameters [Castillo *et al.*, 1997; Coupé and van der Gaag, 2002].

Example 2. *For the pBN $\mathcal{B}_1$ in Fig. 1a, let ϕ_1 be the constraint: $\Pr_{\mathcal{B}_1}(C = no \mid A = pos \wedge P = pos) \leq 0.009$. The left-hand side reduces to the sensitivity function*

$$f_{\mathcal{B}_1,\phi_1} = {}^{361}/_{34900 \cdot p \cdot q + 8758 \cdot q + 361}.$$

Using instantiation u with $u(p) = 0.92075$ and $u(q) = 0.97475$, $f_{\mathcal{B}_1,\phi_1}[u] = 0.008798 \leq 0.009$, i.e., $\mathcal{B}_1[u] \models \phi_1$.

Higher degree sensitivity functions. Contrary to the existing literature that is often restricted to multi-linear sensitivity functions, we are interested in analyzing pBNs with sensitivity functions of higher degrees.

Example 3. *Consider a variant of the COVID-19 example, where the person only takes the antigen test twice rather than taking both the antigen and PCR tests to diagnose COVID-19; see Fig. 2 for the parametric CPTs. Then,*

$$f_{\mathcal{B}_2,\phi_2} = \Pr_{\mathcal{B}_2}(C = no \mid A1 = pos \wedge A2 = pos)$$
$$= {}^{950 \cdot 9 \cdot t^2 \cdot s^2}/_{8850 \cdot t^2 + 349 \cdot p^2 + 950 \cdot s^2 + 151 \cdot r^2}.$$

		Antigen Test 1				Antigen Test 2			
C	S	pos		neg		C	S	pos		neg	
yes	yes	0.72	p	0.28	$1-p$	yes	yes	0.72	p	0.28	$1-p$
yes	no	0.58	r	0.42	$1-r$	yes	no	0.58	r	0.42	$1-r$
no	yes	0.005	s	0.995	$1-s$	no	yes	0.005	s	0.995	$1-s$
no	no	0.01	t	0.99	$1-t$	no	no	0.01	t	0.99	$1-t$

Figure 2: Parametric CPTs for a variant of `COVID-19` example with two antigen tests; see the inter-CPT parameter dependencies.

Parametrization. Let Θ_{modif} denote the CPT entries in BN $B = (G, \Theta)$ that are explicitly changed.

Definition 2 (BN parametrization)**.** *Tuple $\mathcal{B} = (G, X, \Theta')$ is* a parametrization of *BN $B = (G, \Theta)$ over X w.r.t. Θ_{modif} if*

$$\theta'_{(v,.,\overline{par})} = f \text{ for some } f \in \mathbb{Q}[X] \quad \text{iff} \quad \theta_{(v,d,\overline{par})} \in \Theta_{modif}.$$

The parametrization $\mathcal{B}$ is *monotone* iff for each parametric entry f, f is a monotonic polynomial. The parametrization $\mathcal{B}$ is *valid* iff $\mathcal{B}$ is a pBN, i.e., iff $\Theta'_v(\overline{par}) \in pDistr(D_v)$ for each random variable $v \in V$ and its parents configuration $\overline{par}$.
A valid monotone parametrization is obtained by the *linear proportional* [Coupé and van der Gaag, 2002] that is the most established scheme in the literature to parametrize BNs.

Definition 3 (Linear proportional parametrization)**.** *A* linear proportional parametrization *over X maps the CPTs Θ onto the parametric CPTs Θ' w.r.t. Θ_{modif} if*

$$\begin{cases} \theta'_{(v,d_k,\overline{par})} = x \in X & \text{for } \theta_{(v,d_k,\overline{par})} \in \Theta_{modif} \\ \theta'_{(v,d_j,\overline{par})} = \dfrac{1-x}{1-\theta_{(v,d_k,\overline{par})}} \cdot \theta_{(v,d_j,\overline{par})} & \text{for } d_j \neq d_k \in D_v. \end{cases}$$

Note that we allow the repetition of parameters in multiple distributions and multiple CPTs, see e.g., Fig. 2.

Formal problem statement. Consider BN $B = (G, \Theta)$, its valid parametrization $\mathcal{B} = (G, X, \Theta')$ over X with constraint ϕ. Let u_0 be the original value of the parameters X in B and $d : \mathcal{U} \times \mathcal{U} \to \mathbb{R}^{\geq 0}$ a distance measure. Let $\hat{d_0}$ denote an upperbound for $d_{\{u \in \mathcal{U}\}}(u_0, u)$. The *minimum-distance parameter tuning problem* is to find $u_{\min} = \text{argmin}_{\{u \in \mathcal{U} \mid \mathcal{B}[u] \models \phi\}} d(u, u_0)$. Its generalized variant for $0 \leq \epsilon \leq \hat{d_0}$ and $0 \leq \eta \leq 1$ is:

Definition 4. *The (ϵ, η)*-parameter tuning problem *is to find $u \in \mathcal{U}$ s.t. $\mathcal{B}[u] \models \phi$, $d(u, u_0) \leq \epsilon$, and $d(u, u_{\min}) \leq (1-\eta) \cdot \hat{d_0}$.*

$(\hat{d_0}, 1)$-tuning gives the minimum-distance tuning. We call η the coverage factor that determines, intuitively speaking, the minimality of the result; we discuss this in Sec. 5.4. We consider two distance measures.

Euclidean distance (EC distance). The EC-distance between u and u' (both in $\mathcal{U}$) is defined as:

$$EC(u, u') = \sqrt{\sum_{x_i \in X} \left(u(x_i) - u'(x_i)\right)^2}.$$

Corollary 1. *For $n = |X|$, $\hat{d_0} = \sqrt{n}$ is an upperbound for the EC distance of any u_0 from any instantiation $u \in \mathcal{U}$.*

Corollary 2. *Let $0 \leq \epsilon \leq \sqrt{n}$ and $u_0(x_i) - \frac{\epsilon}{n} \leq u(x_i) \leq u_0(x_i) + \frac{\epsilon}{n}$. Then, $EC(u, u_0) \leq \epsilon$.*

Chan-Darwiche distance (CD distance). CD is a measure to quantify the distance between the probability distributions of two BNs, defined as [Chan and Darwiche, 2005]:

$$CD(u, u') = \ln \max_{w \in \text{Eval}(V)} \frac{\Pr_{\mathcal{B}[u']}(w)}{\Pr_{\mathcal{B}[u]}(w)} - \ln \min_{w \in \text{Eval}(V)} \frac{\Pr_{\mathcal{B}[u']}(w)}{\Pr_{\mathcal{B}[u]}(w)},$$

where both $0/0 = 1$ and $\infty/\infty = 1$ by definition. Whereas the EC distance can be computed in $O(n)$, this is for CD distance NP-complete [Kwisthout and van der Gaag, 2008]. It has known closed-forms only for limited cases, e.g., single parameter and single-CPT pBNs [Chan and Darwiche, 2004]. Let Θ_v be the CPT that is parametrized to Θ'_v by a monotone parametrization. Let the minimum (maximum) probability entry of Θ_v be $\theta_{\min}$ ($\theta_{\max}$) parametrized to $\theta'_{\min}$ with x ($\theta'_{\max}$ with y). Let $lb_x, ub_x, lb_y, ub_y \neq 0, 1$ be the upperbounds and the lowerbounds of x and y in $\mathcal{U}$.

Corollary 3. *An upperbound for the CD distance of u_0 from any instantiation $u \in \mathcal{U}$ is:*

$$\hat{d_0} = \ln \frac{\max\{\theta'_{\min}[lb_x], \theta'_{\min}[ub_x]\}}{\theta_{\min}} - \ln \frac{\min\{\theta'_{\max}[lb_y], \theta'_{\max}[ub_y]\}}{\theta_{\max}}$$

as derived from the single CPT closed-form [Chan, 2005].

Corollary 4. *Let $0 \leq \epsilon \leq \hat{d_0}$ and $\alpha = e^{\epsilon/2}$. Let for each $\theta \in \Theta_v$, $\alpha^{-1} \cdot \theta'[u_0] \leq \theta'[u] \leq \alpha \cdot \theta'[u_0]$. Then, $CD(u, u_0) \leq \epsilon$.*

We use the proportional parametrization (Def. 3) that has several useful properties: e.g., it is zero-preserving and minimizes the CD distance for single-CPT pBNs [Renooij, 2014].

4 Parametric Markov Chains

Parametric Markov chains are an extension of Markov chains (MCs) that allow parametric polynomials as transition labels:

Definition 5. *A* parametric Markov chain *(pMC) $\mathcal{M}$ is a tuple $(S, s_I, X, \mathcal{P})$ where S is a* finite *set of states, $s_I \in S$ is the initial state, X is a* finite *set of* real-valued *parameters, and $\mathcal{P} : S \to pDistr(S)$ is a transition function over the states.*

Example 4. *Figures 1b and 1c are examples of pMCs.*

Reachability constraints. A reachability constraint $\varphi = (T, \sim, \lambda)$ is defined over pMC $\mathcal{M} = (S, s_I, X, \mathcal{P})$ with targets $T \subseteq S$, threshold $0 \leq \lambda \leq 1$, and $\sim \in \{\leq, \geq\}$. For pMC $\mathcal{M}$ and instantiation $u : X \to \mathbb{R}$:

$$\mathcal{M}[u] \models \varphi \quad \text{if and only if} \quad \Pr_{\mathcal{M}[u]}(\Diamond T) \sim \lambda,$$

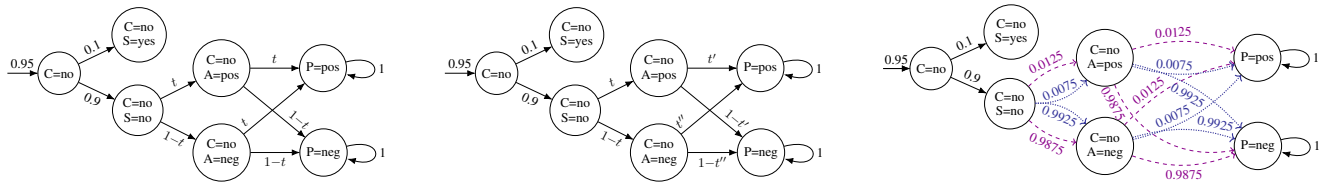
where $\Pr_{\mathcal{M}[u]}(\Diamond T)$ denotes the probability that MC $\mathcal{M}[u]$ reaches a state in T.

Example 5. *Consider the pMC $\mathcal{M}_{\mathcal{B},E}$ in Fig. 1c. Let $T = \{s_{10}\}$. Then,* $\Pr_{\mathcal{M}_{\mathcal{B},E}}(\Diamond s_{10}) = \dfrac{361}{34900 \cdot p \cdot q + 8758 \cdot q + 361}$.

Let $R \subseteq \mathbb{R}^n$ with $n = |X|$ and let

$$\mathcal{M}, R \models \varphi \quad \text{if and only if} \quad \forall u \in R.\ \mathcal{M}[u] \models \varphi.$$

We now define the parameter synthesis problems for pMC $\mathcal{M}$ and reachability constraint φ that are relevant to our setting.



(a) The original (sub-)pMC, $0.0075 \leq t \leq 0.0125$.

(b) No parameter dependencies, $0.0075 \leq t, t', t'' \leq 0.0125$.

(c) Non-parametric MDP.

Figure 3: The parameter lifting steps for the COVID-19 example with parameter dependencies (Fig. 2): (a) the original (sub-)pMC $\xrightarrow{\text{Relaxation}}$ (b) pMC without parameter dependencies $\xrightarrow{\text{Substitution}}$ (c) non-parametric MDP. Note that $0.9875 \leq 1-t, 1-t', 1-t'' \leq 0.9925$.

Definition 6 (Region partitioning)**.** *Partition region R into R_+, R_-, and $R_?$ such that:*

$$R_+ \subseteq \underbrace{\{u \in R \mid \mathcal{M}[u] \models \varphi\}}_{\text{satisfying instantiations}}, \quad R_- \subseteq \underbrace{\{u \in R \mid \mathcal{M}[u] \models \neg\varphi\}}_{\text{refuting instantiations}},$$

and $R_? = R \setminus (R_+ \cup R_-)$ with $||R_?|| \leq (1-\eta)\cdot||R||$ for some given coverage factor $0 \leq \eta \leq 1$.

The sub-region $R_?$ denotes the fragment of R that is *inconclusive* for φ. This fragment should cover at most fraction $1-\eta$ of R's volume. Region partitioning is exact if $R_? = \emptyset$.

Definition 7 (Region verification)**.** *For the region R and the specification ϕ, the problem is to check whether:*

$$\underbrace{\mathcal{M}, R \models \varphi}_{R \text{ is accepting}} \text{ or } \underbrace{\mathcal{M}, R \models \neg\varphi}_{R \text{ is rejecting}} \text{ or } \underbrace{\mathcal{M}, R \not\models \varphi \wedge \mathcal{M}, R \not\models \neg\varphi}_{R \text{ is inconclusive}}.$$

4.1 Parameter Lifting

The parameter lifting algorithm (PLA) [Quatmann *et al.*, 2016] is an abstraction technique that reduces region verification to a probabilistic model checking problem for which efficient algorithms exist [Katoen, 2016]. It first removes parameter dependencies by making all parameters unique. Then it considers for each parameter x_i only its bounds lb_{x_i} and ub_{x_i} within the region R. This yields a non-parametric Markov Decision Process (MDP).

Definition 8. *A* Markov decision process *(MDP) is a tuple $\mathcal{M} = (S, s_I, Act, \mathcal{P})$ with a finite set S of states, an initial state $s_I \in S$, a finite set Act of actions, and a (partial) transition probability function $\mathcal{P} : S \times Act \to Distr(S)$.*

While resolving the non-determinism in the obtained MDP, a trade-off between maximizing or minimizing x_i may occur. This occurs in particular when parameter x_i repeats in the outgoing transitions of multiple states, see e.g., parameter t in Fig. 3a. PLA handles the issue by a *relaxation* step that introduces intermediate parameters; see e.g., Fig. 3b, where parameter t in the outgoing edges of state $C{=}no, A{=}pos$ is replaced by t'. The relaxation step yields an over-approximation of the region verification problem.

Example 6. *Consider the pMC in Fig. 3(a) and region $t \in [0.0075, 0.0125]$. Fig. 3b shows the pMC after relaxation, e.g., parameter t in the outgoing transitions of state $C{=}no, A{=}pos$ is replaced by t'. Fig. 3c shows the MDP obtained by substituting the parameters with their extremal values. Consider e.g., state $C = no, S = no$. Its outgoing dotted (blue) edges have probability 0.0075 and $1-0.0075$ obtained from $lb(t)$. The probabilities 0.0125 and $1-0.0125$ of the dashed (purple) edges stem from $ub(t)$.*

After *substitution*, region verification reduces to a simple model-checking query that over-approximates the original higher-order polynomial, e.g., Example 3. The region accepts φ for the pMC if the resulting MDP satisfies φ for all schedulers. If all the schedulers satisfy $\neg\varphi$, the region is rejecting. If the region is neither accepting nor rejecting (i.e., inconclusive), it is partitioned into smaller subregions that are more likely to be either accepting or rejecting. Partitioning ends when at least $\eta\%$ of the region is conclusive.

4.2 Parametric Markov Chain for pBNs

To enable the use of PLA to parameter tuning of pBNs, we map pBNs onto pMCs as proposed in [Salmani and Katoen, 2021a; Salmani and Katoen, 2021b]; see the next example.

Example 7. *Consider the acyclic pMC in Fig. 1b for the pBN in Fig. 1a for the topological ordering $\varrho : C < S < A < P$. Initially, all variables are* don't care*. The initial state can evolve into C=yes and C=no. Its transition probabilities 0.05 and 0.95 come from the CPT entries of C. Generally, at "level" i of the pMC, the outgoing transitions are determined by the CPT of the $i{+}1$-st variable in ϱ.*

pBN constraints relate to reachability constraints in its pMC.

Example 8. *Consider the pBN $\mathcal{B}_1$ from Fig. 1a and the query $\Pr_{\mathcal{B}}(C = no \,|\, A = pos \wedge P = pos)$ from Example 2. Let $\mathcal{M}^{\varrho}_{\mathcal{B}}$ be the pMC in Fig. 1b. The query coincides with*

$$\frac{1 - \Pr_{\mathcal{M}^{\varrho}_{\mathcal{B}}}(\Diamond\, C = yes \vee A = neg \vee P = neg)}{1 - \Pr_{\mathcal{M}^{\varrho}_{\mathcal{B}}}(\Diamond A = neg \vee P = neg)}$$
$$= \frac{361}{34900 \cdot p \cdot q + 8758 \cdot q + 361} = f_{\mathcal{B},\phi};\ \textit{see Ex. 2.}$$

5 Region-based Minimal Change Tuning

Our approach to the parameter tuning problem for a pBN and constraint ϕ consists of two phases. The ϵ-close partitioning exploits parameter lifting on the pMC for reachability constraint φ of ϕ. We start with a rectangular region enclosing the original instantiation u_0. This region is iteratively expanded

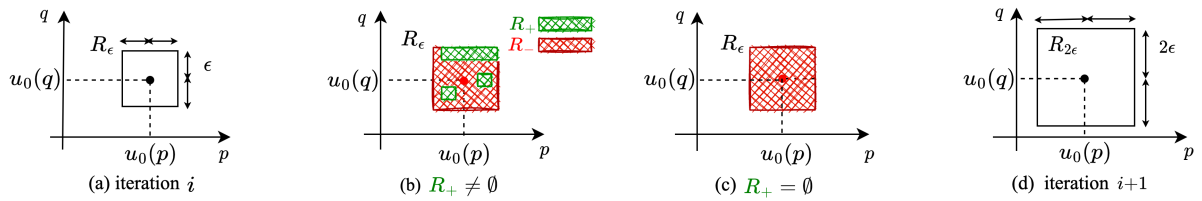


Figure 4: The iterative procedure of region-based parameter tuning for $n = 2$ and $\gamma = 1/2$.

if it rejects φ. Inconclusive regions are partitioned. Any iteration that finds some accepting subregion ends the first phase. The second phase (Alg. 2) extracts a minimal-change instantiation from the set of accepting sub-regions.

5.1 Obtaining Epsilon-Close Subregions

Alg. 1 is the main thread of our approach. Its inputs are the pBN $\mathcal{B} = (V, W, X, \Theta)$, the initial instantiation $u_0 : X \to \mathbb{R}$, and the constraint ϕ. It starts with obtaining the pMC $\mathcal{M}_\mathcal{B}$ of the pBN and reachability constraint φ of ϕ (lines 1–2). The output is either (i) u_0 if $\mathcal{M}[u_0] \models \varphi$, or (ii) the instantiation u with a minimal distance from u_0 or (iii) no instantiation if φ is infeasible, that is, in the $\eta = 100\%$ of the checked parameter space, no satisfying instantiation has been found. The hyper-parameters of the algorithm are the *coverage factor* $0 < \eta < 1$, the *region expansion factor* $0 < \gamma < 1$, and the *maximum number of iterations* $K \in \mathbb{N}$. The hyper-parameters γ and K steer the iterative procedure. They initially determine the distance bound ϵ, see line 3. The bound ϵ determines region R_ϵ, i.e., how far the region bounds deviate from u_0. PLA then verifies R_ϵ. If R_ϵ is rejecting, ϵ is extended by the factor γ^{-1} (l. 9). Figure 4 visualizes the procedure for γ=1/2 and n=2. At iteration i, (a) PLA is invoked on R_ϵ and either (b) $R_+ \neq \emptyset$ or (c) $R_+ = \emptyset$. For case (b), the iterative procedure ends and Alg. 2 is called to obtain an instantiation in R_+ that is closest to u_0. For case (c), the region is expanded by factor γ^{-1} and passed to PLA, see (d). Note that the loop terminates when the distance bound ϵ reaches its upper bound $\hat{d}_0$ (l. 2). We refer to Sec. 3 Corollary 1 and 3 for computing $\hat{d}_0$.

Region expansion schemes. Region R_ϵ is determined by factor ϵ, see line 16. The methods `makeRegion-EC` and `makeRegion-CD` detail this for each of our distance measures. For the EC distance (line 20), the parameters have an absolute deviation from u_0. We refer to Corollary 2, Sec. 3. For CD distance (lines 23 and 24), the deviation is relative to the initial value of the parameter. We refer to Corollary 4, Sec. 3. Such deviation schemes ensure ϵ-closeness of R_ϵ both for EC and CD distance, that is, all the instantiations in R_ϵ have at most distance ϵ from u_0.

5.2 Obtaining a Minimal-Distance Instantiation

Once $R_+ \neq \emptyset \subseteq R_\epsilon$ is found, it remains to find the instantiation u_+ in R_+ that is closest to u_0. Region R_+ includes infinitely many instantiations, yet minimal-distance instantiation is computable in $O(k{\cdot}n)$ for $n = |X|$ and the regions $R_+ = \{R_{+,1}, \cdots, R_{+,k}\}$: (i) PLA ensures that the subregions $R_{+,1}$ to $R_{+,k}$ are *rectangular* and *mutually disjoint*. (ii) Due to the triangle equality of the distance measure, a minimum-distance instantiation can be picked from the bounded region. Alg. 2 simplifies finding u_+. The idea (l. 4–14) is to pick $u_{+,i}$ from $R_{+,i}$ that has minimal distance from u_0. Let lb_{x_j} be the lower bound for parameter x_j in the region $R_{+,i}$ and ub_{x_j} be its upper bound. We set the value for each parameter based on three possible cases. (1) If $u_0(x_j) \in [lb_{x_j}, ub_{x_j}]$, we set $u_{+,i}(x_j) = u_0(x_j)$. This yields the least distance, i.e., 0 in the dimension of x_j, see Fig. 5

Algorithm 1: Minimal change tuning

```
1  M_B ← computeMC(B)
2  ϱ ← reachSpec(φ)
3  ε_0 ← d̂_0 · γ^{K−1}
4  ε ← ε_0
5  function minChgTuning(M_B, u_0, φ, η):
6      while ε ≤ d̂_0 do
7          R_+ ← ε-partitioning(M_B, φ, ε, η)
8          if R_+ = ∅ then
9              ε ← γ^{−1} · ε
10         else
11             return getMinDistInst(R_+, u_0)
12         end if
13     end while
14     return Infeasible
15 function ε-partitioning(M_B, φ, ε, η):
16     R_ε ← makeRegion-d(u_0, ε)
17     R_+, R_−, R_? ← PLA(M_B, R_ε, φ, η)
18     return R_+
19 function makeRegion-EC(u_0, ε):
20     R_ε ← ×_{x_j ∈ X} [u_0(x_j) − ε/n, u_0(x_j) + ε/n]
21     return R_ε
22 function makeRegion-CD(u_0, ε):
23     α ← e^{ε/2}
24     R_ε ← ×_{x_j ∈ X} [u_0(x_j) · α^{−1}, u_0(x_j) · α]
25     return R_ε
```

Algorithm 2: R_+-minimal distance instantiation

```
1  function getMinDistInst(R_+, u_0):
2      // R_+ = {R_{+,1}, ··· , R_{+,k}}
3      U_+ ← ∅
4      for i ← 1 to k do
5          // R_{+,i} = ×_{x_j ∈ X} [lb_{x_j}, ub_{x_j}]
6          for x_j ∈ X do
7              if lb_{x_j} < u_0(x_j) < ub_{x_j} then
8                  u_{+,i}(x_j) = u_0(x_j)
9              else
10                 if u_0(x_j) < lb_{x_j} < ub_{x_j} then
11                     u_{+,i}(x_j) = lb_{x_j}
12                 else
13                     u_{+,i}(x_j) = ub_{x_j}
14             U_+ ← u_{+,i}
15     return argmin_{u_+ ∈ U_+} d(u_+, u_0)
```

(left). (2) The second case is $u_0(x_j) < lb_{x_j}$, see Fig. 5 (middle). Then the lower bound lb_{x_j} has the least distance from $u_0(x_j)$ in the dimension x_j, i.e., $|lb_{x_j} - u_0(x_j)| < |a - u_0(x_j)|$ for every value $a \in [lb_{x_j}, ub_{x_j}]$ and similarly for CD distance, $\ln \frac{lb_{x_j}}{u_0(x_j)} < \ln \frac{a}{u_0(x_j)}$. In this case, we set $u_{+,i}(x_j) = lb_{x_j}$ to minimize the distance in dimension x_j. (3) By symmetry, for $u_0(x_j) > ub_{x_j}$, we set $u_{+,i}(x_j) = ub_{x_j}$; see Fig. 5 (right). It remains to compute the distance for each candidate in $U_+ = \{u_{+,1}, \cdots, u_{+,k}\}$ and pick u_+ closest to u_0 (l. 15).

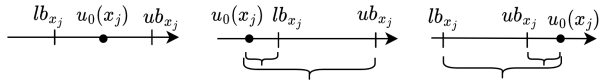


Figure 5: Obtaining the minimal-change value for parameter $x \in X$.

5.3 Example

We apply the entire algorithm on the pBN in Fig. 1a, constraint $\phi = \Pr(\text{C=no} \mid \text{A=pos} \wedge \text{P=pos}) \leq 0.009$, $u_0 : p \to 0.72, q \to 0.95$, and η=0.99. The pMC $\mathcal{M}_{\mathcal{B},E}$ is given in Fig. 1c and the reachability constraint is $\varphi : \Pr(\Diamond s_{10}) \leq 0.009$. We take $\hat{d}_0 = \sqrt{n}$, $\gamma = 1/2$, and $\epsilon_0 = 1/16$.

1. The initial region R_ϵ is obtained by $\epsilon = 1/8$:

   $$[0.72 - 1/8\sqrt{2}, 0.72 + 1/8\sqrt{2}] \times [0.95 - 1/8\sqrt{2}, 0.95 + 1/8\sqrt{2}].$$

2. Running PLA on R_ϵ gives no sub-region accepting φ.
3. Expanding the region by the factor $\gamma^{-1} = 2$ yields:

   $$[0.72 - 1/4\sqrt{2}, 0.72 + 1/4\sqrt{2}] \times [0.95 - 1/4\sqrt{2}, 0.95 + 1/4\sqrt{2}].$$

4. Running PLA gives 12 regions accepting φ, e.g., $R_{+,1} : [0.92075, 0.960375] \times [0.97475, 1]$ and $R_{+,2} : [0.960375, 0.97028175] \times [0.9431875, 0.9495]$.
5. Running Alg. 2 on $R_{+,1}$ through $R_{+,12}$ gives minimal distance candidates $u_{+,1}$ through $u_{+,12}$; e.g., $u_{+,2}(p) = lb_p = 0.960375$ as $u_0(p) < lb_p$ and $u_{+,2}(q) = ub_q = 0.9495$ as $ub_q < u_0(q)$; see Fig. 5.
6. Running Alg. 2 using the distance measure of choice—here: EC distance—for the candidates U_+ returns u_+ closest to u_0: $u_+(p) = 0.92075$ and $u_+(q) = 0.97475$; distance 0.040913125.

5.4 Discussion

Let us shortly discuss the hyper-parameters: η, γ, and K. The hyper-parameter η is the coverage factor for the region partitioning. Our region partitioning algorithm, i.e., parameter lifting (line 17 Alg. 1) works based on this factor: the procedure stops when $R_?$ is at most $(1 - \eta) \cdot ||R||$ of the entire region R, e.g., for $\eta = 0.99$, it continues until at least 99% of R is either rejecting or accepting and at most 1% is unknown. This relates η to the approximation bounds of our algorithm, see Problem statement, Def. 4. Intuitively speaking, the coverage factor η means that if the algorithm provides a minimal-distance value $D(u, u_0) = d$, then with the probability $1 - \eta$ there may exist a smaller value distance than d that works too but has not been found. One can thus consider η as the confidence factor for the minimality of the results.

The hyper-parameter γ specifies the factor by which the region is expanded at each iteration, see Line 9, Alg. 1. The hyper-parameters γ and K specify the size of the initial region, see Lines 3 and 16 of Alg. 1. Our experiments (not detailed in the paper) with $\gamma = \{0.2, 0.5, 0.8\}$ and $K = \{2, \cdots, 12\}$ reveal that $\gamma = 1/2$ and $K = 6$ gave the best balance. Large values for $\gamma\,(0.8)$ and small K lead to unnecessary computations in the initial iteration for the simple cases i.e., when small perturbations of the parameters make the constraint satisfied. Small values for $\gamma\,(0.2)$ lead to large regions in the next iterations due to the expansion by γ^{-1}.

6 Experimental Evaluation

We empirically evaluated our approach using a prototypical realization on top of the probabilistic model checker Storm [Hensel *et al.*, 2022] (version 1.7.0). As baseline, we used the latest version (10.4) of *Bayesserver*[2], a commercial BN analysis tool that features sensitivity analysis and parameter tuning for pBNs. It supports pBNs with a single parameter only. We parametrized benchmarks from `bnlearn` repository and defined different constraints. We (i) parametrized the CPTs of the parents (and grandparents) of the evidence nodes, and (ii) used the *SamIam* tool [3] to pick the CPT entries Θ_{modif} most relevant to the constraint. To get well-formed BNs, we used the linear proportional parametrization, see Def. 3. In all experiments, we picked evidences from the last nodes in the topological order; they have a long dependency path to the BN roots. This selection reflects the worst-case in [Salmani and Katoen, 2020]. We took $\gamma = 1/2$ and $K = 6$ for our experiments, see Sec. 5.4. We conducted all our experiments on a 2.3 GHz Intel Core i5 processor with 16 GB RAM.

RQ1: Comparison to Bayesserver. We used small, medium, and large BNs from the `bnlearn` repository. For each BN, we parameterized one distribution in a single CPT to align with the restrictions from Bayesserver. Figure 6a indicates the results, comparing the tuning times (in sec) of Bayesserver (x-axis) to those of our implementation (y-axis). The latter includes the time for region refinements by PLA. For each pair (pBN, constraint), we did experiments for coverage factors $\eta = 1 - 10^{-i}$ for $i \in \{1, \cdots, 6\}$. The influence of η on the exactness of our results is addressed under RQ2.

Findings. Storm outperforms Bayesserver for several benchmarks (`cancer`, `sachs`, `win95pts`, and multiple instances of `hailfinder`), whereas Bayesserver is faster by about an order of magnitude for the other pBNs, such as `hepar2`. We noted that Bayesserver exploits specific methods for one-way sensitivity analysis and relies on the linear form of the sensitivity function. These techniques are very efficient yet not applicable to pBNs with multiple parameters in multiple CPTs. For our experiments on those subclasses, see RQ4. Such subclasses are not supported by Bayesserver and—to the best of our knowledge—not any existing BN tool. This applies e.g., also to Bayesfusion which considers only the change of a single parameter at a time, and SamIam which is limited to the single parameter and single-CPT.

[2] https://www.bayesserver.com/

[3] http://reasoning.cs.ucla.edu/samiam/

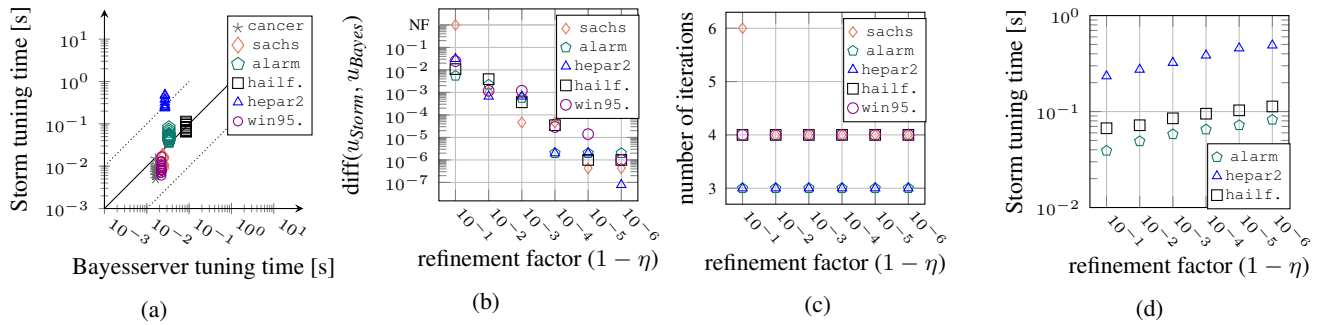


Figure 6: The plots (a), (b): Storm vs. Bayesserver, the tuning time and the instantiation closeness; see RQ1. The plots (b), (c), (d): the effect of the coverage factor (η) on the tightness of the distance, the number of iterations, and the tuning time; see RQ2. Corner case: the number of iterations only changes when a satisfying instantiation was *not found* (NF) with a certain coverage; see `sachs` at η=0.9 and η=0.99.

RQ2: Sensitivity to the coverage factor η. For each pBN and constraint, we decreased (the refinement factor) $1-\eta$ in a step-wise manner by a factor 10^{-1}. To quantify the tightness of our results, we measured how our approximately-close instantiation, denoted u_{Storm}, differs from the absolute minimum-distance from Bayesserver, denoted u_{Bayes}. Figure 6b (log-log scale) plots the tightness of the results $|u_{Storm} - u_{Bayes}|$ (y-axis) against the refinement factor (x-axis). Figures 6c and 6d indicate the number of iterations and the tuning time (log scale, seconds) for each refinement factor.

Findings. (I) Mostly, $1-\eta$ bounds the difference between our approximately-close solution and Bayesserver's solution. For e.g., $\eta = 1-10^{-4}$, the difference is at most 10^{-4}. Recall from Def. 6 that the value of $1-\eta$ bounds the size of the unknown regions in the region partitioning procedure. This explains why $|u_{Storm} - u_{Bayes}|$ is bounded by $1-\eta$. (II) On increasing the coverage, the difference to the minimum distance rapidly decreases. (III) The computation time moderately increases on increasing the coverage, but the number of iterations was mostly unaffected. At a higher coverage factor, the region partitioning is more fine-granular possibly yielding more accepting regions to analyze. Therefore the computation becomes more expensive. (IV) The timing is not correlated to the number of iterations. This is because the ϵ-close iterations before the last iteration are often completed by a single region verification and are very fast. Similar observations have been made for $n>1$ parameters; see RQ4.

RQ3: Sensitivity to the threshold λ. We varied the constraint's threshold (λ) by steps of $1/20$ for the benchmarks `alarm` `hepar2`, and `hailfinder` with n=1 and η=99.99999%. Figures 7a, 7b, and 7c display the outcomes with the x-axis indicating the threshold and the y-axis indicating the tuning time (in seconds), the distance (log-scale), and the number of iterations.

Findings. For small perturbations the original BNs ($\mathcal{B}[u_0]$) may already satisfy the constraints. In general, for the constraints close to the original probability $\Pr_{\mathcal{B}[u_0],\phi}$, the number of iterations is low, the distance is naturally small, and the minimal-change tuning is completed fast. This is because region refinement starts with small regions in the close vicinity of the original values of the parameters without the need to analyze larger regions. By strengthening the threshold, the possibly satisfying regions get further away from u_0. Thus the distance, the number of iterations, and sometimes the tuning time grow. For very large perturbations the constraint may become infeasible with the confidence factor $\eta = 100\%$. Similar findings are valid for $n > 1$ parameters; see RQ4.

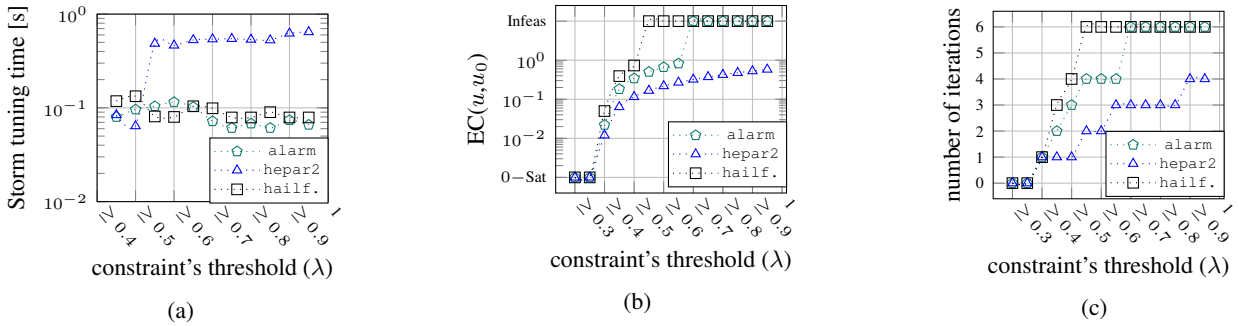


Figure 7: The effect of the constraint restrictiveness (the value of the threshold) on ϵ-bounded tuning: (a) on the tuning time, (b) on the distance, (c) and on the number of iterations; see RQ3. Corner cases: for the constraints with thresholds before 0.4, the original BN satisfies the constraints. For `hailfinder` with the thresholds $\lambda \geq 0.55$ and `alarm` with the thresholds $\lambda \geq 0.7$, the constraints become infeasible.

pBN info		constraint	setting	results		
pCPT	par	thresh.	cover.	EC	iter	**t**(s)
1	2	≥ 0.115	99%	0.8365285878	5	4.200
1	2	≥ 0.25	100%	Infeasible	6	3.750
3	4	≥ 0.115	90%	0.1350477346	2	5.210
3	4	≥ 0.25	80%	0.4479118217	4	5.210
3	4	≥ 0.25	90%	0.4238956239	4	30.82
3	4	≥ 0.25	95%	0.4098399078	4	1128
3	4	≥ 0.25	99%	-	-	TO
4	8	≥ 0.115	80%	0.8805679985	2	4.496
4	8	≥ 0.11	90%	0.1350477346	2	15.91
4	8	≥ 0.30	60%	0.8805679985	6	809.7
4	8	≥ 0.35	99.99%	Infeasible	6	3.757
4	$8^*(2)(2)$	≥ 0.115	80%	0.013153540	1	180.6
4	16	≥ 0.35	10%	-	1	MO

pBN info		constraint	setting	results		
pCPT	par	thresh.	cover.	EC	iter	**t**(s)
1	2	≤ 0.30	85%	1.138343094	5	2.090
1	2	≤ 0.30	99%	1.091371208	5	2.134
1	2	≤ 0.30	99.99%	1.090202067	5	302.1
1	4	≤ 0.30	70%	0.909491616	5	5.701
1	4	≤ 0.30	85%	0.890217670	5	142.1
1	4	≤ 0.30	90%	-	-	TO
1	4	≤ 0.20	100%	Infeasible	6	1.869
1	$4^*(3)$	≤ 0.30	99%	0.472406250	5	2.731
1	8	≤ 0.30	20%	0.5628159113	5	534.6
1	8	≤ 0.25	20%	0.9405485102	5	216.6
1	8	≤ 0.20	20%	1.010395962	6	305.9
1	$8^*(3)$	≤ 0.70	20%	0.118671026	5	608.2
1	8	≤ 0.001	100%	Infeasible	6	1.886

Table 1: The scalability of our approach to pBNs with multiple parameters, detailed results for (left) `win95pts` and (right) `alarm`.

RQ4: Scaling the number of parameters. We took the `win95pts` and `alarm` benchmarks and parameterized them in multiple ways. Their pMCs have $9,947$ and $1,118$ states and $17,948$ and $3,147$ transitions respectively. The set of parameters for each pBN is including (and doubles the number of) parameters in the previous pBN. Table 1(left) and (right) list the results for `win95pts` and `alarm`. We list for each pBN, the number of affected CPTs, the number of parameters, the threshold λ, and the coverage η. E.g., `win95pts` with par=8 has 8 parameters occurring in 4 CPTs. The columns EC, iter, and $t(s)$ report the EC distance, the number of iterations, and the total time in seconds (incl. model building time, time for region refinement, and tuning time) respectively. TO and MO indicate time-out (30 minutes) and memory-out (>16 GB).

Findings. (I) Approximately-close parameter tuning is feasible for pBNs with up to 8 parameters. *This is significantly higher than the state of the art—one parameter.* As the number of sub-regions by PLA grows exponentially, treating more parameters is practically infeasible. (II) More parameters often ease finding satisfying instantiations. E.g., the threshold ≥ 0.25 is unsatisfiable for `win95pts` with n=2, but is satisfied with n=4. (III) The results for multiple parameter pBNs confirm the findings for RQ2 and RQ3; see the rows for each pBN instance with (i) varied coverage and (ii) varied threshold. (IV) The unsatisfiability of a constraint can be computed fast (with 100% confidence), regardless of the number of parameters, see e.g., `alarm` with 8 parameters for the constraint ≤ 0.001. For infeasibility, a single verification suffices; no partitioning is needed.

RQ5: Handling pBNs with parameter dependencies. Parameter lifting algorithm (Section 4.1) enables handling models with parameter dependencies, see e.g., Example 3: the parameters are allowed in multiple local distributions and the pBN sensitivity function is of higher degree. We extended our experiments to such cases for `win95pts` and `alarm`: we parameterized the entries $\theta_1 \cdots \theta_k \in \Theta_{modif}$ over the same parameter $x \in X$ when the original values of $\theta_1 \cdots \theta_k$ were the same in the original BN. The eleventh row in Table 1 (left) and the eighth and eleventh rows in Table 1 (right) correspond to such cases. The term $8^*(2)(2)$ e.g., denotes that out of the 8 parameters, 2 parameters repeatedly occurred in two distinct distributions. The term $8^*(3)$ denotes that out of the 8 parameters, one was occurring in 3 distinct distributions.

Findings. (I) Our method is applicable to pBNs with parameter dependencies where the sensitivity function is of a higher degree. (II) For the same coverage factor, the pBNs with parameter dependency are more expensive to analyze. See e.g., the two rows for `win95pts` with 8 parameters, threshold ≥ 0.115, and the coverage 80%. This is due to more complex sensitivity functions that give a higher number of sub-regions to verify. (III) The pBNs with parameter dependency yielded notably smaller distances.

7 Epilogue

Related work. Kwisthout and van der Gaag [2008] studied the theoretical complexity of tuning problems. Renooij [2014] studied the properties of the co-variation schemes for BN tuning. She shows that the linear proportional scheme optimizes the CD distance for single-CPT pBNs. Similar are the studies by Bolt and van der Gaag [2015; 2017] that consider tuning heuristics for distance optimizations. Peng and Ding [2005] propose an iterative proportional fitting procedure (IPFP) to minimize the KL-divergence distance for a set of constraints. The method does not scale to large networks. Santos *et al.* [2013] exploits linear programming for BN parameter tuning. Yak- aboski and Santos [2018] consider a new distance measure for parameter learning and tuning of BNs. Leonelli [2019] considers nonlinear sensitivity functions, yet only for single parameter pBNs and Ballester-Ripoll and Leonelli [2022] efficiently compute the derivatives of sensitivity functions to select the most relevant parameters to a query, yet they limit to single parameter variation.

Conclusion. A novel algorithm for parameter tuning in Bayesian networks is presented and experimentally evaluated. Whereas existing algorithms come with severe restrictions—single parameters and/or linear functions—our approach is applicable to multiple (in practice about 8) parameters, large BNs (up to 100 variables), and polynomial functions. Future work includes considering balanced tuning heuristic [Bolt and van der Gaag, 2017] and using monotonicity of parameters [Spel *et al.*, 2019].

Acknowledgements

This research was funded by the ERC AdG Projekt FRAPPANT (Grant Nr. 787914). We kindly thank Alexandra Ivanova for her implementation efforts and Tim Quatmann for the fruitful discussions.

References

[Ballester-Ripoll and Leonelli, 2022] Rafael Ballester-Ripoll and Manuele Leonelli. You only derive once (YODO): automatic differentiation for efficient sensitivity analysis in Bayesian networks. In *PGM*, volume 186 of *Proceedings of Machine Learning Research*, pages 169–180. PMLR, 2022.

[Barreiro *et al.*, 2021] Pablo Barreiro, Jesús San-Román, Maria del Mar Carretero, and Francisco Javier Candel. Infection and infectivity: Utility of rapid antigen tests for the diagnosis of covid-19. *Revista Española de Quimioterapia*, 34(Suppl1):46, 2021.

[Bolt and van der Gaag, 2015] Janneke H. Bolt and Linda C. van der Gaag. Balanced tuning of multi-dimensional Bayesian network classifiers. In *ECSQARU*, volume 9161 of *Lecture Notes in Computer Science*, pages 210–220. Springer, 2015.

[Bolt and van der Gaag, 2017] Janneke H. Bolt and Linda C. van der Gaag. Balanced sensitivity functions for tuning multi-dimensional Bayesian network classifiers. *Int. J. Approx. Reason.*, 80:361–376, 2017.

[Castillo *et al.*, 1997] Enrique F. Castillo, José Manuel Gutiérrez, and Ali S. Hadi. Sensitivity analysis in discrete Bayesian networks. *IEEE Trans. Syst. Man Cybern. Part A*, 27(4):412–423, 1997.

[Chan and Darwiche, 2004] Hei Chan and Adnan Darwiche. Sensitivity analysis in Bayesian networks: From single to multiple parameters. In *UAI*, pages 67–75. AUAI Press, 2004.

[Chan and Darwiche, 2005] Hei Chan and Adnan Darwiche. A distance measure for bounding probabilistic belief change. *Int. J. Approx. Reason.*, 38(2):149–174, 2005.

[Chan, 2005] Hei Chan. *Sensitivity analysis of probabilistic graphical models*. University of California, Los Angeles, 2005.

[Coupé and van der Gaag, 2002] Veerle M. H. Coupé and Linda C. van der Gaag. Properties of sensitivity analysis of Bayesian belief networks. *Ann. Math. Artif. Intell.*, 36(4):323–356, 2002.

[Cubuktepe *et al.*, 2018] Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, and Ufuk Topcu. Synthesis in pMDPs: A tale of 1001 parameters. In *ATVA*, volume 11138 of *Lecture Notes in Computer Science*, pages 160–176. Springer, 2018.

[Cubuktepe *et al.*, 2022] Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, and Ufuk Topcu. Convex optimization for parameter synthesis in MDPs. *IEEE Trans. Autom. Control.*, 67(12):6333–6348, 2022.

[Darwiche, 2009] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.

[Dinnes *et al.*, 2022] Jacqueline Dinnes, Pawana Sharma, Sarah Berhane, Susanna S van Wyk, Nicholas Nyaaba, Julie Domen, Melissa Taylor, Jane Cunningham, Clare Davenport, Sabine Dittrich, et al. Rapid, point-of-care antigen tests for diagnosis of sars-cov-2 infection. *Cochrane Database of Systematic Reviews*, 7, 2022.

[Heck *et al.*, 2022] Linus Heck, Jip Spel, Sebastian Junges, Joshua Moerman, and Joost-Pieter Katoen. Gradient-descent for randomized controllers under partial observability. In *VMCAI*, volume 13182 of *Lecture Notes in Computer Science*, pages 127–150. Springer, 2022.

[Hensel *et al.*, 2022] Christian Hensel, Sebastian Junges, Joost-Pieter Katoen, Tim Quatmann, and Matthias Volk. The probabilistic model checker storm. *Int. J. Softw. Tools Technol. Transf.*, 24(4):589–610, 2022.

[Katoen, 2016] Joost-Pieter Katoen. The probabilistic model checking landscape. In *LICS*, pages 31–45. ACM, 2016.

[Kisa *et al.*, 2014] Doga Kisa, Guy Van den Broeck, Arthur Choi, and Adnan Darwiche. Probabilistic sentential decision diagrams. In *KR*. AAAI Press, 2014.

[Kwisthout and van der Gaag, 2008] Johan Kwisthout and Linda C. van der Gaag. The computational complexity of sensitivity analysis and parameter tuning. In *UAI*, pages 349–356. AUAI Press, 2008.

[Leonelli, 2019] Manuele Leonelli. Sensitivity analysis beyond linearity. *Int. J. Approx. Reason.*, 113:106–118, 2019.

[Nishiura *et al.*, 2020] Hiroshi Nishiura, Tetsuro Kobayashi, Takeshi Miyama, Ayako Suzuki, Sung-mok Jung, Katsuma Hayashi, Ryo Kinoshita, Yichi Yang, Baoyin Yuan, Andrei R Akhmetzhanov, et al. Estimation of the asymptomatic ratio of novel coronavirus infections (covid-19). *Int. Journal of Infectious Diseases*, 94:154–155, 2020.

[Pearl, 1988] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[Peng and Ding, 2005] Yun Peng and Zhongli Ding. Modifying Bayesian networks by probability constraints. In *UAI*, pages 459–466. AUAI Press, 2005.

[Quatmann *et al.*, 2016] Tim Quatmann, Christian Dehnert, Nils Jansen, Sebastian Junges, and Joost-Pieter Katoen. Parameter synthesis for Markov models: Faster than ever. In *ATVA*, volume 9938 of *Lecture Notes in Computer Science*, pages 50–67, 2016.

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

[Salmani and Katoen, 2020] Bahare Salmani and Joost-Pieter Katoen. Bayesian inference by symbolic model checking. In *QEST*, volume 12289 of *Lecture Notes in Computer Science*, pages 115–133. Springer, 2020.

[Salmani and Katoen, 2021a] Bahare Salmani and Joost-Pieter Katoen. Fine-tuning the odds in Bayesian networks. In *ECSQARU*, volume 12897 of *Lecture Notes in Computer Science*, pages 268–283. Springer, 2021.

[Salmani and Katoen, 2021b] Bahare Salmani and Joost-Pieter Katoen. Fine-tuning the odds in Bayesian networks. *CoRR*, abs/2105.14371, 2021.

[Santos *et al.*, 2013] Eugene Santos, Qi Gu, and Eunice E. Santos. Bayesian knowledge base tuning. *Int. J. Approx. Reason.*, 54(8):1000–1012, 2013.

[Spel *et al.*, 2019] Jip Spel, Sebastian Junges, and Joost-Pieter Katoen. Are parametric Markov chains monotonic? In *ATVA*, volume 11781 of *Lecture Notes in Computer Science*, pages 479–496. Springer, 2019.

[Yakaboski and Santos, 2018] Chase Yakaboski and Eugene Santos. Bayesian knowledge base distance-based tuning. In *WI*, pages 64–72. IEEE Computer Society, 2018.