Journal of Artificial Intelligence Research 53 (2015) 699-720 Submitted 03/15; published 08/15

Tree-Width and the Computational Complexity of MAP Approximations in Bayesian Networks

Johan Kwisthout J.KWISTHOUT@DONDERS.RU.NL
Radboud University Nijmegen
Donders Institute for Brain, Cognition and Behaviour
Montessorilaan 3, 6525 HR Nijmegen, The Netherlands

Abstract

The problem of finding the most probable explanation to a designated set of variables given partial evidence (the MAP problem) is a notoriously intractable problem in Bayesian networks, both to compute exactly and to approximate. It is known, both from theoretical considerations and from practical experience, that low tree-width is typically an essential prerequisite to efficient exact computations in Bayesian networks. In this paper we investigate whether the same holds for approximating MAP. We define four notions of approximating MAP (by value, structure, rank, and expectation) and argue that all of them are intractable in general. We prove that efficient value-approximations, structure-approximations, and rank-approximations of MAP instances with high tree-width will violate the Exponential Time Hypothesis. In contrast, we show that MAP can sometimes be efficiently expectation-approximated, even in instances with high tree-width, if the most probable explanation has a high probability. We introduce the complexity class FERT, analogous to the class FPT, to capture this notion of fixed-parameter expectation-approximability. We suggest a road-map to future research that yields fixed-parameter tractable results for expectation-approximate MAP, even in graphs with high tree-width.

1. Introduction

One of the most important computational problems in Bayesian networks is the MAP problem, i.e., the problem of finding the joint value assignment to a designated set of variables (the MAP variables) with the maximum posterior probability, given partial observation of the remaining variables. The MAP problem is notably intractable; as it is $\mathsf{NP}^{\mathsf{PP}}$-hard, it is strictly harder (given usual assumptions in computational complexity theory) than the PP-hard inference problem (Park & Darwiche, 2004). In a sense, it can be seen as combining an *optimization* problem with an *inference* problem, both of which potentially contribute to the problem's complexity (Park & Darwiche, 2004, p. 113). Even when all variables in the network are binary and the network has the (very restricted) polytree topology, MAP remains NP-hard (De Campos, 2011). Only when both the optimization *and* the inference part of the problem can be computed tractably (for example, if both the tree-width of the network is small, the cardinality of the variables is low, *and* the most probable joint value assignment has a high probability) MAP can be computed tractably (Kwisthout, 2011). It is known that, for arbitrary probability distributions and under the assumption of the Exponential Time Hypothesis, low tree-width of the moralized graph of a Bayesian network is a necessary condition for the Inference problem in Bayesian networks to be tractable

(Kwisthout, Bodlaender, & van der Gaag, 2010); this result can easily be extended to MAP, as we will show in Section 4.

MAP is also intractable to approximate (Abdelbar & Hedetniemi, 1998; Kwisthout, 2011, 2013; Park & Darwiche, 2004). While it is obviously the case that a particular instance to the MAP problem can be approximated efficiently when it can be computed exactly efficiently, it is as yet unclear whether approximate MAP computations can be rendered tractable under *different* conditions than exact MAP computations. Crucial here is the question *what we mean* with a statement as ‘algorithm A approximates the MAP problem’. Typically, in computer science, approximation algorithms guarantee that the output of the algorithm has a value that is within some bound of the value of the optimal solution. For example, the canonical approximation algorithm to the Vertex Cover problem selects an edge at random, puts both endpoints in the vertex cover, and removes these nodes from the instance. This algorithm is guaranteed to get a solution that has at most twice the number of nodes in the vertex cover as the optimal vertex set. However, typical Bayesian approximation algorithms have no such guarantee; in contrast, they may converge to the optimal value given enough time (such as the Metropolis-Hastings algorithm), or they may find an optimal solution with a high probability of success (such as repeated local search strategies).

In this paper we assess four different notions of approximation relevant for the MAP problem; in particular value-approximation, structure-approximation, rank-approximation, and expectation-approximation of MAP. After introducing notation and providing some preliminaries (Section 2), we show that each of these approximations is intractable under the assumption that $\mathsf{P} \neq \mathsf{NP}$, respectively $\mathsf{NP} \not\subseteq \mathsf{BPP}$ (Section 3). Building on the result by Kwisthout et al. (2010) we show in Section 4 that bounded tree-width is indeed a necessary condition for efficient value-approximation, structure-approximation, and rank-approximation of MAP. In Section 5 we argue that this need not be the case for expectation-approximation. We introduce the parameterized complexity classes FERT (Fixed Error Randomized Tractable) and FPERT (Fixed Parameter and Error Randomized Tractable) as natural extensions to the class FPT. We introduce a MAP variant with some additional constraints and we show that Constrained-MAP is intractable (PP-hard) in general; however, Constrained-MAP is in FERT when parameterized by the probability of the most probable explanation, even when tree-width is high. We conclude the paper in Section 6.

2. Preliminaries

In this section, we introduce our notational conventions and provide some preliminaries on Bayesian networks, graph theory, and complexity theory; in particular definitions of the MAP problem, tree-width, parameterized complexity theory, and the Exponential Time Hypothesis. For a more thorough discussion of these concepts, the reader is referred to textbooks such as those by Darwiche (2009), Arora and Barak (2009), and Downey and Fellows (1999).

2.1 Bayesian Networks

A Bayesian network $\mathcal{B} = (\mathbf{G}_\mathcal{B}, \Pr)$ is a graphical structure that succinctly represents a joint probability distribution over a set of stochastic variables. $\mathcal{B}$ includes a directed acyclic graph $\mathbf{G}_\mathcal{B} = (\mathbf{V}, \mathbf{A})$, where $\mathbf{V}$ models (in a one-to-one mapping) the stochastic variables and $\mathbf{A}$ models the conditional (in)dependences between them, and a set of parameter probabilities Pr in the form of conditional probability tables (CPTs), capturing the strengths of the relationships between the variables. The network models a joint probability distribution $\Pr(\mathbf{V}) = \prod_{i=1}^{n} \Pr(V_i \mid \pi(V_i))$ over its variables; here, $\pi(V_i)$ denotes the parents of V_i in $\mathbf{G}_\mathcal{B}$. As notational convention we will use upper case letters to denote individual nodes in the network, upper case bold letters to denote sets of nodes, lower case letters to denote value assignments to nodes, and lower case bold letters to denote joint value assignments to sets of nodes. We will use 'node' and 'variable' interchangeably.

One of the key computational problems in Bayesian networks is the problem to find the most probable explanation for a set of observations, i.e., a joint value assignment to a designated set of variables (the explanation set) that has maximum posterior probability given the observed variables (the joint value assignment to the evidence set) in the network. If the network is bi-partitioned into explanation variables and evidence variables this problem is known as Most Probable Explanation (MPE). The more general problem, where the network also includes variables that are neither observed nor to be explained (referred to as intermediate variables) is known as (Partial or Marginal) MAP. This problem is typically defined formally as follows:

MAP
Instance: A Bayesian network $\mathcal{B} = (\mathbf{G}_\mathcal{B}, \Pr)$, where $\mathbf{V}$ is partitioned into a set of evidence nodes $\mathbf{E}$ with a joint value assignment $\mathbf{e}$, a set of intermediate nodes $\mathbf{I}$, and an explanation set $\mathbf{H}$.
Output: A joint value assignment $\mathbf{h}$ to $\mathbf{H}$ such that for all joint value assignments $\mathbf{h}'$ to $\mathbf{H}$, $\Pr(\mathbf{h} \mid \mathbf{e}) \geq \Pr(\mathbf{h}' \mid \mathbf{e})$.

In the remainder, we use the following definitions. For an arbitrary MAP instance $\{\mathcal{B}, \mathbf{H}, \mathbf{E}, \mathbf{I}, \mathbf{e}\}$, let $cansol_\mathcal{B}$ refer to the set of candidate solutions to $\{\mathcal{B}, \mathbf{H}, \mathbf{E}, \mathbf{I}, \mathbf{e}\}$, with $optsol_\mathcal{B} \in cansol_\mathcal{B}$ denoting the *optimal* solution (or, in case of a draw, one of the optimal solutions) to the MAP instance. When $cansol_\mathcal{B}$ is ordered according to the probability of the candidate solutions (breaking ties between candidate solutions with the same probability arbitrarily), then $optsol_\mathcal{B}^{1\ldots m}$ refers to the set of the first m elements in $cansol_\mathcal{B}$, viz. the m most probable solutions to the MAP instance. For a particular notion of approximation, we refer to an (unspecified) approximate solution as $approxsol_\mathcal{B} \in cansol_\mathcal{B}$.

2.2 Tree-Width

An important structural property of a Bayesian network $\mathcal{B}$ is its *tree-width*, which can be defined as the minimum width of any tree-decomposition (or equivalently, the minimal size of the largest clique in any triangulation) of the moralization $\mathbf{G}_\mathcal{B}^\mathbf{M}$ of the network. Tree-width plays an important role in the complexity analysis of Bayesian networks, as many otherwise intractable computational problems can be rendered tractable, provided that the tree-width of the network is small. The moralization (or 'moralized graph') $\mathbf{G}_\mathcal{B}^\mathbf{M}$ is the undirected graph that is obtained from $\mathbf{G}_{\mathcal{B}}$ by adding arcs so as to connect all pairs of parents of a variable, and then dropping all directions. A triangulation of $\mathbf{G}^{\mathbf{M}}_{\mathcal{B}}$ is any chordal graph $\mathbf{G_T}$ that embeds $\mathbf{G}^{\mathbf{M}}_{\mathcal{B}}$ as a subgraph. A chordal graph is a graph that does not include loops of more than three variables without any pair being adjacent.

A tree-decomposition (Robertson & Seymour, 1986) of a triangulation $\mathbf{G_T}$ now is a tree $\mathbf{T_G}$ such that each node $\mathbf{X_i}$ in $\mathbf{T_G}$ is a bag of nodes which constitute a clique in $\mathbf{G_T}$; and for every i, j, k, if $\mathbf{X_j}$ lies on the path from $\mathbf{X_i}$ to $\mathbf{X_k}$ in $\mathbf{T_G}$, then $\mathbf{X_i} \cap \mathbf{X_k} \subseteq \mathbf{X_j}$. In the context of Bayesian networks, this tree-decomposition is often referred to as the *junction tree* or *clique tree* of $\mathcal{B}$. The width of the tree-decomposition $\mathbf{T_G}$ of the graph $\mathbf{G_T}$ is defined as the size of the largest bag in $\mathbf{T_G}$ minus 1, i.e., $\max_i(|\mathbf{X_i}| - 1)$. The tree-width tw of a Bayesian network $\mathcal{B}$ now is the minimum width over all possible tree-decompositions of triangulations of $\mathbf{G}^{\mathbf{M}}_{\mathcal{B}}$.

2.3 Complexity Theory

We assume that the reader is familiar with basic notions from complexity theory, such as intractability proofs, the computational complexity classes P and NP, and polynomial-time reductions. In this section we shortly review some additional concepts that we use throughout the paper, namely the complexity classes PP and BPP, the Exponential Time Hypothesis and some basic principles from parameterized complexity theory.

The complexity classes PP and BPP are defined as classes of decision problems that are decidable by a probabilistic Turing machine (i.e., a Turing machine that makes stochastic state transitions) in polynomial time with a particular (two-sided) probability of error. The difference between these two classes is in the bound on the error probability. *Yes*-instances for problems in PP are accepted with probability $1/2 + \epsilon$, where ϵ may depend exponentially on the input size (i.e., $\epsilon = 1/c^n$ for a constant $c > 1$). *Yes*-instances for problems in BPP are accepted with a probability that is polynomially bounded away from $1/2$ (i.e., $\epsilon = 1/n^c$). PP-complete problems, such as the problem of determining whether the *majority* of truth assignments to a Boolean formula ϕ satisfies ϕ, are considered to be intractable; indeed, it can be shown that $\mathsf{NP} \subseteq \mathsf{PP}$. In contrast, problems in BPP are considered to be tractable. Informally, a decision problem Π is in BPP if there exists an efficient randomized (Monte Carlo) algorithm that decides Π with high probability of correctness. Given that the error is polynomially bounded away from $1/2$, the probability of answering correctly can be boosted to be arbitrarily close to 1 while still requiring only polynomial time. While obviously $\mathsf{BPP} \subseteq \mathsf{PP}$, the reverse is unlikely; in particular, it is conjectured that $\mathsf{BPP} = \mathsf{P}$ (Clementi, Rolim, & Trevisan, 1998).

The *Exponential Time Hypothesis* (ETH), introduced by Impagliazzo and Paturi (2001), states that there exists a constant $c > 1$ such that deciding any 3Sat instance with n variables takes at least $\Omega(c^n)$ time. Note that the ETH is a stronger assumption than the assumption that $\mathsf{P} \neq \mathsf{NP}$. A sub-exponential but not polynomial-time algorithm for 3Sat, such as an algorithm running in $\mathcal{O}(2^{\sqrt[3]{n}})$, would contradict the ETH but would not imply that $\mathsf{P} = \mathsf{NP}$. We will assume the ETH in our proofs that show the necessity of low tree-width for efficient approximation of MAP.

Sometimes problems are intractable (i.e., NP-hard) in general, but become tractable if some *parameters* of the problem can be assumed to be small. A problem Π is called fixed-parameter tractable for a parameter κ (or a set $\{\kappa_1, \ldots, \kappa_m\}$ of parameters) if it can be solved in time, exponential (or even worse) *only* in κ and polynomial in the input size $|x|$, i.e., in time $\mathcal{O}(f(\kappa) \cdot |x|^c)$ for a constant $c > 1$ and an arbitrary computable function f. In practice, this means that problem instances can be solved efficiently, even when the problem is NP-hard in general, if κ is known to be small. In contrast, if a problem is NP-hard even when κ is small, the problem is denoted as para-NP-hard for κ. The parameterized complexity class FPT consists of all fixed parameter tractable problems $\kappa-\Pi$. While traditionally κ is defined as a mapping from problem instances to natural numbers (e.g., Flum & Grohe, 2006, p. 4), one can easily enhance the theory for rational parameters (Kwisthout, 2011). In the context of this paper, we will in particular consider rational parameters in the range $[0, 1]$, and we will liberally mix integer and rational parameters.

3. Approximating MAP

It is widely known, both from practical experiences and from theoretical results, that 'small tree-width' is often a necessary constraint to render exact Bayesian inferences tractable. However, it is often assumed that such intractable computations can be efficiently *approximated* using inexact algorithms; this assumption appears to be warranted by the observation that in many cases approximation algorithms seem to do a reasonable job in, e.g., estimating posterior distributions, even in networks with high tree-width where exact computations are infeasible (Cheng & Druzdzel, 2000; Sontag, Meltzer, Globerson, Weiss, & Jaakkola, 2008). Whether this observation has a firm theoretical basis, i.e., whether approximation algorithms can or cannot in principle perform well even in situations where tree-width can grow large, is to date not known.

Crucial in answering this question is to make precise what *efficiently approximated* actually pertains to. The on-line Merriam-Webster dictionary lists as one of its entries for *approximate* 'to be very similar to but not exactly like (something)'. In computer science, this similarity is typically defined in terms of *value*: 'approximate solution A has a value that is close to the value of the optimal solution'. However, other notions of approximation can be relevant. One can think of approximating not the *value* of the optimal solution, but the *appearance*: 'approximate solution A' closely resembles the optimal solution'. Also, one can define an approximate solution as one that ranks close to the optimal solution: 'approximate solution A'' ranks within the top-m solutions'. Note that these notions can refer to completely different solutions. One can have situations where the second-best solution does not resemble the optimal solution at all, whereas solutions that look almost the same have a very low value as compared to the optimal solution (Van Rooij & Wareham, 2012; Kwisthout, 2013). Similarly, the second-best solution may either have a value that is almost as good as the optimal solution, or much worse.

In many practical applications, in particular of Bayesian inferences, these definitions of 'approximation' do not (fully) capture the actual notion we are interested in. For example, when trying to approximate the MAP explanation using some sort of randomized computation, we have no *guarantee* on the quality of the solution found, however, we may have a bound on the *likeliness* of a good solution. The current state-of-the-art approximate algorithms for MAP (AnnealedMAP, Yuan, Lu, & Druzdzel, 2004; P-Loc, Park & Darwiche, 2001; BP-LS, Park & Darwiche, 2004) all employ this strategy. The added notion of approximation here, induced by the use of randomized computations, is the allowance of a bounded amount of error.[1]

In the remainder of this section we will elaborate on these notions of approximation when applied to the MAP problem. We will give formal definitions of these approximate problems and show why all of them are intractable in general. For MAP-approximation by value and by structure we will interpret known results in the literature. For MAP-approximation by rank we give a formal proof of intractability; for MAP-approximation using randomized algorithms we give an argument from complexity theory.

3.1 Value-Approximation

Value-approximating MAP is the problem of finding an explanation $approxsol_{\mathcal{B}} \in cansol_{\mathcal{B}}$ that has a value, close to the value of the optimal solution. This 'closeness' can be defined in an additive or in a relative manner; additive meaning that the *absolute difference* between the probability of the optimal and the approximate solution is smaller than some value ρ; relative that the *ratio* between the probability of the optimal and the approximate solution is smaller than some value ρ. Both problems are intractable in general. Abdelbar and Hedetniemi (1998) proved NP-hardness of relative value-approximation for any constant $\rho \geq 1$. This result holds for networks with only binary variables, with at most three incoming arcs per variable, and no evidence. In addition, Kwisthout (2011) showed that it is NP-hard in general to find an explanation $approxsol_{\mathcal{B}}$ with $\Pr(approxsol_{\mathcal{B}}, \mathbf{e}) > \epsilon$ for any constant $\epsilon > 0$, and thus that $\Pr(optsol_{\mathcal{B}}, \mathbf{e}) - \Pr(approxsol_{\mathcal{B}}, \mathbf{e}) \leq \rho$ for $\rho > \Pr(optsol_{\mathcal{B}}, \mathbf{e}) - \epsilon$. The latter result holds even for networks with only binary variables, at most two incoming arcs per variable, a single evidence variable, and no intermediate variables (i.e., when we approximate an MPE problem).

Definition 3.1 (additive value-approximation of MAP) *Let $optsol_{\mathcal{B}}$ be the optimal solution to a MAP problem. An explanation $approxsol_{\mathcal{B}} \in cansol_{\mathcal{B}}$ is defined to ρ-additive value-approximate $optsol_{\mathcal{B}}$ if $\Pr(optsol_{\mathcal{B}}, \mathbf{e}) - \Pr(approxsol_{\mathcal{B}}, \mathbf{e}) \leq \rho$.*

Result 3.2 (Kwisthout, 2011) *It is NP-hard to ρ-additive value-approximate MAP for $\rho > \Pr(optsol_{\mathcal{B}}, \mathbf{e}) - \epsilon$ for any constant $\epsilon > 0$.*

Definition 3.3 (relative value-approximation of MAP) *Let $optsol_{\mathcal{B}}$ be the optimal solution to a MAP problem. An explanation $approxsol_{\mathcal{B}} \in cansol_{\mathcal{B}}$ is defined to ρ-relative value-approximate $optsol_{\mathcal{B}}$ if $\frac{\Pr(optsol_{\mathcal{B}} \mid \mathbf{e})}{\Pr(approxsol_{\mathcal{B}} \mid \mathbf{e})} \leq \rho$.*

Result 3.4 (Abdelbar & Hedetniemi, 1998) *It is NP-hard to ρ-relative value-approximate MAP for $\frac{\Pr(optsol_{\mathcal{B}} \mid \mathbf{e})}{\Pr(approxsol_{\mathcal{B}} \mid \mathbf{e})} \leq \rho$ for any $\rho > 1$.*

1. Observe that some algorithms always converge to the optimal solution, but may take exponential time to do so (e.g., MCMC-type approaches). However, we can turn such an algorithm into an expectation-approximation algorithm by adding a *clock* that halts computations after time, polynomially in the input size, and returning the current best solution which may or may not be optimal (Gill, 1977).

3.2 Structure-Approximation

Structure-approximating MAP is the problem of finding an explanation $\mathit{approxsol}_{\mathcal{B}} \in \mathit{cansol}_{\mathcal{B}}$ that structurally resembles the optimal solution. This is captured using a *solution distance function*, a metric associated with each optimization problem relating candidate solutions with the optimal solution (Hamilton, Müller, van Rooij, & Wareham, 2007). For MAP, the typical structure distance function $d_H(\mathit{approxsol}_{\mathcal{B}}, \mathit{optsol}_{\mathcal{B}})$ is the Hamming distance between explanation $\mathit{approxsol}_{\mathcal{B}}$ and the most probable explanation $\mathit{optsol}_{\mathcal{B}}$. It has been shown by Kwisthout (2013) that no algorithm can calculate the value of even a single variable in the most probable explanation in polynomial time, unless $\mathsf{P} = \mathsf{NP}$; that is, it is NP-hard to find an explanation with $d_H(\mathit{approxsol}_{\mathcal{B}}, \mathit{optsol}_{\mathcal{B}}) \leq |\mathit{optsol}_{\mathcal{B}}| - 1$, even if the variables of the network are bi-partitioned into explanation and evidence variables, and each variable has at most three possible values.

Definition 3.5 (structure-approximation of MAP) *Let $\mathit{optsol}_{\mathcal{B}}$ be the optimal solution to a MAP problem and let d_H be the Hamming distance. An explanation $\mathit{approxsol}_{\mathcal{B}} \in \mathit{cansol}_{\mathcal{B}}$ is defined to d-structure-approximate $\mathit{optsol}_{\mathcal{B}}$ if $d_H(\mathit{approxsol}_{\mathcal{B}}, \mathit{optsol}_{\mathcal{B}}) \leq d$.*

Result 3.6 (Kwisthout, 2013) *It is NP-hard to d-structure-approximate MAP for any $d \leq |\mathit{optsol}_{\mathcal{B}}| - 1$.*

3.3 Rank-Approximation

Apart from allowing an explanation that resembles, or has a probability close to, the most probable explanation, we can also define an approximate solution $\mathit{approxsol}_{\mathcal{B}}$ as an explanation which is one of the m best explanations, for a constant m, that is, $\mathit{approxsol}_{\mathcal{B}} \in \mathit{optsol}_{\mathcal{B}}^{1\ldots m}$ for some m. Note that this explanation may not resemble the most probable explanation nor needs to have a relatively high probability, only that it is *ranked* within the m most probable explanations. We will denote this approximation as a rank-approximation.

Definition 3.7 (rank-approximation of MAP) *Let $\mathit{optsol}_{\mathcal{B}}^{1\ldots m} \subseteq \mathit{cansol}_{\mathcal{B}}$ be the set of the m most probable solutions to a MAP problem and let $\mathit{optsol}_{\mathcal{B}}$ be the optimal solution. An explanation $\mathit{approxsol}_{\mathcal{B}} \in \mathit{cansol}_{\mathcal{B}}$ is defined to m-rank-approximate $\mathit{optsol}_{\mathcal{B}}$ if $\mathit{approxsol}_{\mathcal{B}} \in \mathit{optsol}_{\mathcal{B}}^{1\ldots m}$.*

We will prove that it is NP-hard to m-rank-approximate MAP for any constant m. We do so by a reduction from a variant of LexSat, based on the reduction in Kwisthout, Bodlaender, and van der Gaag (2011). LexSat is defined as follows:

LexSAT
Instance: A Boolean formula ϕ with n variables $X_1, \ldots, X_n$.
Output: The lexicographically largest truth assignment $\mathbf{x}$ to $\mathbf{X} = \{X_1, \ldots, X_n\}$ that satisfies ϕ; the output is $\perp$ if ϕ is not satisfiable.

Here, the lexicographical order of truth assignments maps a truth assignment $\mathbf{x} = x_1, \ldots, x_n$ to a string $\{0, 1\}^n$, with $\{0\}^n$ (all variables set to false) is the lexicographically *smallest*, and $\{1\}^n$ (all variables set to true) is the lexicographically *largest* truth assignment. LexSat is NP-hard; in particular, LexSat has been proven to be complete for the class $\mathsf{FP}^{\mathsf{NP}}$ (Krentel,

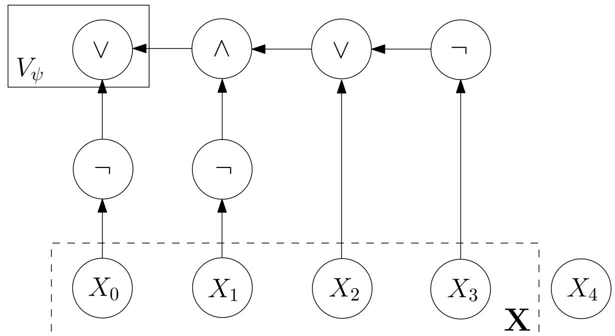

Figure 1: Example construction of $\mathcal{B}_{\phi_{\text{ex}}}$ from LexSat′ instance ϕ_{ex}

1988). In our proofs we will use the following variant that always returns a truth assignment (rather than $\perp$, in case ϕ is unsatisfiable):

LexSAT′
Instance: A Boolean formula ϕ with n variables $X_1, \ldots, X_n$.
Output: The lexicographically largest satisfying truth assignment $\mathbf{x}$ to $\psi = (\neg X_0) \vee \phi$ that satisfies ψ.

Note that if ϕ is satisfiable, then X_0 is never set to false in the lexicographically largest satisfying truth assignment to ψ, yet X_0 is necessarily set to false if ϕ is not satisfiable; hence, unsatisfying truth assignments to ϕ are always ordered after satisfying truth assignments in the lexicographical ordering. Note that LexSat trivially reduces to LexSat′ using a simple transformation. We claim the following.

Theorem 3.8 *No algorithm can find an approximation approxsol$_\mathcal{B}$ $\in$ optsol$_\mathcal{B}^{1\ldots m}$, for any constant m, in polynomial time, unless* $\mathsf{P} = \mathsf{NP}$.

In our proof we describe a polynomial-time one-Turing reduction[2] from LexSat′ to m-rank-approximated-MAP for an arbitrary constant m. The reduction largely follows the reduction as presented by Kwisthout et al. (2011) with some additions. We will take the following LexSat′-instance as running example in the proof: $\phi_{\text{ex}} = \neg X_1 \wedge (X_2 \vee \neg X_3)$; correspondingly, $\psi_{\text{ex}} = (\neg X_0) \vee (\neg X_1 \wedge (X_2 \vee \neg X_3))$ in this example. We set $m = 3$ in the example construct. We now construct a Bayesian network $\mathcal{B}_\phi$ from ψ as follows (Figure 1).

For each variable X_i in ψ, we introduce a binary root variable X_i in $\mathcal{B}_\phi$ with possible values true and false. We set the prior probability distribution of these variables to $\Pr(X_i = \text{true}) = 1/2 - \frac{2^{i+1}-1}{2^{n+2}}$. In addition, we include a uniformly distributed variable X_{n+1} in $\mathcal{B}_\phi$ with m values $x_{n+1}^1, \ldots, x_{n+1}^m$. The variables $X_0, \ldots, X_n$ together form the set $\mathbf{X}$. Note that the prior probability of a joint value assignment $\mathbf{x}$ to $\mathbf{X}$ is higher than the prior probability of a different joint value assignment $\mathbf{x}'$ to $\mathbf{X}$, if and only if the corresponding

2. A a function problem f is polynomial-time one-Turing reducible to a function problem g if there exist polynomial-time computable functions T_1 and T_2 such that for every $x, f(x) = T_l(x, g(T_2(x)))$ (Toda, 1994). One-Turing reductions can be seen as equivalent to many-one reductions, but then applied to function problems.

truth assignment $\mathbf{x}$ to the LexSat$'$ instance has a lexicographically larger truth assignment than $\mathbf{x}'$. In the running example, we have that $\Pr(X_0 = \text{TRUE}) = 15/32$, $\Pr(X_1 = \text{TRUE}) = 13/32$, $\Pr(X_2 = \text{TRUE}) = 9/32$, and $\Pr(X_3 = \text{TRUE}) = 1/32$, and $\Pr(X_4 = x_4^1) = \Pr(X_4 = x_4^2) = \Pr(X_4 = x_4^3) = 1/3$. Observe that we have that $\Pr(X_0) \cdot \ldots \cdot \Pr(X_{i-1}) \cdot \overline{\Pr(X_i)} > \overline{\Pr(X_0)} \cdot \ldots \cdot \overline{\Pr(X_{i-1})} \cdot \Pr(X_i)$ for every i, i.e., the ordering property such as stated above is attained.

For each logical operator T in ψ, we introduce an additional binary variable in $\mathcal{B}_\phi$ with possible values TRUE and FALSE, and with as parents the sub-formulas (or single sub-formula, in case of a negation operator) that are bound by the operator. The conditional probability distribution of that variable matches the truth table of the operator, i.e., $\Pr(T = \text{TRUE} \mid \pi(T)) = 1$ if and only if the operator evaluates to TRUE for that particular truth value of the sub-formulas bound by T. The top-level operator is denoted by V_ψ. It is readily seen that $\Pr(V_\psi = \text{TRUE} \mid \mathbf{x}) = 1$ if and only if the truth assignment to the variables in ψ that matches $\mathbf{x}$ satisfies ψ, and $\Pr(V_\psi = \text{TRUE} \mid \mathbf{x}) = 0$ otherwise. Observe that the m-valued variable X_{n+1} is independent of every other variable in $\mathcal{B}_\phi$. Further note that the network, including all prior and conditional probabilities, can be described using a number of bits which is polynomial in the size of ϕ. In the MAP instance constructed from ϕ, we set V_ψ as evidence set with $V_\psi = \text{TRUE}$ as observation and we set $\mathbf{X} \cup \{X_{n+1}\}$ as explanation set.

Proof. Let ϕ be an instance of LexSat$'$, and let $\mathcal{B}_\phi$ be the network constructed from ϕ as described above. We have for any joint value assignment $\mathbf{x}$ to $\mathbf{X}$ that $\Pr(\mathbf{X} = \mathbf{x} \mid V_\psi = \text{TRUE}) = \alpha \cdot \Pr(\mathbf{X} = \mathbf{x})$ for a normalization constant $\alpha > 0$ if $\mathbf{x}$ corresponds to a satisfying truth assignment to ψ, and $\Pr(\mathbf{X} = \mathbf{x} \mid V_\psi = \text{TRUE}) = 0$ if $\mathbf{x}$ corresponds to a non-satisfying truth assignment to ψ. Given the prior probability distribution of the variables in $\mathbf{X}$, we have that all satisfying joint assignments $\mathbf{x}$ to $\mathbf{X}$ are ordered by the posterior probability $\Pr(\mathbf{x} \mid V_\psi = \text{TRUE}) > 0$, where all non-satisfying joint value assignments have probability $\Pr(\mathbf{x} \mid V_\psi = \text{TRUE}) = 0$ and thus are ordered after satisfying assignments. The joint value assignment that has the highest posterior probability thus is the lexicographically largest satisfying truth assignment to ψ.

If we take the m-th valued variable X_{n+1} into account, we have that for every $\mathbf{x}$, the m joint value assignments to $\mathbf{X} \cup \{X_{n+1}\}$ have the same probability since $\Pr(\mathbf{x}, X_{n+1} \mid V_\psi = \text{TRUE}) = \Pr(\mathbf{x} \mid V_\psi = \text{TRUE}) \cdot \Pr(X_{n+1})$. But then, the m joint value assignments $\mathbf{x}^\mathbf{m}$ to $\mathbf{X} \cup \{X_{n+1}\}$ that correspond to the lexicographically largest satisfying truth assignment $\mathbf{x}$ to ψ all have the same posterior probability $\Pr(\mathbf{x}^\mathbf{m} \mid V_\psi = \text{TRUE})$. Thus, any algorithm that returns one of the m-th ranked joint value assignments to the explanation set $\mathbf{X} \cup \{X_{n+1}\}$ with evidence $V_\psi = \text{TRUE}$ can be transformed in polynomial time to an algorithm that solves LexSat$'$. We conclude that no algorithm can m-rank-approximate MAP, for any constant m, in polynomial time, unless $\mathsf{P} = \mathsf{NP}$. □

Note that, technically speaking, our result is even stronger: as LexSat$'$ is $\mathsf{FP}^{\mathsf{NP}}$-complete and the reduction described above actually is a one-Turing reduction from LexSat$'$ to m-rank-approximation-MAP, the latter problem is $\mathsf{FP}^{\mathsf{NP}}$-hard. We can strengthen the result further by observing that all variables (minus V_ψ) that mimic operators deterministically depend on their parents and thus can be added to the explanation set without substantially changing the proof above. This implies that m-rank-approximation-MPE is also $\mathsf{FP}^{\mathsf{NP}}$-hard. Lastly, we can strengthen the result by replacing the m-th valued variable

X_{n+1} by $\lceil \log_2 m \rceil$ unconnected binary variables X_{n+1} to $X_{n+\lceil \log_2 m \rceil}$ with uniform probability. Still, any algorithm returning one of the m-th ranked joint value assignments to $\mathbf{X} \cup \{X_{n+1}, \ldots, X_{n+\lceil \log_2 m \rceil}\}$ in polynomial time will effectively solve LexSat$'$ in polynomial time.

Result 3.9 *It is* NP*-hard to m-rank-approximate MAP for any constant m.*

3.4 Expectation-Approximation

The last notion of MAP approximation we will discuss here returns in polynomial time an explanation $approxsol_{\mathcal{B}} \in cansol_{\mathcal{B}}$ that is likely to be the most probable explanation, but allows for a small margin of error; i.e., there is a small probability that the answer is not the optimal solution, and then no guarantees are given on the quality of that solution. These approximations are closely related to randomized algorithms that run in polynomial time but whose output has a small probability of error, viz., Monte Carlo algorithms. This notion of approximation–which we will refer to as *expectation-approximation* (Kwisthout & van Rooij, 2013)–is particularly relevant for typical Bayesian approximation methods, such as Monte Carlo sampling and repeated local search algorithms.

Definition 3.10 (expectation-approximation of MAP) *Let $optsol_{\mathcal{B}}$ be the optimal solution to a MAP problem and let $\mathbb{E}$ be the the expectation function (Papoulis, 1984). An explanation $approxsol_{\mathcal{B}} \in cansol_{\mathcal{B}}$ is defined to ϵ-expectation-approximate $optsol_{\mathcal{B}}$ if $\mathbb{E}(\Pr(optsol_{\mathcal{B}}) \neq \Pr(approxsol_{\mathcal{B}})) < \epsilon$.*

In order to be of practical relevance, we want the error to be *small*, i.e., when casted as a decision problem, we want the probability of answering correctly to be bounded away from $1/2$. In that case, we can amplify the probability of answering correctly arbitrarily close to 1 in polynomial time, by repeated evocation of the algorithm. Otherwise, e.g., if the error depends exponentially on the size of the input, we need an exponential number of repetitions to achieve such a result. Problems that enjoy polynomial-time Monte Carlo algorithms are in the complexity class BPP; problems that may need exponential time to reduce the probability of error arbitrarily close to 0 are in the complexity class PP.

As MAP is NP-hard, an efficient randomized algorithm solving MAP in polynomial time with a bounded probability of error, would imply that $\mathsf{NP} \subseteq \mathsf{BPP}$. This is considered to be highly unlikely, as almost every problem that enjoys an efficient randomized algorithm has been proven to be in P, i.e., be decidable in deterministic polynomial time.[3] On various grounds it is believed that $\mathsf{BPP} = \mathsf{P}$ (Clementi et al., 1998), and thus an efficient randomized algorithm for MAP would (under that assumption) establish $\mathsf{P} = \mathsf{NP}$. Therefore, no algorithm can expectation-approximate MAP in polynomial time with bounded margin of error unless $\mathsf{NP} \subseteq \mathsf{BPP}$. This result holds also for MPE, which is in itself already NP-hard, even for binary variables and in-degree 2 (Kwisthout, 2011).[4]

3. The most dramatic example of such a problem is PRIMES: given a natural number, decide whether it is prime. While efficient randomized algorithms for PRIMES have been around quite some time (establishing that PRIMES $\in$ BPP), only fairly recently it has been proven that PRIMES is in P (Agrawal, Kayal, & Saxena, 2004).
4. In fact, it holds for value-approximation, structure-approximation, and rank-approximation of MAP as well, as all three problems are in themselves NP-hard (see also Abdelbar & Hedetniemi, 1998, p. 35).

Result 3.11 *There cannot exist a randomized algorithm that ϵ-expectation-approximates MAP in polynomial time for $\epsilon < 1/2 - 1/n^c$ for a constant c unless* $\mathsf{NP} \subseteq \mathsf{BPP}$.

3.5 Discussion

In the previous subsections we showed that all approximation notions we established are in fact intractable, under various assumptions. The results hold for MAP in general, but can in many cases be strengthened to hold for MPE (i.e., where the network is two-partitioned into evidence and explanation variables); in either case, both the cardinality c and in-degree d of the nodes (and consequently, the size of the CPTs) is bounded. The results hold with empty (or singleton) evidence sets. The results are summarized in Table 1.

Approximation	constraints	assumption	reference
value, additive	$c = 2, d = 2,$ $\lvert\mathbf{E}\rvert = 1, \mathbf{I} = \varnothing$	$\mathsf{P} \neq \mathsf{NP}$	(Kwisthout, 2011, p. 1462)
value, ratio	$c = 2, d = 3,$ $\mathbf{E} = \varnothing$	$\mathsf{P} \neq \mathsf{NP}$	(Abdelbar & Hedetniemi, 1998, p. 24)
structure	$c = 3, d = 3,$ $\mathbf{I} = \varnothing$	$\mathsf{P} \neq \mathsf{NP}$	(Kwisthout, 2013, p. 345)
rank	$c = 2, d = 2,$ $\lvert\mathbf{E}\rvert = 1, \mathbf{I} = \varnothing$	$\mathsf{P} \neq \mathsf{NP}$	Section 3.3
expectation	$c = 2, d = 2,$ $\lvert\mathbf{E}\rvert = 1, \mathbf{I} = \varnothing$	$\mathsf{NP} \not\subseteq \mathsf{BPP}$	Section 3.4

Table 1: Summary of intractability results for MAP approximations

4. The Necessity of Low Tree-Width for Efficient Approximation of MAP

In the previous section we have shown that for four notions of approximating MAP, no efficient general approximation algorithm can be constructed unless either $\mathsf{P} = \mathsf{NP}$ or $\mathsf{NP} \subseteq \mathsf{BPP}$. However, MAP is *fixed-parameter tractable* for a number of problem parameters; for example, $\{\text{tw}, c, q\}$−MAP is in FPT for parameters tree-width (tw), cardinality of the variables ($c = \max_i |\Omega(V_i \in \mathbf{V})|$), and probability of the most probable solution ($q = \Pr(\mathit{optsol}_{\mathcal{B}}, \mathbf{e})$). Surely, if we can compute $\{\kappa_1, \ldots, \kappa_m\}$−MAP exactly in FPT time, we can also approximate $\{\kappa_1, \ldots, \kappa_m\}$−MAP in FPT time. A question remains, however, whether approximate MAP can be fixed-parameter tractable for a *different* set of parameters than exact MAP.

Tree-width has been shown to be a *necessary* parameter for efficient exact computation of the Inference problem (and, by a trivial adjustment illustrated in Section 4.3, also of MAP), under the assumption that the ETH holds (Kwisthout et al., 2010). In this section, we will show that low tree-width is also a necessary parameter for efficient *approximate* computation for value-approximations, structure-approximations, and rank-approximations. We also argue (in Section 5) that it is *not* a necessary parameter for efficient expectation-approximation. In the next sub-section we will review so-called tree-width-preserving reductions (tw-reductions), a special kind of polynomial many-one reductions that preserve tree-width of the instances (Kwisthout et al., 2010). In Sub-section

4.2 we sketch how this notion can be used to tw-reduce Constraint Satisfaction to Inference. Together with the known result that Constraint Satisfaction instances with high tree-width cannot have sub-exponential algorithms, unless the ETH fails (Marx, 2007), it was established by Kwisthout et al. that there cannot be a (general-purpose) algorithm that decides Inference on instances with high tree-width in sub-exponential time, unless the ETH fails. Here, the Inference problem is the problem of deciding whether in a Bayesian network $\mathcal{B}$ with designated sets $\mathbf{H}$ and $\mathbf{E}$ and a rational number q, it is the case that $\Pr(\mathbf{H} = \mathbf{h} \mid \mathbf{E} = \mathbf{e}) > q$. More precisely, the following theorem was proved:

Theorem 4.1 *If there exists a computable function f such that* Inference *can be decided by an algorithm running in time*

$$f(\mathbf{G}^{\mathbf{M}}_{\mathcal{B}}) \cdot \|\mathcal{B}\|^{o\left(\frac{\mathrm{tw}(\mathbf{G}^{\mathbf{M}}_{\mathcal{B}})}{\log \mathrm{tw}(\mathbf{G}^{\mathbf{M}}_{\mathcal{B}})}\right)}$$

for arbitrary Inference *instances $(\mathcal{B}, \mathbf{H}, \mathbf{h}, \mathbf{E}, \mathbf{e}, q)$ with a moralized graph $\mathbf{G}^{\mathbf{M}}_{\mathcal{B}}$ with tree-width $\mathrm{tw}(\mathbf{G}^{\mathbf{M}}_{\mathcal{B}})$, then the* ETH *fails.*

The reader is referred to Kwisthout et al. (2010) for the full proof.[5] In the remainder of this section, we will show how this proof can be augmented to establish similar results for MAP, value-approximate MAP, structure-approximate MAP, and rank-approximate MAP (Sub-sections 4.3 and 4.4).

4.1 Tree-Width-Preserving Reductions

Tree-width-preserving reductions are defined by Kwisthout et al. (2010) as a means to reduce Constraint Satisfaction to Inference while ensuring that tree-width is preserved between instances in the reduction, modulo a linear factor.

Definition 4.2 (Kwisthout et al., 2010) *Let A and B be computational problems such that* tree-width *is defined on instances of both A and B. We say that A is* polynomial-time tree-width-preserving reducible*, or* tw-reducible*, to B if there exists a polynomial-time computable function g and a linear function l such that $x \in A$ if and only if $g(x) \in B$ and* $\mathrm{tw}(g(x)) = l(\mathrm{tw}(x))$. *The pair (g, l) is called a* tw-reduction.

We will use this notion to show that Constraint Satisfaction also tw-reduces to MAP, value-approximate MAP, structure-approximate MAP, and rank-approximate MAP.

4.2 Proof Sketch

The tw-reduction from (binary) Constraint Satisfaction to Inference, as presented by Kwisthout et al. (2010), constructs a Bayesian network $\mathcal{B}_{\mathcal{I}}$ from an instance $\mathcal{I} = (\mathbf{V}, \mathbf{D}, \mathbf{C})$ of Constraint Satisfaction, where $\mathbf{V}$ denotes the set of variables of $\mathcal{I}$, $\mathbf{D}$ denotes the set of values of these variables, and $\mathbf{C}$ denotes the set of binary constraints defined over $\mathbf{V} \times \mathbf{V}$.

5. The results by Kwisthout et al. (2010) did not rule out the existence of special-case algorithms, that assume (and utilize) a particular property of the instance, such as a particular orientation of the arcs or particular planarity properties of the graph structure, failing when the assumption is violated. The results in the current paper, that are built on this result, inherit this constraint.

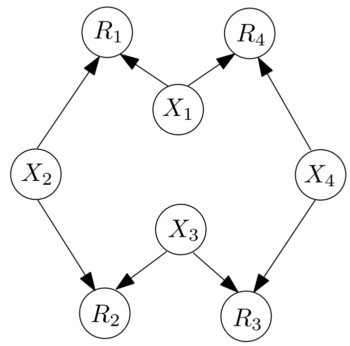

Figure 2: Example construction of $\mathcal{B}_{\mathcal{I}}$ from example CSP instance $\mathcal{I}$

The constructed network $\mathcal{B}_{\mathcal{I}}$ includes uniformly distributed variables X_i, corresponding with the variables in $\mathbf{V}$, and binary variables R_j, corresponding with the constraints in $\mathbf{C}$. The parents of the variables R_j are the variables X_i that are bound by the constraints; their conditional probability distributions match the imposed constraints on the variables (i.e., $\Pr(R_j = \textsc{true} \mid \mathbf{x} \in \Omega(\pi(R_j))) = 1$ if and only if the joint value assignment $\mathbf{x}$ to the variables bound by R_j matches the constraints imposed on them by R_j. Figure 2, taken from Kwisthout et al., shows the result of the construction so far for an example Constraint Satisfaction instance with four variables X_1 to X_4, where $\mathbf{C}$ contains four constraints that bind respectively (X_1, X_2), (X_1, X_4), (X_2, X_3), and (X_3, X_4).

The tree-width of the thus obtained network equals $\max(2, \mathrm{tw}(\mathbf{G}_{\mathcal{I}}))$, where $\mathbf{G}_{\mathcal{I}}$ is the primal graph of $\mathcal{I}$; note that the tree-width of $\mathcal{B}_{\mathcal{I}}$ at most increases the tree-width of $\mathbf{G}_{\mathcal{I}}$ by 1. In order to enforce that *all* constraints are simultaneously enforced, the constraint nodes R_j need to be connected by extra nodes mimicking 'and' operators. A crucial aspect of the tw-reduction is the topography of this connection of the nodes R_j: care must be taken not to blow up tree-width by arbitrarily connecting the nodes, e.g., by a log-deep binary tree. The original proof uses a minimal tree-decomposition of the moralization of $\mathcal{B}_{\mathcal{I}}$ and describes a procedure to select which nodes need to be connected such that the tree-width of the resulting graph is at most the tree-width of $\mathbf{G}_{\mathcal{I}}$ plus 3. The conditional probability distribution of the nodes A_k is defined as follows.

$$\Pr(A_k = \textsc{true} \mid \mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \bigwedge_{V \in \pi(A_k)} (V = \textsc{true}) \\ 0 & \text{otherwise} \end{cases}$$

For a node A_k without any parents, $\Pr(A_k = \textsc{true}) = 1$. The graph that results from applying this procedure to the example is given in Figure 3 (also taken from Kwisthout et al., 2010). Now, $\Pr(A_1 = \textsc{true} \mid \mathbf{x}) = 1$ if $\mathbf{x}$ corresponds to a satisfying value assignment to $\mathbf{V}$ and 0 otherwise; correspondingly, $\Pr(A_1 = \textsc{true}) > 0$ if and only if the Constraint Satisfaction instance is satisfiable.

4.3 MAP Result

The tw-reduction described in the previous sub-section can be easily modified to a tw-reduction from Constraint Satisfaction to MAP. We do this by adding a binary node

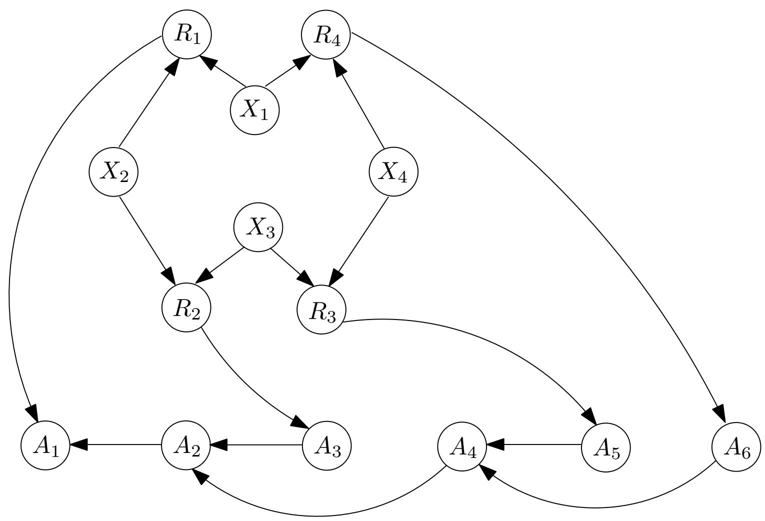

Figure 3: Resulting graph $\mathcal{B}_\mathcal{I}$ after adding nodes A_k and appropriate arcs

---

$V_\mathcal{I}$ to the thus obtained graph, with A_1 as its only parent and with conditional probability $\Pr(V_\mathcal{I} = \text{TRUE} \mid A_1 = \text{TRUE}) = 1$ and $\Pr(V_\mathcal{I} = \text{TRUE} \mid A_1 = \text{FALSE}) = 1/2 - \epsilon$, where ϵ is a number, smaller than $1/|\mathbf{D}|^{|\mathbf{V}|}$. Consequently, we have that $\Pr(V_\mathcal{I} = \text{TRUE}) > 1/2$ if $\mathcal{I}$ is satisfiable, and $\Pr(V_\mathcal{I} = \text{TRUE}) < 1/2$ if $\mathcal{I}$ is not satisfiable; hence, a MAP query with explanation set $\mathbf{H} = \{V_\mathcal{I}\}$ will return $V_\mathcal{I} = \text{TRUE}$ if and only if $\mathcal{I}$ is satisfiable. We added a single node to $\mathcal{B}_\mathcal{I}$, with A_1 as only parent, thus increasing the tree-width of $\mathcal{B}_\mathcal{I}$ by at most 1. Hence, Constraint Satisfaction tw-reduces to MAP.

4.4 Approximation Intractability Results

In a similar way we can modify the reduction from Sub-section 4.2 to show that value-approximations, structure-approximations, and rank-approximations can be tw-reduced from Constraint Satisfaction, as sketched below.

4.4.1 Value-Approximation

We add a binary node $V_\mathcal{I}$, with A_1 as its only parent, and with conditional probability $\Pr(V_\mathcal{I} = \text{TRUE} \mid A_1 = \text{TRUE}) = 1$ and $\Pr(V_\mathcal{I} = \text{TRUE} \mid A_1 = \text{FALSE}) = 0$. We observe this variable to be set to TRUE. This enforces that $\Pr(A_1 = \text{TRUE} \mid V_\mathcal{I} = \text{TRUE})$ has a non-zero probability (i.e., $\mathcal{I}$ is solvable) since otherwise there is conflicting evidence in the thus constructed network. Thus, any value-approximation algorithm with explanation set $\mathbf{H} = \{A_1\}$ and evidence $\mathbf{e} = \{V_\mathcal{I} = \text{TRUE}\}$ that can return a solution $\mathit{approxsol}_\mathcal{B} \in \mathit{cansol}_\mathcal{B}$ with $\Pr(\mathit{approxsol}_\mathcal{B}, \mathbf{e}) > \epsilon$ for any constant $\epsilon > 0$, (that is, approximates additively for $\rho > \Pr(\mathit{optsol}_\mathcal{B}) - \epsilon$ or relatively for $\rho > \frac{\Pr(\mathit{optsol}_\mathcal{B})}{\epsilon}$) effectively solves Constraint Satisfaction: if there *exists* a solution with non-zero probability, the construction dictates that $\mathcal{I}$ must be solvable. Given that we added a single node to $\mathcal{B}_\mathcal{I}$, with A_1 as only parent, this increases the tree-width of $\mathcal{B}_\mathcal{I}$ by at most 1. Hence, Constraint Satisfaction tw-reduces to value-approximate MAP.

4.4.2 Structure-Approximation

Observe from the tw-reduction to MAP in Sub-section 4.3 that, since $\mathbf{H}$ consists of a singleton binary variable, we trivially have that no algorithm can find an explanation $approxsol_{\mathcal{B}} \in cansol_{\mathcal{B}}$ with $d_H(approxsol_{\mathcal{B}}, optsol_{\mathcal{B}}) \leq |optsol_{\mathcal{B}}| - 1 = 0$ since that would solve the MAP query. We can extend this result to hold for explanation sets with size k for any constant k, i.e., no structure-approximation algorithm can guarantee to return the correct value of *one* of the k variables in $\mathbf{H}$ in polynomial time in instances of high tree-width, unless the ETH fails.

Instead of adding a single binary node $V_{\mathcal{I}}$ as in the tw-reduction to MAP, we add k binary nodes $V_{\mathcal{I}}^1 \ldots V_{\mathcal{I}}^k$, all with A_1 as their only parent and with $\Pr(V_{\mathcal{I}}^j = \text{TRUE} \mid A_1 = \text{TRUE}) = 1$ and $\Pr(V_{\mathcal{I}}^j = \text{TRUE} \mid A_1 = \text{FALSE}) = 1/2 - \epsilon$ for $1 \leq j \leq k$ and with ϵ as described in Sub-section 4.3. A MAP query with explanation set $\mathbf{H} = \bigcup_{1 \leq j \leq k} V_{\mathcal{I}}^j$ will then return $\forall_{1 \leq j \leq k} V_{\mathcal{I}}^k = \text{TRUE}$ if and only if $\mathcal{I}$ is satisfiable; if $\mathcal{I}$ is not satisfiable, a MAP query will return $\forall_{1 \leq j \leq k} V_{\mathcal{I}}^k = \text{FALSE}$ as most probable explanation. Hence, any structure-approximation algorithm that can correctly return the value of one of the variables in $\mathbf{H}$, effectively solves Constraint Satisfaction. As we added k nodes to $\mathcal{B}_{\mathcal{I}}$, with A_1 as their only parent and no outgoing arcs, the tree-width of $\mathcal{B}_{\mathcal{I}}$ increases by at most 1. Hence, Constraint Satisfaction tw-reduces to structure-approximate MAP.

4.4.3 Rank-Approximation

We modify the proof of Sub-section 4.3 as follows. In addition to adding a binary node $V_{\mathcal{I}}$ as specified in that section, we also add $\lceil \log_2 m \rceil$ unconnected binary variables $\mathbf{M}_{\mathcal{I}} = \{M_{\mathcal{I}}^1 \ldots M_{\mathcal{I}}^{\lceil \log_2 m \rceil}\}$ with uniform probability to $\mathbf{H}$; an m-rank-approximate MAP query with explanation set $\mathbf{H} = \{V_{\mathcal{I}}\} \cup \mathbf{M}_{\mathcal{I}}$ will return $V_{\mathcal{I}} = \text{TRUE}$ (and $\mathbf{M}_{\mathcal{I}}$ set to an arbitrary value) if and only if $\mathcal{I}$ is satisfiable. The addition of $\mathbf{M}_{\mathcal{I}}$ does not increase tree-width, hence, Constraint Satisfaction tw-reduces to m-rank-approximate MAP.

4.5 Discussion

For efficient exact computation, value-approximation, structure-approximation, and rank-approximation of MAP we showed that bounded tree-width is a necessary condition, under the assumption of the ETH, for any general-purpose algorithm that accepts arbitrary instances. This does not rule out the possibility that there may exist special-purpose algorithms that can compute or approximate MAP explanations of specific networks with a special structure or distribution (as was already concluded by Kwisthout et al. (2010) for the Inference problem in Bayesian networks). However, as the previous sub-section shows, the approximation problems are intractable even for extreme lower bounds on the approximation quality, and by the very nature of the reductions it follows that if we can effectively approximate MAP explanations by value, structure, or rank, we can decide the problem exactly as well. This leaves little room for efficient approximation algorithms for MAP instances with high tree-width when we approximate by value, structure, or rank.

5. Expectation-Approximation and the Classes FERT and FPERT

In the previous section we showed that we cannot value-approximate, structure-approximate, or rank-approximate MAP on instances with high tree-width, unless the ETH fails. Now what about expectation-approximation? It appears that the strategy that was employed in the previous subsection cannot be used to show a similar result for expectation-approximation. In fact, we have reasons to believe that efficient expectation-approximation of MAP indeed depends on a *different* set of parameters than the other notions of approximation discussed above, and that bounded tree-width is *not necessary* for this particular notion of approximation. This notion of parameterized approximability is not well captured by the 'traditional' fixed parameter tractable class FPT; therefore, we will introduce the parameterized complexity classes FERT (Fixed Error Randomized Tractable) and FPERT (Fixed Parameter and Error Randomized Tractable) that characterize this notion of efficient parameterized expectation-approximation. Intuitively, in contrast to the class FPT, that parameterizes running time, these classes parameterize the error probability (FERT), respectively both the running time and the error probability (FPERT).

To the best of our knowledge, there is no previous work that proposes to parameterize the probability of acceptance of a probabilistic Turing machine. Montoya and Müller (2013) define the class BPFPT that assumes a bounded error which is independent of the parameterization κ, but where the 'amount of randomness' (operationalized by the number of coins used) is bounded. Arvind and Raman (2002) propose randomized approximation algorithms for counting problems, where both the running time and the approximation ratio are parameterized, but the error probability is constant. Both authors, however, assume a bounded (rather than parameterized) error.

In the next section we will set up the formal machinery for our results. We introduce natural parameterizations of MajSAT, respectively E-MajSAT, that are in FERT, respectively FPERT. We will show that a restricted variant of MAP indeed is in FERT, parameterized by the probability of the most probable explanation, and that the Most Frugal Explanations problem (Kwisthout, 2015) is in FPERT for a number of parameterizations. We will elaborate on the relation between these classes and the classes BPP, PP, and FPT, and finally, we will propose a 'road map' for future research.

5.1 Parameterizing the Error Bound in Randomized Algorithms

We formally define the complexity class FERT as follows:

Definition 5.1 *Let* Π *be a decision problem and let* $\kappa-\Pi$ *be a parameterization of* Π*. We have that* $\kappa-\Pi \in$ FERT *if and only if there exists a probabilistic Turing machine* $\mathcal{M}$ *that halts after time, polynomial in the size of the input* x*, with the following acceptance criteria.* $\mathcal{M}$ *accepts* Yes*-instances of* Π *with probability* $1/2 + \min(f(\kappa), 1/|x|^c)$ *for a constant* c *and arbitrary function* $f : \mathbb{R} \rightarrow \langle 0, 1/2]$*;* No*-instances are accepted with probability at most* $1/2$*.*

Observe that in this definition we demand that $\mathcal{M}$ halts after time, polynomial in the input size (and independent from the parameterization κ), yet that the probability of acceptance of *Yes*-instances may depend on any function of κ. Intuitively, the class FERT characterizes problems that can be efficiently computed with a randomized algorithm (i.e., in polynomial time, with error arbitrarily close to 0) if κ is bounded. The canonical parameterized problem that is in FERT is $\{r\}-$MajSAT, where r denotes the fraction of satisfying truth assignments (or, equivalently, the probability that a random truth assignment accepts). This follows as a corollary from the following result by Littman, Majercik, and Pitassi (2001):

Lemma 5.2 (adapted from Littman et al., 2001) *Let v be the number of accepting truth assignments to a Boolean formula ϕ, and let $\hat{v}$ be the estimate of v found via random sampling using w samples over the variables of ϕ. Let $\epsilon > 0$ be a target approximation error. The probability that $|v - \hat{v}| > \epsilon$ is less than $2e^{-2\epsilon^2 w}$.*

Note that when solving a MajSAT-instance, the target approximation error ϵ directly depends on the probability r that a random truth assignment accepts, as the acceptable error (i.e., the error that still gives the correct answer to the MajSAT instance) in the random sampling algorithm is $\epsilon = |r - 1/2|$. So, if the probability of acceptance of a random truth assignment is polynomially bounded away from $1/2$, we can guarantee an arbitrarily small error using only polynomially many samples using a straightforward randomized algorithm.

Corollary 5.3 $\{r\}-$MajSAT $\in$ FERT.

When we allow both parameterization of the running time *and* of the probability of acceptance, we get the complexity class FPERT defined as follows:

Definition 5.4 *Let Π be a decision problem and let $\{\kappa_1, \kappa_2\}-\Pi$ be a parameterization of Π. We have that $\{\kappa_1, \kappa_2\}-\Pi \in$ FPERT if and only if there exists a probabilistic Turing machine $\mathcal{M}$ that halts after time $\mathcal{O}(f_1(\kappa_1) \cdot |x|^{c_1})$, that accepts* Yes-*instances of Π with probability $1/2 + \min(f_2(\kappa_2), 1/|x|^{c_2})$, and accepts* No-*instances with probability at most $1/2$. Here, $f_1 : \mathbb{R} \to \mathbb{R}$ and $f_2 : \mathbb{R} \to \langle 0, 1/2]$ are arbitrary computable functions and c_1 and c_2 are constants.*

We can also define a canonical problem in FPERT, based on the observation that $\{p\}$-SAT is in FPT (Flum & Grohe, 2006, here parameter p denotes the number of variables in the formula) and on Corollary 5.3:

E-MajSAT
Instance: Let ϕ be a Boolean formula with n variables $x_i, i = 1, \ldots, n, n \geq 1$, furthermore we partition the variables into sets $\mathbf{X_E}$ and $\mathbf{X_M}$.
Question: Is there a truth assignment to $\mathbf{X_E}$ such that the majority of the truth assignments to $\mathbf{X_M}$ satisfy ϕ?
Parameter: $\kappa_1 = p; \kappa_2 = r$; here p is the number of variables in the set $\mathbf{X_E}$; we define r as follows. Let $r_{\mathbf{x_E}}$ denote the ratio of accepting truth assignments to $\mathbf{X_M}$ given a particular truth assignment $\mathbf{x_E}$ to $\mathbf{X_E}$. We then define $r = \min_{\mathbf{x_E}}(|1/2 - r_{\mathbf{x_E}}|)$.

Informally, r describes the minimum absolute distance to $1/2$ of the fraction of accepting truth assignments for any truth assignment $\mathbf{x_E}$ to $\mathbf{X_E}$. Observe that we can try (brute-force) all truth assignments to $\mathbf{X_E}$ and, for each truth assignment, expectation-approximate whether that truth assignment is such that the majority of truth assignments to $\mathbf{X_M}$ satisfy ϕ. This algorithm runs in time $\mathcal{O}(2^p \cdot n^c)$ for a constant c, and has a probability at least $1/2 + f(r)$ of answering correctly (using a polynomial number of samples).

Corollary 5.5 $\{p, r\}-$E-MajSAT $\in$ FPERT.

5.2 Parameterized Expectation-Approximation of MAP

Proving that a problem $\kappa-\Pi$ is in FPT is normally done constructively, i.e., by giving a deterministic algorithm that decides Π in time $\mathcal{O}(f(\kappa) \cdot |x|^c)$ for a constant $c > 1$. Similarly, proving that $\kappa-\Pi$ is in FERT is done by giving a randomized algorithm[6] that decides Π in polynomial time with error at most $1/2 - \min(f(\kappa), 1/|x|^c)$. We did not succeed in giving such an algorithm for MAP in general, however, we can prove that a restricted variant of MAP is in FERT when parameterized *only* by the probability of the most probable explanation, *despite* that this restricted variant remains PP-complete in general and that bounded tree-width is a necessary parameter to approximate this problem by value, structure, or rank:

Constrained MAP
Instance: As in MAP. In addition, we demand that $\mathbf{E} = \varnothing$, $\mathbf{H}$ consists of a singleton node H with no outgoing edges, and $\Omega(H) = \{\text{TRUE}, \text{FALSE}\}$.
Question: Is $\Pr(H = \text{TRUE}) > 1/2$?
Parameter: $q = \Pr(H = \text{TRUE})$.

PP-completeness of Constrained-MAP follows from a trivial modification of the PP-completeness proof of Inference as described by Kwisthout (2009, Lemma 2.7 and Lemma 2.9). Furthermore, given that the reductions from Constraint Satisfaction to MAP, value-approximate MAP, structure-approximate MAP, and rank-approximate MAP respect the same restrictions as imposed to Constrained-MAP, the necessity of bounded tree-width follows.

To show that $\{q\}-$Constrained-MAP is in FERT, for the parameter $q = \Pr(H = \text{TRUE})$, we give the following approximation algorithm. Observe that H is a binary sink node (i.e., has no outgoing edges) and that $\mathcal{B}$ has no evidence. A simple forward sampling strategy (Henrion, 1986) can approximate the distribution of H by sampling values for the variables in the network according to the probability distribution in the CPTs. We thus estimate $\Pr(H)$ by taking samples; we decide upon $approxsol_{\mathcal{B}}$ using this estimation. Note that the degree of error given a particular number of samples depends directly on the probability q. To be precise, using the Chernoff bound we can compute than the number of samples N needed to have a degree of error lower than δ is $1/(q - 1/2)^2 \ln 1/\sqrt{\delta}$. This gives us a fixed-parameter randomized tractable algorithm for parameter $\{q\}$.

Corollary 5.6 $\{q\}-$Constrained-MAP $\in \mathsf{FERT}$.

Another parameterized problem that can be shown to be fixed-error, fixed parameter randomized tractable is the Most Frugal Explanations heuristic approach to MAP, introduced by Kwisthout (2015). This heuristic (that either marginalizes or samples over intermediate variables, based on some subjective partition of these intermediate variables (into a set $\mathbf{I}^+$ and a set $\mathbf{I}^-$) according to their expected contribution to deciding upon the MAP explanation) can be expectation-approximated tractably when the tree-width of the network is low, the cardinality of the variables is small, the set $\mathbf{I}^+$ is small, and the probability distribution is such that a few samples over $\mathbf{I}^-$ suffice to decide upon the MFE explanation with high probability. The first three parameters ensure bounded running time, whereas

6. We will refer to such an algorithm as a *fixed-error randomized tractable* algorithm.

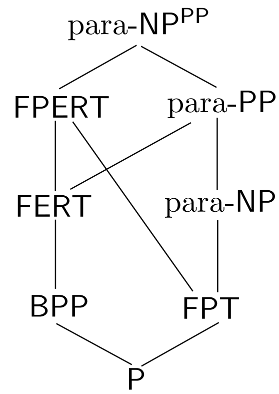

Figure 4: Inclusion properties for the complexity classes P, BPP, FERT, FPERT, FPT, para-NP, para-PP, and para-NP^{PP}

---

the final parameter ensures bounded probability of error. Hence, $\{\text{tw}, c, |\mathbf{I}^+|\} \cup \{b\}-$MFE is in FPERT, where tw denotes the tree-width of the network, c denotes the cardinality of the variables, $|\mathbf{I}^+|$ denotes the number of variables we marginalize (not sample) over, and b denotes some parameter describing a bias towards a particular explanation. In addition it can be shown that $\{|\mathbf{H}|, c, |\mathbf{I}^+|\} \cup \{b\}-$MFE is in FPERT, where $|\mathbf{H}|$ denotes the size of the explanation set.

Corollary 5.7 $\{\text{tw}, c, |\mathbf{I}^+|\} \cup \{b\}-$MFE $\in$ FPERT *and* $\{|\mathbf{H}|, c, |\mathbf{I}^+|\} \cup \{b\}-$MFE $\in$ FPERT.

5.3 The Relation Between FERT, FPERT, and Other Complexity Classes

The complexity class FERT introduced here is the randomized analog of FPT. Rather than parameterizing the *running time* (as an arbitrary function of κ and polynomially in the input size), here we parameterize *the probability of acceptance* of *Yes*-instances. BPP, FERT, and para-PP form natural analogs of P, FPT, and para-NP, respectively. The class FPERT parameterizes *both* the running time *and* the probability of acceptance, using two parameter sets κ_1 and κ_2. We thus have that BPP $\subseteq$ FERT $\subseteq$ PP, that FERT $\subseteq$ FPERT, and that FPT $\subseteq$ FPERT. Obviously, FPERT $\subseteq$ para-NP^{PP}, as every 'slice' $\{\kappa_1, \kappa_2\}-\Pi_{\kappa_1} \in$ NP^{PP} (see Flum & Grohe, 2006, for a discussion of 'slices' of parameterized problems). The inclusion relations are depicted in Figure 4.

It is conjectured that BPP = P (Clementi et al., 1998); however, it is not clear whether this conjecture can be transposed to the parameterized world; that is, whether it can be conjectured that FERT = FPT. It is known that $\{q, |\mathbf{I}|\}-$MAP and $\{q, \text{tw}\}-$MAP are fixed parameter tractable (Kwisthout, 2011); as Constrained-MAP is a special case of MAP, these results also hold for Constrained-MAP. However, in the intractability proof above neither $|\mathbf{I}|$ nor tw is bounded. To the best of our knowledge there is no parameterized complexity result known (in either direction) for $\{q\}-$Constrained-MAP.

5.4 Efficient MAP Approximation: A Road-map for Future Research

We have established that a particular, constrained version of MAP can be efficiently approximated under expectation-approximations if the probability of the MAP explanation is high (where the tree-width of the instance may be unbounded). A next step would be to investigate the parameterized approximability of current state-of-the-art approximation algorithms for MAP and show under which parameterization regimes these algorithms can be shown to be in FERT or FPERT.

From a different perspective, it is interesting to further explore the parameterization of the error in randomized algorithms and (for example) establish an analog for the W-hierarchy for these parameterizations. That allows us to derive more fine-grained negative parameterization results, in a similar way as proving W[1]-hardness leads to more fine-grained negative results than proving para-NP-hardness.

6. Conclusion

In this paper we analyzed whether low tree-width is a prerequisite for approximating MAP in Bayesian networks. We formalized four distinct notions of approximating MAP (by value, structure, rank, or expectation) and argued that approximate MAP is intractable in general using either of these notions. In case of value-approximation, structure-approximation, and rank-approximation we showed that MAP cannot be approximated using these notions in (non-trivial) instances with high tree-width, if the ETH holds. However, we showed that a constrained version of MAP, despite being PP-hard in general, can be tractably expectation-approximated when the most probable explanation has a high probability. We proposed the complexity classes FERT and FPERT that capture the parameterization of the error (respectively error and running time), rather than running time. With these results we contributed to a fuller understanding of what does and does not make state-of-the-art approximation algorithms for MAP feasible in practice.

7. Acknowledgements

A previous version of this paper (Kwisthout, 2014) was published in the Proceedings of the Seventh European Workshop on Probabilistic Graphical Models (PGM 2014). The author wishes to thank the workshop participants and (both sets of) anonymous reviewers for stimulating discussion and worthwhile suggestions. He thanks in particular Hans Bodlaender and Todd Wareham for valuable comments on an earlier version of this manuscript.

References

Abdelbar, A. M., & Hedetniemi, S. M. (1998). Approximating MAPs for belief networks is NP-hard and other theorems. *Artificial Intelligence*, *102*, 21–38.

Agrawal, M., Kayal, N., & Saxena, N. (2004). PRIMES is in P. *Annals of Mathematics*, *160*(2), 781–793.

Arora, S., & Barak, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge University Press.

Arvind, V., & Raman, V. (2002). Approximation algorithms for some parameterized counting problems. In Bose, P., & Morin, P. (Eds.), *Algorithms and Computation*, Vol. 2518 of *Lecture Notes in Computer Science*, pp. 453–464. Springer Berlin Heidelberg.

Cheng, J., & Druzdzel, M. (2000). AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, *13*(1), 155–188.

Clementi, A., Rolim, J., & Trevisan, L. (1998). Recent advances towards proving P=BPP. In Allender, E. (Ed.), *Bulletin of the EATCS*, Vol. 64. EATCS.

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

De Campos, C. P. (2011). New complexity results for MAP in Bayesian networks. In Walsh, T. (Ed.), *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pp. 2100–2106.

Downey, R. G., & Fellows, M. R. (1999). *Parameterized Complexity*. Springer Verlag, Berlin.

Flum, G., & Grohe, M. (2006). *Parameterized Complexity Theory*. Springer, Berlin.

Gill, J. T. (1977). Computational complexity of Probabilistic Turing Machines. *SIAM Journal on Computing*, *6*(4), 675–695.

Hamilton, M., Müller, M., van Rooij, I., & Wareham, H. (2007). Approximating solution structure. In Demaine, E., Gutin, G., Marx, D., & Stege, U. (Eds.), *Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs*, No. 07281 in Dagstuhl Seminar Proceedings.

Henrion, M. (1986). Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In Kanal, L., & Lemmer, J. (Eds.), *Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence*, pp. 149–164. New York: Elsevier Science.

Impagliazzo, R., & Paturi, R. (2001). On the complexity of k-SAT. *Journal of Computer and System Sciences*, *62*(2), 367 – 375.

Krentel, M. W. (1988). The complexity of optimization problems. *Journal of Computer and System Sciences*, *36*, 490–509.

Kwisthout, J. (2009). *The Computational Complexity of Probabilistic Networks*. Ph.D. thesis, Faculty of Science, Utrecht University, The Netherlands.

Kwisthout, J. (2011). Most probable explanations in Bayesian networks: Complexity and tractability. *International Journal of Approximate Reasoning*, *52*(9), 1452 – 1469.

Kwisthout, J. (2013). Structure approximation of most probable explanations in Bayesian networks. In van der Gaag, L. (Ed.), *Proceedings of the Twelfth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, Vol. 7958 of *LNAI*, pp. 340–351. Springer-Verlag.

Kwisthout, J. (2014). Treewidth and the computational complexity of MAP approximations. In van der Gaag, L., & Feelders, A. (Eds.), *Proceedings of the Seventh European Workshop on Probabilistic Graphical Models*, Vol. 8754 of *Lecture Notes in Computer Science*, pp. 271–285. Springer International Publishing.

Kwisthout, J. (2015). Most frugal explanations in Bayesian networks. *Artificial Intelligence*, *218*, 56 – 73.

Kwisthout, J., Bodlaender, H. L., & van der Gaag, L. C. (2010). The necessity of bounded treewidth for efficient inference in Bayesian networks. In Coelho, H., Studer, R., & Wooldridge, M. (Eds.), *Proceedings of the 19th European Conference on Artificial Intelligence*, pp. 237–242. IOS Press.

Kwisthout, J., Bodlaender, H. L., & van der Gaag, L. C. (2011). The complexity of finding kth most probable explanations in probabilistic networks. In Cerná, I., Gyimóthy, T., Hromkovic, J., Jefferey, K., Královic, R., Vukolic, M., & Wolf, S. (Eds.), *Proceedings of the 37th International Conference on Current Trends in Theory and Practice of Computer Science*, Vol. LNCS 6543, pp. 356–367. Springer.

Kwisthout, J., & van Rooij, I. (2013). Bridging the gap between theory and practice of approximate Bayesian inference. *Cognitive Systems Research*, *24*, 2–8.

Littman, M. L., Majercik, S. M., & Pitassi, T. (2001). Stochastic boolean satisfiability. *Journal of Automated Reasoning*, *27*(3), 251–296.

Marx, D. (2007). Can you beat treewidth?. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pp. 169–179.

Montoya, J.-A., & Müller, M. (2013). Parameterized random complexity.. *Theory of Computing Systems*, *52*(2), 221–270.

Papoulis, A. (1984). *Probability, Random Variables, and Stochastic Processes* (2nd edition). New York: McGraw-Hill.

Park, J. D., & Darwiche, A. (2001). Approximating MAP using local search. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pp. 403–410. Morgan Kaufmann Publishers, San Francisco, California, 2001.

Park, J. D., & Darwiche, A. (2004). Complexity results and approximation settings for MAP explanations. *Journal of Artificial Intelligence Research*, *21*, 101–133.

Robertson, N., & Seymour, P. (1986). Graph minors II: Algorithmic aspects of tree-width. *Journal of Algorithms*, *7*, 309–322.

Sontag, D., Meltzer, T., Globerson, A., Weiss, Y., & Jaakkola, T. (2008). Tightening LP relaxations for MAP using message-passing. In *Proceedings of the 24th Conference in Uncertainty in Artificial Intelligence*, pp. 503–510. AUAI Press.

Toda, S. (1994). Simple characterizations of P(#P) and complete problems. *Journal of Computer and System Sciences*, *49*, 1–17.

Van Rooij, I., & Wareham, H. (2012). Intractability and approximation of optimization theories of cognition. *Journal of Mathematical Psychology*, *56*(4), 232 – 247.

Yuan, C., Lu, T., & Druzdzel, M. J. (2004). Annealed MAP. In Chickering, D., & Halpern, J. (Eds.), *Proceedings of the Twentieth Conference in Uncertainty in Artificial Intelligence*, pp. 628–635. AUA.