Treewidth and the Computational Complexity of MAP Approximations

Johan Kwisthout

Radboud University Nijmegen, Donders Institute for Brain, Cognition and Behaviour, Montessorilaan 3, 6525 HR Nijmegen, The Netherlands, j.kwisthout@donders.ru.nl

Abstract. The problem of finding the most probable explanation to a designated set of variables (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 experiences, that low treewidth 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-, structure-, and rank-approximations of MAP instances with high treewidth will violate the Exponential Time Hypothesis. In contrast, we hint that expectation-approximation can be done efficiently, even in MAP instances with high treewidth, if the most probable explanation has a high probability.

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. 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 [17]. 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 [17, p. 113]. Even when all variables in the network are binary and the network has the (very restricted) polytree topology, MAP remains NP-hard [5]. Only when both the optimization *and* the inference part of the problem can be computed tractably (for example, if both the treewidth of the network and the cardinality of the variables are small *and* the most probable joint value assignment has a high probability) MAP can be computed tractably [11]. It is known that, for arbitrary probability distributions and under the assumption of the Exponential Time Hypothesis, a small treewidth of the moralized graph of a Bayesian network is a necessary condition for the inference problem to be tractable [13]; this result can easily be extended to MAP.

MAP is also intractable to approximate [1, 11, 12, 17]. While it is obviously the case that a particular instance to the MAP problem can be approximated efficiently when it can be efficiently computed exactly, 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 different notions of approximation as 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 in [13] we show in Section 4 that bounded treewidth is indeed a necessary condition for efficient value-, structure-, and rank-approximation of MAP; however, we show that MAP can sometimes be efficiently expectation-approximated, even on networks where the moralized graph has a high treewidth, if the most probable joint value assignment to the MAP variables has a high probability. We conclude the paper in Section 5.

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, treewidth, parameterized complexity theory, and the Exponential Time Hypothesis. For a more thorough discussion of these concepts, the reader is referred to textbooks such as [4], [3], and [6].

2.1 Bayesian Networks

A Bayesian network $\mathcal{B} = (\mathbf{G}_\mathcal{B}, \Pr)$ is a graphical structure that models 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 the 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 $\mathbf{G}_\mathcal{B}$. 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.

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., the joint value assignment to a designated set of variables (the explanation set) that has highest 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 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 $\{\mathcal{B}, \mathbf{H}, \mathbf{E}, \mathbf{e}\}$, let $cansol_\mathcal{B}$ denote a function returning candidate solutions to $\{\mathcal{B}, \mathbf{H}, \mathbf{E}, \mathbf{e}\}$, with $optsol_\mathcal{B}$ denoting a function returning the *optimal* solution (or, in case of a draw, one of the optimal solutions) to the MAP instance.

2.2 Treewidth

An important structural property of a Bayesian network $\mathcal{B}$ is its *treewidth*, which can be defined as the minimum width over all tree-decompositions of triangulations of the moralization $\mathbf{G}_\mathcal{B}^\mathbf{M}$ of the network. Treewidth plays an important role in the complexity analysis of Bayesian networks, as many otherwise intractable computational problems can be rendered tractable, provided that the treewidth 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 $\mathbf{G}_\mathcal{B}^\mathbf{M}$ is any chordal graph $\mathbf{G_T}$ that embeds $\mathbf{G}_\mathcal{B}^\mathbf{M}$ 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 [18] 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 $\mathbf{X_i} \cap \mathbf{X_k} \subseteq \mathbf{X_j}$. 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 treewidth tw of a Bayesian network $\mathcal{B}$ now is the minimum width over all possible tree-decompositions of triangulations of $\mathbf{G}_\mathcal{B}^\mathbf{M}$.

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, 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., $\epsilon = 1/c^n$). *Yes*-instances for problems in BPP are accepted with a probability that is polynomially bounded away from $1/2$, i.e., (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 NP $\subseteq$ 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 obviously BPP $\subseteq$ PP, the reverse is unlikely; in particular, it is conjectured that BPP = P.

The *Exponential Time Hypothesis* (ETH), introduced by [8], 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 P $\neq$ NP. A sub-exponential but not polynomial-time algorithm for 3SAT, such as an algorithm running in $O(2^{\sqrt[3]{n}})$, would contradict the ETH but would not imply that P = NP. We will assume the ETH in our proofs that show the necessity of low treewidth 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. Informally, a problem is called fixed-parameter tractable for a parameter k (or a set $\{k_1, \ldots, k_n\}$ of parameters) if it can be solved in time, exponential (or even worse) *only* in k and polynomial in the input size $|x|$, i.e., in time $\mathcal{O}(f(k) \cdot |x|^c)$ for a constant c and an arbitrary function f. In practice, this means that problem instances can be solved efficiently, even when the problem is NP-hard in general, if k is known to be small. In contrast, if a problem is NP-hard even when k is small, the problem is denoted as para-NP-hard for k.

3 Approximating MAP

It is widely known, both from practical experiences and from theoretical results, that 'small treewidth' is often a necessary constraint to render exact Bayesian

inferences tractable.[1] 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. Whether this observation has a firm theoretical basis, i.e., whether approximation algorithms can or cannot in principle perform well even in situations where treewidth 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-k solutions'. Note that these notions can refer to completely different solutions. One can have situations where the second-best solution does not resemble at all the optimal solution, whereas solutions that look almost the same have a very low value as compared to the optimal solution [12]. 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 a distribution using some sampling method we have no guarantee on how well the approximate distribution matches the original distribution (e.g., in terms of the Kullback-Leibler divergence); likely, we will (need to) settle for 'probably approximately correct' (PAC) approximations [19]. The added notion of approximation here, induced by the use of randomized computations, is the allowance of a bounded amount of error.

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 that has a value, close to the value of the optimal solution. This problem is intractable in

[1] An exception to this general observation might be algorithms that employ specific local structures, such as context-specific dependences, in the network, as one of the anonymous reviewers noted.

general, even if the variables of the network are bi-partitioned into explanation and evidence variables (i.e., when we approximate an MPE problem). Abdelbar and Hedetniemi proved that it is NP-hard in general to find an explanation $\mathbf{h} \in \mathit{cansol}_{\mathcal{B}}$ with a constant ratio bound $\frac{\Pr(\mathit{optsol}_{\mathcal{B}} \mid \mathbf{e})}{\Pr(\mathbf{h} \mid \mathbf{e})} \leq \rho$ for any constant $\rho \geq 1$ [1]. In addition, it can be shown that it is NP-hard in general to find an explanation $\mathbf{h} \in \mathit{cansol}_{\mathcal{B}}$ with $\Pr(\mathbf{h}, \mathbf{e}) > \epsilon$ for any constant $\epsilon > 0$ [11]. The latter result holds even for networks with only binary variables and at most two incoming arcs per variable.

3.2 Structure-approximation

Structure-approximating MAP is the problem of finding an explanation 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 [7]. For MAP, the typical structure distance function $d_H(\mathbf{h} \in \mathit{cansol}_{\mathcal{B}}, \mathit{optsol}_{\mathcal{B}})$ is the Hamming distance between explanation $\mathbf{h} \in \mathit{cansol}_{\mathcal{B}}$ and the most probable explanation $\mathit{optsol}_{\mathcal{B}}$. It has been shown in [12] that no algorithm can calculate the value of even a single variable in the most probable explanation in polynomial time, unless P = NP; that is, it is NP-hard to find an explanation with $d_H(\mathbf{h} \in \mathit{cansol}_{\mathcal{B}}, \mathit{optsol}_{\mathcal{B}}) \leq |\mathit{optsol}_{\mathcal{B}}| - 1$, even if the variables of the network are bi-partitioned into explanation and evidence variables.

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 as an explanation which is one of the k best explanations, for a constant k. 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 k most probable explanations. We will denote this approximation as a rank-approximation, and we will prove that it is NP-hard to approximate MAP using a rank-approximation for any constant k. We do so by a reduction from a variant of LexSat, based on the reduction in [14]. 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 $\mathsf{FP}^{\mathsf{NP}}$ [9]. In our proofs we will use the following variant that always returns a truth assignment (rather than $\perp$, in case ϕ is unsatisfiable):

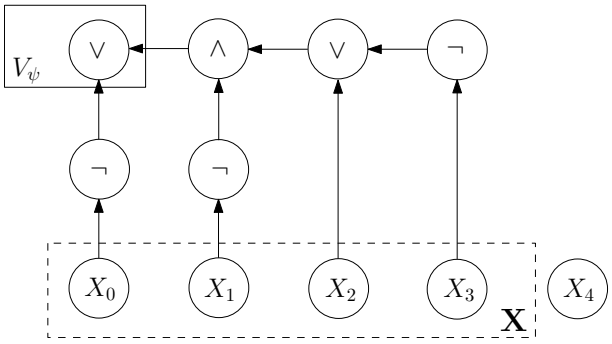

Fig. 1. Example construction of $\mathcal{B}_{\phi_{\text{ex}}}$ from LexSat' instance ϕ_{ex}

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 1. *No algorithm can k-rank-approximate MAP, for any constant k, in polynomial time, unless* $\mathsf{P} = \mathsf{NP}$.

In our proof we describe a polynomial-time Turing reduction from LexSat' to k-rank-approximated-MAP for an arbitrary constant k. The reduction largely follows the reduction as presented in [14] 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 $k = 3$ 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 k values $x^1_{n+1}, \ldots, x^k_{n+1}$. 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 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 $\Pr(X_1) \cdot \ldots \cdot \Pr(X_{i-1}) \cdot \overline{\Pr(X_i)} > \overline{\Pr(X_1)} \cdot \cdots \cdot \overline{\Pr(X_{i-1})} \cdot \Pr(X_i)$ 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. Observe that the k-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 α 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 k-th valued variable X_{n+1} into account, we have that for every $\mathbf{x}$, the k joint value assignments to $\Pr(\mathbf{x}, X_{n+1} \mid V_\psi = \text{TRUE})$ 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 k joint value assignments $\mathbf{x^k}$ 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 $\Pr(\mathbf{x^k} \mid V_\psi = \text{TRUE})$. Thus, any algorithm that returns one of the k-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 k-rank-approximate MAP, for any constant k, 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 k-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 k-rank-approximation-MPE is also $\mathsf{FP}^{\mathsf{NP}}$-hard.

3.4 Expectation-approximation

The last notion of MAP approximation we will discuss here returns in polynomial time an explanation that are 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* [15]–is particularly relevant for typical Bayesian approximation methods, such as Monte Carlo sampling and repeated local search algorithms.

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. Monte Carlo randomized algorithms are in the complexity class BPP; randomized algorithms 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.[2] On various grounds it is believed that $\mathsf{P} = \mathsf{BPP}$, 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.

4 The Necessity of Low Treewidth 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, $\{\mathrm{tw}, c, 1-p\}$-MAP is in FPT for parameters treewidth (tw), cardinality of the variables (c), and probability of the most probable solution $1-p$. Surely, if we can compute $\{k_1, \ldots, k_m\}$-MAP exactly in FPT time, we can also approximate $\{k_1, \ldots, k_m\}$-MAP in FPT time.

[2] 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 $\in \mathsf{BPP}$), only fairly recently it has been proven that PRIMES is in P [2].

A question remains, however, whether approximate MAP can be fixed-parameter tractable for a *different* set of parameters than exact MAP.

Treewidth has been shown to be a *necessary* parameter for efficient exact computation of the INFERENCE problem (and, by a trivial adjustment, also of MAP), under the assumption that the ETH holds [13]. In this section, we will show that low treewidth is also a necessary parameter for efficient *approximate* computation for value-, structure-, and rank-approximations. We also show that it is *not* a necessary parameter for efficient expectation-approximation. In the next sub-section we will review so-called treewidth-preserving reductions (tw-reductions), a special kind of polynomial many-one reduction that preserves treewidth of the instances [13]. In Subsection 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 treewidth cannot have sub-exponential algorithms, unless the ETH fails [16], it was established in [13] that there cannot be a polynomial-time algorithm that decides INFERENCE on instances with high treewidth in sub-exponential time, unless the ETH fails; the reader is referred to [13] for the full proof.

Subsequently, 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). In the last sub-section we will give a small example where a simple forward-sampling algorithm can efficiently expectation-approximate MAP despite high treewidth; we will elaborate on the constraints needed to render such algorithms provably fixed-parameter tractable and give pointers for future work.

4.1 Treewidth-preserving Reductions

Treewidth-preserving reductions are defined in [13] as a means to reduce CONSTRAINT SATISFACTION to INFERENCE while ensuring that treewidth is preserved between instances in the reduction, modulo a linear factor.

Definition 1 ([13]). *Let A and B be computational problems such that* treewidth *is defined on instances of both A and B. We say that A is* polynomial-time treewidth-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 in [13], 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

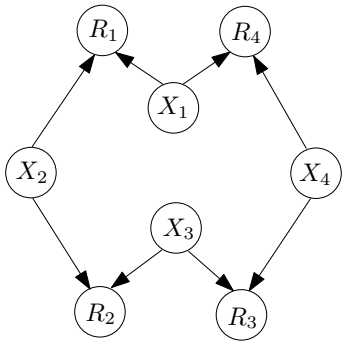

Fig. 2. Example construction of $\mathcal{B}_\mathcal{I}$ from example CSP instance $\mathcal{I}$

of $\mathcal{I}$, $\mathbf{D}$ denotes the set of values of these variables, and $\mathbf{C}$ denotes the set of binary constraints defined over $\mathbf{V} \times \mathbf{V}$. 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., $\Pr(R_j = \text{TRUE} \mid \mathbf{x} \in \Omega(\pi(R_j))) = 1$ 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 [13], 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 treewidth of the thus obtained network equals $\max(2, \text{tw}(\mathbf{G}_\mathcal{I}))$, where $\mathbf{G}_\mathcal{I}$ is the primal graph of $\mathcal{I}$; note that the treewidth of $\mathcal{B}_\mathcal{I}$ at most increases the treewidth 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 most be taken not to blow up treewidth 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 treewidth of the resulting graph is at most the treewidth of $\mathbf{G}_\mathcal{I}$ plus 3. The conditional probability distribution of the nodes A_k is defined as follows.

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

For a node A_k without any parents, $\Pr(A_k = \text{TRUE}) = 1$. The graph that results from applying this procedure to the example is given in Figure 3 (also taken from [13]). Now, $\Pr(A_1 = \text{TRUE} \mid \mathbf{x}) = 1$ if $\mathbf{x}$ corresponds to a satisfying value

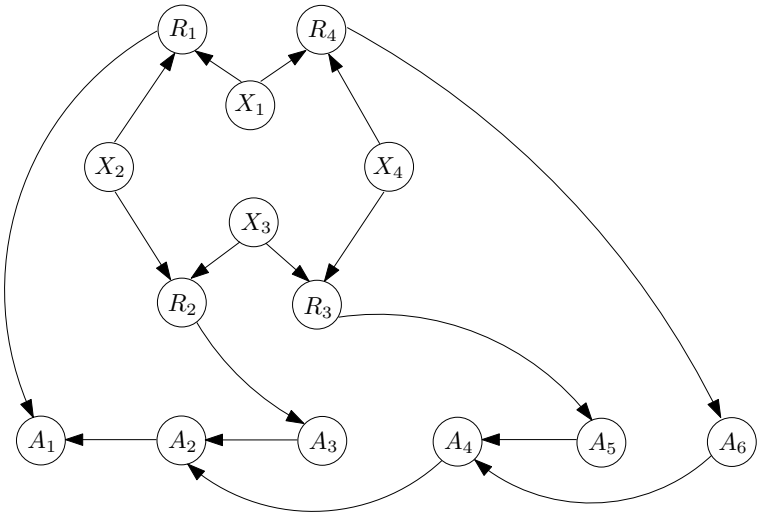

Fig. 3. Resulting graph $\mathcal{B}_\mathcal{I}$ after adding nodes A_k and appropriate arcs

assignment to $\mathbf{V}$ and 0 otherwise; correspondingly, $\Pr(A_1 = \text{TRUE}) > 0$ 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 be modified to a tw-reduction from CONSTRAINT SATISFACTION to MAP. We do this by adding a binary node $V_\mathcal{I}$ 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 treewidth 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-, structure-, and rank-approximations can be tw-reduced from CONSTRAINT SATISFACTION, as sketched below.

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 with explanation set $\mathbf{H} = A_1$ and evidence $\mathbf{e} = V_{\mathcal{I}} = \text{TRUE}$ that can return a solution $\mathbf{h} \in \mathit{cansol}_{\mathcal{B}}$ with $\Pr(\mathbf{h}, \mathbf{e}) > \epsilon$ for any constant $\epsilon > 0$, effectively solves CONSTRAINT SATISFACTION. Given that we added a single node to $\mathcal{B}_{\mathcal{I}}$, with A_1 as only parent, this increases the treewidth of $\mathcal{B}_{\mathcal{I}}$ by at most 1. Hence, CONSTRAINT SATISFACTION tw-reduces to value-approximate MAP.

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 with $d_H(\mathbf{h} \in \mathit{cansol}_{\mathcal{B}}, \mathit{optsol}_{\mathcal{B}}) \leq |\mathit{optsol}_{\mathcal{B}}| - 1 = 0$ 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 treewidth, 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, the treewidth of $\mathcal{B}_{\mathcal{I}}$ increases by at most k. Hence, CONSTRAINT SATISFACTION tw-reduces to structure-approximate MAP.

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 a uniformly distributed unconnected node $K_{\mathcal{I}}$ with k values to $\mathbf{H}$; a k-rank-approximate MAP query with explanation set $\mathbf{H} = \{V_{\mathcal{I}}, K_{\mathcal{I}}\}$ will return $V_{\mathcal{I}} = \text{TRUE}$ (and $K_{\mathcal{I}}$ set to an arbitrary value) if and only if $\mathcal{I}$ is satisfiable. The addition of $K_{\mathcal{I}}$ does not increase treewidth, hence, CONSTRAINT SATISFACTION tw-reduces to k-rank-approximate MAP.

4.5 Expectation-approximation

In the previous section we showed that we cannot value-, structure-, or rank-approximate MAP on instances with high treewith, unless the ETH fails. Now what about expectation-approximation? We will argue that there are MAP instances with high treewidth that *can* be efficiently expectation-approximated,

provided that the probability $\Pr(\mathit{optsol}_{\mathcal{B}} \mid \mathbf{e})$ is high. Note that it remains NP-hard (to be precise: para-PP-hard) to decide INFERENCE, even if the probability of interest is arbitrarily close to 1 [10]; as INFERENCE is a degenerate special case of MAP, it follows that computing MAP exactly is also NP-hard in this case. While this sketchy argument is not a fully worked-out proof, it hints that efficient expectation-approximation of MAP indeed depends on a *different* set of parameters than the other notions of approximation discussed above.

The argument goes as follows. In order to generate MAP instances with high treewidth, we construct them from SAT instances in a similar way as described in Section 3.3. We can generate SAT instances with high treewidth by, e.g., picking an arbitrary formula ϕ and then boosting the treewidth by "inserting" tautologies $\wedge(x_i \vee \neg x_i)$ at strategic places in ϕ. We then construct a Bayesian network $\mathcal{B}_\phi$ from ϕ by including binary root *truth-setting* variables X_i for all variables X_i in ϕ, and adding binary *operator* variables T_j for all logical operators in ϕ, and connecting them as described in Section 3.3. We again denote the top-level operator as V_ϕ and we observe that $\Pr(V_\phi = \text{TRUE}) = \#SAT/2^n$, i.e., the probability distribution over $\Pr(V_\phi)$ corresponds to the number of satisfying truth assignments to ϕ. If a majority of truth assignments satisfy ϕ, a MAP query with $\mathbf{H} = V_\phi$ will return TRUE, if a minority of truth assignments satisfy ϕ, the same MAP query will return FALSE. Now, if the probability p of the most probable joint value assignment is bounded away from $1/2$, i.e., is guaranteed to be $1/2 + 1/n^c$ for a constant c, a simple forward sampling strategy (assigning random joint value assignments to the variables X_i and propagating the assignments according to the CPTs of the operator variables T_j) can decide this MAP query with a bounded degree of error. To be precise, using the Chernoff bound we can compute than the number of samples needed to have a degree of error lower than δ is $1/(p - 1/2)^2 \ln 1/\sqrt{\delta} = n^{c^2} \ln 1/\sqrt{\delta}$.

5 Conclusion

In this paper we analysed whether low treewidth 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-, structure-, and rank-approximation we showed that MAP cannot be approximated using these notions in instances with high treewidth, if the ETH holds. We argued that expectation-approximation, in contrast, may be rendered fixed-parameter tractable, even in instances with high treewidth, if the probability q of the most probable explanation is high (and the cardinality c of the variables is bounded). As INFERENCE (and thus also MAP) is intractable even when the probability of the most probable explanation is high, this result may indeed lead to a $\{q, c\}$-fixed parameter tractable expectation-approximation algorithm for MAP. We leave the proof of existence and the actual development and analysis of such an algorithm for future work.

References

1. A. M. Abdelbar and S. M. Hedetniemi. Approximating MAPs for belief networks is NP-hard and other theorems. *Artificial Intelligence*, 102:21–38, 1998.
2. M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004.
3. S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.
4. A. Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge University Press, 2009.
5. C. P. De Campos. New complexity results for MAP in Bayesian networks. In T. Walsh, editor, *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, pages 2100–2106, 2011.
6. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer Verlag, Berlin, 1999.
7. M. Hamilton, M. Müller, I. van Rooij, and H.T. Wareham. Approximating solution structure. In E. Demaine, G.Z. Gutin, D. Marx, and U. Stege, editors, *Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs*, number 07281 in Dagstuhl Seminar Proceedings, 2007.
8. R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367 – 375, 2001.
9. M. W. Krentel. The complexity of optimization problems. *Journal of Computer and System Sciences*, 36:490–509, 1988.
10. J. Kwisthout. The computational complexity of probabilistic inference. Technical Report ICIS–R11003, Radboud University Nijmegen, 2011.
11. J. Kwisthout. Most probable explanations in Bayesian networks: Complexity and tractability. *International Journal of Approximate Reasoning*, 52(9):1452 – 1469, 2011.
12. J. Kwisthout. Structure approximation of most probable explanations in Bayesian networks. In L.C. van der Gaag, editor, *Proceedings of the Twelfth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, volume 7958 of *LNAI*, pages 340–351. Springer-Verlag, 2013.
13. J. Kwisthout, H. L. Bodlaender, and L. C. van der Gaag. The necessity of bounded treewidth for efficient inference in Bayesian networks. In H. Coelho, R. Studer, and M. Wooldridge, editors, *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI'10)*, pages 237–242. IOS Press, 2010.
14. J. Kwisthout, H. L. Bodlaender, and L. C. van der Gaag. The complexity of finding kth most probable explanations in probabilistic networks. In I. Cerná, T. Gyimóthy, J. Hromkovic, K. Jefferey, R. Královic, M. Vukolic, and S. Wolf, editors, *Proceedings of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM 2011)*, volume LNCS 6543, pages 356–367. Springer, 2011.
15. J. Kwisthout and I. van Rooij. Bridging the gap between theory and practice of approximate Bayesian inference. *Cognitive Systems Research*, 24:2–8, 2013.
16. D. Marx. Can you beat treewidth? In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 169–179, 2007.
17. J. D. Park and A. Darwiche. Complexity results and approximation settings for MAP explanations. *Journal of Artificial Intelligence Research*, 21:101–133, 2004.
18. N. Robertson and P.D. Seymour. Graph minors II: Algorithmic aspects of treewidth. *Journal of Algorithms*, 7:309–322, 1986.

19. L.G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.