The Complexity of Finding kth Most Probable Explanations in Probabilistic Networks*

Johan H.P. Kwisthout, Hans L. Bodlaender, and Linda C. van der Gaag

Department of Information and Computing Sciences,
Utrecht University, Utrecht, The Netherlands

Abstract. In modern decision-support systems, probabilistic networks model uncertainty by a directed acyclic graph quantified by probabilities. Two closely related problems on these networks are the Kth MPE and Kth Partial MAP problems, which both take a network and a positive integer k for their input. In the Kth MPE problem, given a partition of the network's nodes into evidence and explanation nodes and given specific values for the evidence nodes, we ask for the kth most probable combination of values for the explanation nodes. In the Kth Partial MAP problem in addition a number of unobservable intermediate nodes are distinguished; we again ask for the kth most probable explanation. In this paper, we establish the complexity of these problems and show that they are $\mathsf{FP}^{\mathsf{PP}}$- and $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-complete, respectively.

1 Introduction

For modern decision-support systems, probabilistic networks are rapidly becoming the models of choice for representing and reasoning with uncertainty. Applications of these networks have been developed for a range of problem domains which are fraught with uncertainty. Most notably, applications are being realised in the biomedical field where they are designed to support medical and veterinary practitioners in their diagnostic reasoning processes; examples from our own engineering experiences include a network for diagnosing ventilator-associated pneumonia in critically ill patients [1] and a network for the early detection of an infection with the Classical Swine Fever virus in pigs [2].

A probabilistic network is a concise model of a joint probability distribution over a set of stochastic variables [3]. It consists of a directed acyclic graph, encoding the relevant variables and their probabilistic interdependencies, and an associated set of conditional probabilities. Various algorithms have been designed for probabilistic inference, that is, for computing probabilities of interest from a probabilistic network. These algorithms typically exploit structural properties of the network's graph to decompose the computations involved. Probabilistic inference is known to be PP-complete in general [4]. Many other problems to be

* This work was supported by the Netherlands Organisation for Scientific Research (NWO), project *Complexity of Probabilistic Networks*.

I. Černá et al. (Eds.): SOFSEM 2011, LNCS 6543, pp. 356–367, 2011.

solved in practical applications of probabilistic networks are also known to have a quite unfavourable complexity.

In many practical applications, the nodes of a probabilistic network are partitioned into evidence nodes, explanation nodes and intermediate nodes. The evidence nodes model variables whose values can be observed in reality; in a medical application, these nodes typically model a patient's observable symptoms. The explanation nodes model the variables for which a most likely value needs to be found; these nodes typically capture possible diagnoses. The intermediate nodes are included in the network to correctly represent the probabilistic dependencies among the variables; in a medical application, these nodes often model physiological processes hidden in a patient. An important problem in probabilistic networks now is to find the most likely value combination for the explanation nodes given a specific joint value for the evidence nodes. When the network's set of intermediate nodes is empty, the problem is known as the most probable explanation, or MPE, problem; the problem is coined the partial maximum aposteriori probability, or Partial MAP, problem otherwise. The MPE problem is known to have various NP-complete decision variants [5,6]; for the Partial MAP problem $\mathsf{NP}^{\mathsf{PP}}$-completeness was established [7].

In many applications of probabilistic networks, the user is interested not just in finding the most likely explanation for a combination of observations, but in finding alternative explanations as well. In biomedicine, for example, a practitioner may wish to start antibiotic treatment for multiple likely pathogens before the actual cause of infection in a patient is known; alternative explanations may also reveal whether or not further diagnostic testing can help distinguishing between possible diagnoses. In the absence of intermediate nodes in a network, the problem of finding the kth most likely explanation is known as the Kth MPE problem; it is called the Kth Partial MAP problem otherwise. While efficient algorithms have been designed for solving the kth most probable explanation problem with the best explanation as additional input [8], the Kth MPE problem without this extra information is NP-hard in general [9]. The complexity of the Kth Partial MAP problem is unknown as yet.

In this paper, we study the computational complexity of the Kth MPE and Kth Partial MAP problems in probabilistic networks and show that these problems are complete for the complexity classes $\mathsf{FP}^{\mathsf{PP}}$ and $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$, respectively. This finding suggests that the two problems are much harder than the (already intractable) restricted problems of finding a most likely explanation. Finding the kth most probable explanation in a probabilistic network given partial evidence to our best knowledge is the first problem with a practical application that is shown to be $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-complete, which renders our result interesting at least from a theoretical point of view.

The paper is organised as follows. In Section 2, our notational conventions are introduced, as are the definitions used in the paper. We discuss the computational complexity of finding kth joint value assignments with full and partial evidence in the Sections 3 and 4, respectively. Section 5 concludes the paper.

2 Definitions

In this section, we provide definitions for the concepts used in the sequel. In Section 2.1, we briefly review probabilistic networks and introduce our notational conventions. In Section 2.2, we describe the problems under study. In Section 2.3, we review some complexity classes and state complete problems for these classes.

2.1 Probabilistic Networks

A probabilistic network is a model of a joint probability distribution over a set of stochastic variables. Before defining the concept of probabilistic network more formally, we introduce some notational conventions. Stochastic variables are denoted by capital letters with a subscript, such as X_i; we use bold-faced upper-case letters $\mathbf{X}$ to denote sets of variables. A lower-case letter x is used for a value of a variable X; a combination of values for a set of variables $\mathbf{X}$ is denoted by a bold-faced lower-case letter $\mathbf{x}$ and will be termed a joint value assignment to $\mathbf{X}$. In the sequel, we assume that all joint value assignments to a set of variables $\mathbf{X}$ are uniquely ordered by their (possibly posterior) probability in numerically descending order; if two joint value assignments $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same probability, they are ordered lexicographically in descending order by their respective binary representation, taking the value for the variable X_1 to be the most significant element.

A probabilistic network now is a tuple $\mathcal{B} = (\mathbf{G}, \Gamma)$ where $\mathbf{G} = (\mathbf{V}, A)$ is a directed acyclic graph and Γ is a set of conditional probability distributions. Each node $V_i \in \mathbf{V}$ models a stochastic variable. The set of arcs A of the graph captures probabilistic independence: two nodes V_i and V_j are independent given a set of nodes $\mathbf{W}$, if either V_i or V_j is in $\mathbf{W}$, or if every chain between V_i and V_j in $\mathbf{G}$ contains a node from $\mathbf{W}$ with at least one emanating arc or a node V_k with two incoming arcs such that neither V_k itself nor any of its descendants are in $\mathbf{W}$. For a topological sort $V_1, \ldots, V_n$ of $\mathbf{G}$, we now have that any node V_i is independent of the preceding nodes $V_1, \ldots, V_{i-1}$ given its set of parents $\pi(V_i)$. The set Γ of the network includes for each node V_i the conditional probability distributions $\Pr(V_i \mid \pi(V_i))$ that describe the influence of the various assignments to V_i's parents $\pi(V_i)$ on the probabilities of the values of V_i itself.

A probabilistic network $\mathcal{B} = (\mathbf{G}, \Gamma)$ defines a joint probability distribution $\Pr(\mathbf{V}) = \prod_{V_i \in \mathbf{V}} \Pr(V_i \mid \pi(V_i))$ that respects the independencies portrayed by its digraph. Since it defines a unique distribution, a probabilistic network allows the computation of any probability of interest over its variables [10].

2.2 The *k*th Most Probable Explanation Problems

The main problem studied in this paper is the problem of finding a kth most probable explanation for a particular combination of observations, for arbitrary values of k. Formulated as a functional problem, it is defined as follows.

KTH MPE
Instance: A probabilistic network $\mathcal{B} = (\mathbf{G}, \Gamma)$, where $\mathbf{V}$ is partitioned into a set of evidence nodes $\mathbf{E}$ and a set of explanation nodes $\mathbf{M}$; a joint value assignment $\mathbf{e}$ to $\mathbf{E}$; and a positive natural number k.
Output: The kth most probable joint value assignment $\mathbf{m}_k$ to $\mathbf{M}$ given $\mathbf{e}$; if no such assignment exists, the output is $\perp$, that is, the universal *false*.

Note that the KTH MPE problem defined above includes the MPE problem as a special case with $k = 1$. From $\Pr(\mathbf{m} \mid \mathbf{e}) = \frac{\Pr(\mathbf{m},\mathbf{e})}{\Pr(\mathbf{e})}$, we further observe that $\Pr(\mathbf{e})$ can be regarded a constant if we are interested in the relative order only of the conditional probabilities $\Pr(\mathbf{m} \mid \mathbf{e})$ of all joint value assignments $\mathbf{m}$.

While for the KTH MPE problem a network's nodes are partitioned into evidence and explanation nodes only, the KTH PARTIAL MAP problem discerns also intermediate nodes. We define a bounded variant of the latter problem.

BOUNDED KTH PARTIAL MAP
Instance: A probabilistic network $\mathcal{B} = (\mathbf{G}, \Gamma)$, where $\mathbf{V}$ is partitioned into a set of evidence nodes $\mathbf{E}$, a set of intermediate nodes $\mathbf{I}$, and a set of explanation nodes $\mathbf{M}$; a joint value assignment $\mathbf{e}$ to $\mathbf{E}$; a positive natural number k; and rational numbers a, b with $0 \leq a \leq b \leq 1$.
Output: The tuple $(\mathbf{m}_k, p_k)$, where $\mathbf{m}_k$ is the kth most probable assignment to $\mathbf{M}$ given $\mathbf{e}$ from among all joint value assignments $\mathbf{m}_i$ to $\mathbf{M}$ with $p_i = \Pr(\mathbf{m}_i, \mathbf{e}) \in [a, b]$; if no such assignment exists, the output is $\perp$.

Note that the original KTH PARTIAL MAP problem without bounds is a special case of the problem defined above with $a = 0$ and $b = 1$. Further note that the bounded problem can be transformed into a problem without bounds in polynomial time and vice versa, which renders the two problems Turing equivalent. In the sequel, we will use the bounded problem to simplify our proofs.

2.3 Complexity Classes and Complete Problems

We assume throughout the paper that the reader is familiar with the standard notion of a Turing machine and with the basic concepts from complexity theory. We further assume that the reader is acquainted with complexity classes such as $\mathsf{NP}^{\mathsf{PP}}$, for which certificates of membership can be verified in polynomial time given access to an oracle. For these classes, we recall that the defining Turing machine can write a string to an oracle tape and takes the next step conditional on whether or not the string on this tape belongs to a particular language; for further details on complexity classes involving oracles, we refer to [11,12,13].

While Turing machines are tailored to solving decision problems, halting either in an accepting state or in a rejection state, Turing transducers can generate functional results: if a Turing transducer halts in an accepting state, it returns a result on an additional output tape. The complexity classes FP and FNP now are the functional variants of P and NP, and are defined using Turing transducers instead of Turing machines. Just like a Turing machine, a Turing transducer can have access to an oracle; for example, $\mathsf{FP}^{\mathsf{NP}}$ is the class of functions computable in polynomial time by a Turing transducer with access to an NP oracle. Since the kth most probable explanation problems under study require the computation of a result, we will use Turing transducers in the sequel.

Metric Turing machines are used to show membership in complexity classes like P^{NP} or P^{PP} [12]. A metric Turing machine $\hat{\mathcal{M}}$ is a polynomial-time bounded non-deterministic Turing machine in which every computation path halts with a binary number on a designated output tape. $\text{Out}_{\hat{\mathcal{M}}}(x)$ denotes the set of outputs of $\hat{\mathcal{M}}$ on input x; $\text{Opt}_{\hat{\mathcal{M}}}(x)$ is the smallest number in $\text{Out}_{\hat{\mathcal{M}}}(x)$, and $\text{KthValue}_{\hat{\mathcal{M}}}(x,k)$ is the k-th smallest number in $\text{Out}_{\hat{\mathcal{M}}}(x)$. Metric Turing transducers $\hat{\mathcal{T}}$ are defined likewise as Turing transducers with an additional output tape; these transducers are used for proving membership in $\mathsf{FP}^{\mathsf{NP}}$ or $\mathsf{FP}^{\mathsf{PP}}$.

A function f is polynomial-time one-Turing reducible to a function g, written $f \leq^{\mathsf{FP}}_{\text{1-T}} g$, if there exist polynomial-time computable functions T_1 and T_2 such that $f(x) = T_1(x, g(T_2(x)))$ for every x [13]. A function f now is in $\mathsf{FP}^{\mathsf{NP}}$ if and only if there exists a metric Turing transducer $\hat{\mathcal{T}}$ such that $f \leq^{\mathsf{FP}}_{\text{1-T}} \text{Opt}_{\hat{\mathcal{T}}}$. Correspondingly, a set L is in P^{NP} if and only if a metric Turing machine $\hat{\mathcal{M}}$ can be constructed, such that $\text{Opt}_{\hat{\mathcal{M}}}(x)$ is odd if and only if $x \in L$. Similar observations hold for $\mathsf{FP}^{\mathsf{PP}}$ and P^{PP}, and the $\text{KthValue}_{\hat{\mathcal{M}}}$ and $\text{KthValue}_{\hat{\mathcal{T}}}$ functions [12,13]. $\mathsf{FP}^{\mathsf{NP}}$- and $\mathsf{FP}^{\mathsf{PP}}$-hardness can be proved by a reduction from a known $\mathsf{FP}^{\mathsf{NP}}$- and $\mathsf{FP}^{\mathsf{PP}}$-hard problem, respectively, using a polynomial-time one-Turing reduction.

We define two functional variants of the well-known satisfiability problem and state their completeness in terms of functional complexity classes.

KTH SAT
Instance: A Boolean formula $\phi(X_1, \ldots, X_n), n \geq 1$; and a positive natural number k.
Output: The lexicographically kth largest truth assignment $\mathbf{x}_k$ to $\mathbf{X} = \{X_1, \ldots, X_n\}$ that satisfies ϕ; if no such assignment exists, the output is $\perp$.

The LEXSAT problem is the special case of the KTH SAT problem with $k = 1$. LEXSAT and KTH SAT are complete for $\mathsf{FP}^{\mathsf{NP}}$ and $\mathsf{FP}^{\mathsf{PP}}$, respectively [12,13].

KTHNUMSAT
Instance: A Boolean formula $\phi(X_1, \ldots, X_m, \ldots, X_n), m \leq n, n \geq 1$; and positive natural numbers k, l.
Output: The lexicographically kth largest assignment $\mathbf{x}_k$ to $\{X_1, \ldots, X_m\}$ for which exactly l assignments $\mathbf{x}_l$ to $\{X_{m+1}, \ldots, X_n\}$ satisfy ϕ; the output is $\perp$ if no such assignment exists.

The LEXNUMSAT problem is the special case of the KTHNUMSAT problem with $k = 1$. KTHNUMSAT and LEXNUMSAT are $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$- and $\mathsf{FP}^{\mathsf{NP}^{\mathsf{PP}}}$-complete; proofs will be provided in a full paper [14].

To facilitate the reductions in our proofs in the sequel, we further introduce slightly modified variants of the KTH SAT and KTHNUMSAT problems which serve to circumvent the need of explicitly dealing with outputs equal to $\perp$.

KTH SAT′
Instance: A Boolean formula $\phi(X_1, \ldots, X_n), n \geq 1$, with

$$\phi(X_1, X_2, \ldots, X_n) = (\neg X_1) \vee \phi'(X_2, \ldots, X_n)$$

for some Boolean formula $\phi'(X_2, \ldots, X_n)$; and a positive natural number $k \leq 2^{n-1}$.
Output: The lexicographically kth largest truth assignment $\mathbf{x}_k$ to $\mathbf{X} = \{X_1, \ldots, X_n\}$ that satisfies ϕ.

Note that the KTH SAT′ problem differs from the original KTH SAT problem only in that it never has the output $\perp$ for its solution: since the problem's formula ϕ has at least 2^{n-1} satisfying truth assignments, a kth largest satisfying assignment is guaranteed to exist for any value of k with $k \leq 2^{n-1}$. For the original KTH SAT problem on the other hand, the number of values k for which a satisfying truth assignment is returned, depends on the precise formula ϕ. For the KTH SAT′ problem we observe moreover that, within the descending lexicographic order, all satisfying truth assignments that set X_1 to *false* come after all assignments with $X_1 = \mathit{true}$ that satisfy ϕ'. We further note that a simple transformation suffices to show that KTH SAT′ is complete for the complexity class $\mathsf{FP}^{\mathsf{PP}}$.

Using a similar yet slightly more involved construction, we pose a variant of the KTHNUMSAT problem which never has the output $\perp$ for its solution.
KTHNUMSAT′
Instance: A Boolean formula $\phi(X_1, \ldots, X_m, \ldots, X_n)$, $m \leq n$, $n \geq 1$, with

$$\phi(X_1, \ldots, X_n) = \phi'(X_2, \ldots, X_n) \vee ((\neg X_1) \wedge \psi^l(X_{m+1}, \ldots, X_n))$$

for some Boolean formula $\phi'(X_2, \ldots, X_n)$ and terms $\psi^l(X_{m+1}, \ldots, X_n)$ which express that $X_{m+1} \cdots X_n$ seen as a binary number is at most l; and positive natural numbers $k \leq 2^{m-1}$, $l \leq 2^{n-m-1}$.
Output: The lexicographically kth largest assignment $\mathbf{x}_k$ to $\{X_1, \ldots, X_m\}$ with which exactly l assignments $\mathbf{x}_l$ to $\{X_{m+1}, \ldots, X_n\}$ satisfy ϕ.

Note that each term ψ^l in the Boolean formula above has exactly l satisfying truth assignments. We further note that a simple transformation again suffices to show that the KTHNUMSAT′ problem is complete for $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$.

3 Complexity of Kth MPE

We study the complexity of the KTH MPE problem introduced in Section 2.2 and prove $\mathsf{FP}^{\mathsf{PP}}$-completeness. To prove membership of $\mathsf{FP}^{\mathsf{PP}}$, we show that the problem can be solved in polynomial time by a metric Turing transducer; we prove hardness by a reduction from the KTH SAT′ problem defined above.

We begin by describing the construction of a probabilistic network $\mathcal{B}_\phi$ from the Boolean formula ϕ of an instance of the KTH SAT′ problem; upon doing so, we use the formula $\phi_{\mathrm{ex}} = (\neg X_1) \vee (\neg X_2 \wedge (X_3 \vee \neg X_4))$ for our running example. For each Boolean variable X_i in ϕ, we include a root node X_i in the network

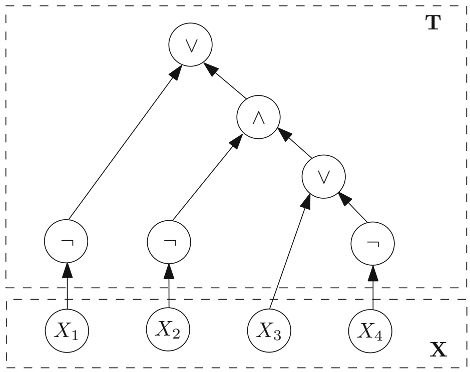

Fig. 1. The acyclic directed graph of the probabilistic network $\mathcal{B}_{\phi_{\rm ex}}$ constructed from the Boolean formula $\phi_{\rm ex} = (\neg X_1) \vee (\neg X_2 \wedge (X_3 \vee \neg X_4))$

$\mathcal{B}_\phi$, with *true* and *false* for its possible values; the nodes X_i with each other are called the variable-instantiation part $\mathbf{X}$ of the network. The prior probabilities $p_i = \Pr(X_i = \mathit{true})$ for the nodes X_i are chosen such that the prior probability of a joint value assignment $\mathbf{x}$ to $\mathbf{X}$ is higher than that of $\mathbf{x}'$ if and only if the corresponding truth assignment $\mathbf{x}$ to the KTH SAT′ variables $X_1, \ldots, X_n$ is ordered before $\mathbf{x}'$ in descending lexicographic order. More specifically, we set $p_i = \frac{1}{2} + \frac{2^{n+1-i}-1}{2^{n+1}}$; in our running example with four variables, the prior probabilities for the nodes $X_1, \ldots, X_4$ thus are set to $p_1 = \frac{31}{32}$, $p_2 = \frac{23}{32}$, $p_3 = \frac{19}{32}$, and $p_4 = \frac{17}{32}$. We observe that we have that $p_1 \cdot \ldots \cdot p_{i-1} \cdot \overline{p_i} > \overline{p_1} \cdot \cdots \cdot \overline{p_{i-1}} \cdot p_i$ for every i. Since the root nodes X_i are modelled as being mutually independent, the ordering property stated above is thereby satisfied in the network under construction. Further note that the assigned probabilities can be formulated using a number of bits which is polynomial in the number of variables of the KTH SAT′ instance.

For each logical operator in the Boolean formula ϕ, we create an additional node in the network $\mathcal{B}_\phi$. The parents of this node are the nodes corresponding with the subformulas joined by the operator; its conditional probability table is set to mimic the operator's truth table. The node associated with the top-level operator of ϕ will be denoted by V_ϕ. The operator nodes with each other constitute the truth-setting part $\mathbf{T}$ of the network. The probabilistic network $\mathcal{B}_{\phi_{\rm ex}}$ that is constructed from the example formula $\phi_{\rm ex}$ is shown in Figure 1. From the above construction, it is now readily seen that, given a value assignment $\mathbf{x}$ to the variable-instantiation part of the network, we have $\Pr(V_\phi = \mathit{true} \,|\, \mathbf{x}) = 1$ if and only if the truth assignment $\mathbf{x}$ to the Boolean variables X_i satisfies ϕ.

Theorem 1. KTH MPE *is* $\mathsf{FP}^{\mathsf{PP}}$*-complete.*

Proof. To prove membership of $\mathsf{FP}^{\mathsf{PP}}$ for the KTH MPE problem, we show that a metric Turing transducer can be constructed to solve the problem. Let $\hat{\mathcal{T}}$ be a metric Turing transducer that on input $(\mathcal{B}, \mathbf{e}, k)$ performs the following computations: it traverses a topological sort of the network's nodes $\mathbf{V}$; in each step i, it non-deterministically chooses a value v_i for node V_i (for a node E_i from the set $\mathbf{E}$ of evidence nodes, the value conform $\mathbf{e}$ is chosen) and multiplies the corresponding (conditional) probabilities along its path. Each computation path thereby establishes a joint probability $\Pr(\mathbf{v}) = \prod_{V_i \in \mathbf{V}} \Pr(v_i \mid \pi(V_i))$ for a thus constructed joint value assignment $\mathbf{v}$ to $\mathbf{V}$. Note that $\Pr(\mathbf{v}) = \Pr(\mathbf{m}, \mathbf{e})$ for the constructed assignment $\mathbf{m}$ to the explanation variables $\mathbf{M}$. Further note that the computations involved take a time which is polynomial in the number of variables in the Kth MPE instance. The output of the transducer is, for each computation path, a binary representation of $1 - \Pr(\mathbf{m}, \mathbf{e})$ with sufficient (but polynomial) precision, combined with the logical inverse of the binary representation of the assignment $\mathbf{m}$ itself. $\text{KthValue}_{\hat{\mathcal{T}}}(\mathcal{B}, \mathbf{e}, k)$ now returns an encoding of the kth most probable explanation for $\mathbf{e}$. We conclude that Kth MPE is in $\mathsf{FP}^{\mathsf{PP}}$.

To prove hardness for the class $\mathsf{FP}^{\mathsf{PP}}$, we reduce the Kth Sat′ problem defined in Section 2.3 to Kth MPE. Let (ϕ, k) be an instance of Kth Sat′. From the Boolean formula ϕ we construct the probabilistic network $\mathcal{B}_\phi$ as described above; we further let $\mathbf{E} = \{V_\phi\}$ and let $\mathbf{e}$ be the value assignment $V_\phi = \mathit{true}$. The thus constructed instance of the Kth MPE problem is $(\mathcal{B}_\phi, V_\phi = \mathit{true}, k)$; note that the construction can be performed in polynomial time. For any joint value assignment $\mathbf{x}$ to the variable-instantiation part $\mathbf{X}$ of $\mathcal{B}_\phi$, we now have that $\Pr(\mathbf{X} = \mathbf{x} \,|\, V_\phi = \mathit{true}) = \frac{\Pr(\mathbf{X}=\mathbf{x}, V_\phi = \mathit{true})}{\Pr(V_\phi = \mathit{true})} = \alpha \cdot \Pr(\mathbf{X} = \mathbf{x}, V_\phi = \mathit{true})$, where $\alpha = \Pr(V_\phi = \mathit{true})^{-1}$ can be regarded a normalisation constant. For any assignment $\mathbf{x}$ to the variables $\mathbf{X}$ that satisfies ϕ, we further find that $\Pr(\mathbf{X} = \mathbf{x} \,|\, V_\phi = \mathit{true}) = \alpha \cdot \Pr(\mathbf{X} = \mathbf{x})$; for any non-satisfying assignment $\mathbf{x}$ on the other hand, we have that $\Pr(\mathbf{X} = \mathbf{x}, V_\phi = \mathit{true}) = 0$ and hence $\Pr(\mathbf{X} = \mathbf{x} \,|\, V_\phi = \mathit{true}) = 0$. In terms of their posterior probabilities given $V_\phi = \mathit{true}$, therefore, all satisfying joint value assignments are ordered before all non-satisfying ones. Since the values of the nodes from the truth-setting part $\mathbf{T}$ are fully determined by the values of their parents, we thus have that, given evidence $V_\phi = \mathit{true}$, the kth MPE corresponds to the lexicographically kth satisfying value assignment to the variables in ϕ, and vice versa. Given an algorithm for solving Kth MPE, we can thus solve KthSat′ as well, which proves $\mathsf{FP}^{\mathsf{PP}}$-hardness of Kth MPE. □

We now turn to the case where $k = 1$, that is, to the basic MPE problem, for which we show $\mathsf{FP}^{\mathsf{NP}}$-completeness by a similar construction as above.

Proposition 1. MPE *is* $\mathsf{FP}^{\mathsf{NP}}$*-complete.*

Proof. To prove membership of $\mathsf{FP}^{\mathsf{NP}}$, a metric Turing transducer as above is constructed. $\text{Opt}_{\hat{\mathcal{T}}}(\mathcal{B}, \mathbf{e})$ then returns the most probable explanation given the evidence $\mathbf{e}$. To prove hardness, we apply the same construction as above to reduce, in polynomial time, the LexSat problem to the MPE problem. □

Note that the functional variant of the MPE problem is in $\mathsf{FP}^{\mathsf{NP}}$, while its decision variant is in NP [6]. This relation between the decision and functional variants of a problem is quite commonly found in optimisation problems: if the solution of a functional problem variant has polynomially bounded length, then there exists a polynomial-time Turing reduction from the functional variant to the decision variant of that problem, and hence if the decision variant is in NP, then the functional variant of the problem is in $\mathsf{FP}^{\mathsf{NP}}$ [15].

4 Complexity of K-th Partial MAP

While the decision variant of the MPE problem is complete for the class NP, the decision variant of the Partial MAP problem is known to be $\mathsf{NP}^{\mathsf{PP}}$-complete [7]. In the previous section, we proved that the functional variant of the Kth MPE problem is $\mathsf{FP}^{\mathsf{PP}}$-complete. Intuitively, these results suggest that the Kth Partial MAP problem is complete for the complexity class $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$. To the best of our knowledge, no complete problems have been discussed in the literature for this class. We will now show that the Kth Partial MAP problem indeed is complete for the class $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$, by a reduction from the KthNumSat′ problem.

We address the construction of a probabilistic network $\mathcal{B}_\phi$ from the Boolean formula ϕ of an instance of the KthNumSat′ problem. We recall from Section 2.3 that we defined KthNumSat′ so as to forestall the need of dealing with outputs equal to $\perp$. While formally our reduction should be from KthNumSat′, we decided, for ease of exposition, to use an instance of the original KthNumSat problem for our running example and to assume that the Boolean formula involved has 'sufficiently many' satisfying truth assignments, that is, sufficiently many in terms of the constants k, l. For our running example, we use the Boolean formula $\phi_{\text{ex}} = ((X_1 \vee \neg X_2) \wedge X_3) \vee \neg X_4$, for which we want to find the lexicographically second assignment to the variables $\{X_1, X_2\}$ with which exactly three truth assignments to $\{X_3, X_4\}$ satisfy ϕ_{ex}, that is, $k = 2$, $l = 3$; the reader can verify that the instance has the solution $X_1 = \mathit{true}, X_2 = \mathit{false}$.

As before, we create a root node X_i for each Boolean variable X_i from the formula ϕ, this time with a uniform prior probability distribution. The nodes $X_1, \ldots, X_m$ with each other constitute the variable-instantiation part $\mathbf{X}$ of the network $\mathcal{B}_\phi$; the nodes from this part will be taken as the explanation nodes for the Kth Partial MAP instance under construction. For the logical operators from the formula ϕ, we add nodes to the probabilistic network as before, with V_ϕ being the node associated with the top-level operator. Note that for any joint value assignment $\mathbf{x}$ to the explanation nodes $\mathbf{X}$, we now have that $\Pr(V_\phi = \mathit{true} \mid \mathbf{x}) = \frac{s}{2^{n-m}}$, where s is the number of truth value assignments to the Boolean variables $\{X_{m+1}, \ldots, X_n\}$ that, jointly with $\mathbf{x}$, satisfy ϕ.

We further construct an enumeration part $\mathbf{N}$ for the network. For this purpose, we add nodes $Y_1, \ldots, Y_m$, with the possible values *true* and *false*, to the variable-instantiation part $\mathbf{X}$ such that for all $i = 1, \ldots, m$, node Y_i has node X_i for its unique parent; for each such node Y_i, we set $\Pr(Y_i = \mathit{true} \mid X_i = \mathit{true}) = \frac{1}{2^{i+n-m+1}}$ and $\Pr(Y_i = \mathit{true} \mid X_i = \mathit{false}) = 0$. We further add a binary-tree structure to the network, composed of nodes E_j mimicking the disjunction operator; the arcs of this tree structure are directed from the leaves $Y_1, \ldots, Y_m$ to the root node, which will be denoted by E_ϕ. For our running example, we thus add nodes Y_1 and Y_2 as successors to the nodes X_1 and X_2, respectively, with $\Pr(Y_1 = \mathit{true} \mid X_1 = \mathit{true}) = \frac{1}{16}$ and $\Pr(Y_2 = \mathit{true} \mid X_2 = \mathit{true}) = \frac{1}{32}$. In addition, a single

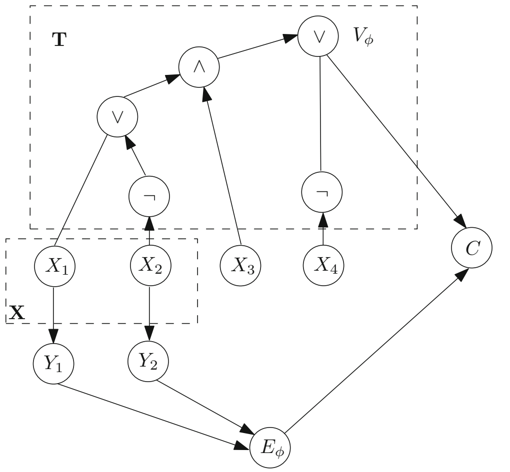

Fig. 2. The acyclic directed graph of the probabilistic network $\mathcal{B}_{\phi_{\text{ex}}}$ constructed from the KthNumSat instance with the Boolean formula $\phi_{\text{ex}} = ((X_1 \vee \neg X_2) \wedge X_3) \vee \neg X_4$ and the explanation nodes X_1, X_2

root node E_ϕ is added to the network under construction, with Y_1 and Y_2 for its parents. Note that for each joint value assignment $\mathbf{x}$ to the explanation nodes $X_1, \ldots, X_m$, we now have that $\Pr(E_\phi = \mathit{true} \mid \mathbf{x}) < \frac{1}{2^{n-m}}$; more specifically, for the jth lexicographically lowest ordered joint value assignment $\mathbf{x}$ to $X_1, \ldots, X_m$, we have $\Pr(E_\phi = \mathit{true} \mid \mathbf{x}) = \frac{j-1}{2^{n+1}}$.

To conclude the construction, we add to the network a node C with V_ϕ and E_ϕ for its parents, with the following conditional probability distributions:

$$\Pr(C = \mathit{true} \mid V_\phi, E_\phi) = \begin{cases} 1 & \text{if } V_\phi = \mathit{true}, E_\phi = \mathit{true} \\ \frac{1}{2} & \text{if } V_\phi = \mathit{true}, E_\phi = \mathit{false} \\ \frac{1}{2} & \text{if } V_\phi = \mathit{false}, E_\phi = \mathit{true} \\ 0 & \text{if } V_\phi = \mathit{false}, E_\phi = \mathit{false} \end{cases}$$

Note that since $\Pr(E_\phi = \mathit{true} \mid \mathbf{x}) < \frac{1}{2^{n-m}}$ for any joint value assignment $\mathbf{x}$ to the explanation nodes $X_1, \ldots, X_m$, these probabilities ensure that the posterior probability $\Pr(C = \mathit{true} \mid \mathbf{x})$ lies within the interval $[\frac{s}{2^{n-m+1}}, \frac{s+1}{2^{n-m+1}}]$, where s is the number of value assignments to the Boolean variables $\{X_{m+1}, \ldots, X_n\}$ that, jointly with $\mathbf{x}$, satisfy the formula ϕ. Figure 2 shows the graphical structure of the probabilistic network that is thus constructed from the example formula ϕ_{ex}.

Theorem 2. Bounded Kth Partial MAP *is* $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$*-complete.*

Proof. The membership proof for the Bounded Kth Partial MAP problem is quite similar to the membership proof for the Kth MPE problem from Theorem 1, that is, we construct a metric Turing transducer to solve the problem. Note that for the complexity class $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$ we are now allowed to consult a more powerful oracle than for the class $\mathsf{FP}^{\mathsf{PP}}$. We observe that for the Bounded Kth

Partial MAP problem, we actually *need* an oracle of higher power, since we need to solve the #P-complete problem of Exact Inference to compute the required joint probabilities: while for the Kth MPE problem we could efficiently compute probabilities for joint value assignments to all variables taking polynomial time, we must now compute probabilities of joint value assignments to a subset of the variables, which involves summing over all assignments to the intermediate variables involved. Now, if the probability $\Pr(\mathbf{m}, \mathbf{e})$ obtained for a joint value assignment $\mathbf{m}$ to the explanation variables $\mathbf{M}$ is within the interval $[a, b]$, the transducer outputs a binary representation of $1 - \Pr(\mathbf{m}, \mathbf{e})$ along with the logical inverse of the binary representation of $\mathbf{m}$; otherwise, it outputs $\perp$. Clearly, $\text{KthValue}_{\hat{\mathcal{T}}}$ returns an encoding of the kth most probable value assignment to the explanation variables in view of the evidence $\mathbf{e}$. We conclude that Bounded Kth Partial MAP is in $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$.

To prove hardness for the class $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$, we reduce the KthNumSat problem to Bounded Kth Partial MAP. Let (ϕ, k, l) be an instance of KthNumSat. From the Boolean formula ϕ we construct a probabilistic network $\mathcal{B}_\phi$ as described above; we further let $\mathbf{E} = \{C\}$ and let $\mathbf{e}$ be the value assignment $C = \mathit{true}$. The conditional probability distributions of the constructed network ensure that the posterior probability $\Pr(C = \mathit{true} \mid \mathbf{x})$ for a joint value assignment $\mathbf{x}$ to the nodes $\{X_1, \ldots, X_m\}$ with which exactly l truth value assignments to $\{X_{m+1}, \ldots, X_n\}$ satisfy ϕ, lies within the interval $[\frac{l}{2^{n-m+1}}, \frac{l+1}{2^{n-m+1}}]$. Moreover, if two assignments $\mathbf{x}$ and $\mathbf{x}'$ both are such that exactly l truth value assignments to $\{X_{m+1}, \ldots, X_n\}$ serve to satisfy ϕ, then $\Pr(C = \mathit{true} \mid \mathbf{x}) > \Pr(C = \mathit{true} \mid \mathbf{x}')$ if $\mathbf{x}$ is ordered before $\mathbf{x}'$ in descending lexicographic order. For the constructed instance of the Bounded Kth Partial MAP problem, we thus have that the kth most probable joint value assignment to the explanation nodes $X_1, \ldots, X_m$ corresponds with the lexicographically kth truth value assignment to the Boolean variables $\{X_1, \ldots, X_m\}$ with which exactly l assignments to $\{X_{m+1}, \ldots, X_n\}$ satisfy ϕ. Clearly, the above reduction is a polynomial-time one-Turing reduction from KthNumSat to Kth Partial MAP. Given an algorithm for solving Bounded Kth Partial MAP, we can thus solve the KthNumSat problem as well, which proves $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-hardness of Bounded Kth Partial MAP. □

$\mathsf{FP}^{\mathsf{NP}^{\mathsf{PP}}}$-completeness of Bounded Partial MAP, which is the special case of Bounded Kth Partial MAP with $k = 1$, now follows by a very similar proof.

Proposition 2. Bounded Partial MAP *is* $\mathsf{FP}^{\mathsf{NP}^{\mathsf{PP}}}$*-complete.*

5 Conclusion

We addressed the computational complexity of two problems that arise in practical applications of probabilistic networks. Informally spoken, these problems ask for the kth most likely explanation for a given collection of observations in a network. For the Kth MPE problem, an explanation is defined as a joint value assignment to all non-observed variables; we showed that this problem is $\mathsf{FP}^{\mathsf{PP}}$-complete. For the Kth Partial MAP problem, a designated subset of non-observed variables is distinguished; these variables are taken as the explanation variables for which a joint value assignment is being sought. We showed that this particular problem is $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-complete. In the future, we would like to further study the complexity of the two problems when the constant k is bounded by a polynomial function in the number of variables involved.

By our results we pinpointed the precise complexity of two practical problems, although it is fair to mention that from a practitioners' point of view knowing NP-hardness would have sufficed. Interesting from a theoretical point of view, however, is the observation that our complexity results are among the very few showing practically relevant problems to be complete for complexity classes that are as special as $\mathsf{FP}^{\mathsf{PP}}$and $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$.

References

1. Charitos, T., van der Gaag, L.C., Visscher, S., Schurink, C.A.M., Lucas, P.J.F.: A dynamic Bayesian network for diagnosing ventilator-associated pneumonia in ICU patients. Expert Systems with Applications 26, 1249–1258 (2009)
2. Geenen, P.L., Elbers, A.R.W., van der Gaag, L.C., Loeffen, W.L.A.: Development of a probabilistic network for clinical detection of classical swine fever. In: Proceedings of the 11th Symposium of the International Society for Veterinary Epidemiology and Economics, Cairns, Australia, pp. 667–669 (2006)
3. Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, Palo Alto (1988)
4. Roth, D.: On the hardness of approximate reasoning. Artificial Intelligence 82, 273–302 (1996)
5. Shimony, S.E.: Finding MAPs for belief networks is NP-hard. Artificial Intelligence 68, 399–410 (1994)
6. Bodlaender, H.L., van den Eijkhof, F., van der Gaag, L.C.: On the complexity of the MPA problem in probabilistic networks. In: van Harmelen, F. (ed.) Proceedings of the 15th European Conference on Artificial Intelligence, pp. 675–679. IOS Press, Amsterdam (2002)
7. Park, J.D., Darwiche, A.: Complexity results and approximation settings for MAP explanations. Journal of Artificial Intelligence Research 21, 101–133 (2004)
8. Charniak, E., Shimony, S.E.: Cost-based abduction and MAP explanation. Acta Informatica 66, 345–374 (1994)
9. Abdelbar, A., Hedetniemi, S.: Approximating MAPs for belief networks is NP-hard and other theorems. Artificial Intelligence 102, 21–38 (1998)
10. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs. Springer, New York (2007)
11. Papadimitriou, C.H.: On the complexity of unique solutions. Journal of the ACM 31, 392–400 (1984)
12. Krentel, M.W.: The complexity of optimization problems. Journal of Computer and System Sciences 36, 490–509 (1988)
13. Toda, S.: Simple characterizations of P(#P) and complete problems. Journal of Computer and System Sciences 49, 1–17 (1994)
14. Kwisthout, J.H.P.: The Computational Complexity of Probabilistic Networks. PhD thesis, Universiteit Utrecht (2009)
15. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley, Reading (1994)