# The Complexity of Finding $k$th 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 $k$th 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 $k$th 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 model of a joint probability distribution over a set of stochastic variables [3]. It consists of a directed acyclic graph, encoding the variables and their probabilistic interdependences, and an associated set of conditional probabilities. Various algorithms have been designed for probabilistic inference, that is, for computing probabilities of interest from a network. These algorithms typically exploit structural properties of the network's graph to decompose the computations involved. Probabilistic inference is known to be $\mathsf{PP}$-complete in general. Many other problems to be solved in practical applications of probabilistic networks are also known to have an 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 dependences 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 [4, 5]; for the PARTIAL MAP problem $\mathsf{NP}^{\mathsf{PP}}$-completeness was established [6].

In many applications, one 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 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 $k$th 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 $k$th most probable explanation problem with the best explanation as additional input [7], the KTH MPE problem without this extra information is NP-hard in general [8]. 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 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 $k$th 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 from a theoretical viewpoint.

The paper is organised as follows. In Section 2, our notational conventions as well as the definitions used in the paper are introduced. We discuss the computational complexity of finding $k$th 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 the definitions used in this paper. 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 various complexity classes and state some 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 $\mathbf{X}$ are uniquely ordered. If $\Pr(\mathbf{x}_i) = \Pr(\mathbf{x}_j)$ for two joint value assignments $\mathbf{x}_i$ and $\mathbf{x}_j$, they are ordered lexicographically by their respective binary representation, taking the value for $X_1$ to be the most significant element. In general, $\mathbf{x}_i \prec \mathbf{x}_j$ if and only if $\text{bin}(\Pr(\mathbf{x}_i), \mathbf{x}_i) \prec \text{bin}(\Pr(\mathbf{x}_j), \mathbf{x}_j)$.

A probabilistic network now is a tuple $\mathcal{B} = (\mathbf{G}, \Gamma)$ where $\mathbf{G} = (\mathbf{V}, A)$ is a directed acyclic graph and $\Gamma$ 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 $V_1, \ldots, V_n$, $n \geq 1$, 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 $\Gamma$ 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 $\mathcal{B} = (\mathbf{G}, \Gamma)$ uniquely defines a joint probability distribution $\Pr(\mathbf{V}) = \prod_{V_i \in \mathbf{V}} \Pr(V_i \mid \pi(V_i))$ that respects the independences portrayed by its digraph. Since it defines a unique distribution, a probabilistic network allows the computation of any probability of interest over its variables [9].

### 2.2 The $k$th Most Probable Explanation Problems

The main problem studied in this paper is the problem of finding a $k$th 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 natural number $k$.
**Output:** A $k$th 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 $\Pr(\mathbf{m} | \mathbf{e})$ 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 natural number $k$; and rational numbers $a, b$ with $0 \leq a \leq b \leq 1$.
**Output:** A tuple $(\mathbf{m}_k, p_k)$, where $\mathbf{m}_k$ is a $k$th most probable assignment to $\mathbf{M}$ given $\mathbf{e}$ from among all 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 [10–12].

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 $k$th 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 $\mathsf{P}^{\mathsf{NP}}$or $\mathsf{P}^{\mathsf{PP}}$[11]. 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. $\mathrm{Out}_{\hat{\mathcal{M}}}(x)$ denotes the set of outputs of $\hat{\mathcal{M}}$ on input $x$; $\mathrm{Opt}_{\hat{\mathcal{M}}}(x)$ is the smallest number in $\mathrm{Out}_{\hat{\mathcal{M}}}(x)$, and $\mathrm{KthValue}_{\hat{\mathcal{M}}}(x,k)$ is the $k$-th smallest number in $\mathrm{Out}_{\hat{\mathcal{M}}}(x)$. 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 $x$ [12]. 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 $f \leq^{\mathsf{FP}}_{\text{1-T}} \mathrm{Opt}_{\hat{\mathcal{T}}}$. Correspondingly, a set $L$ is in $\mathsf{P}^{\mathsf{NP}}$ if and only if a metric Turing Machine $\hat{\mathcal{M}}$ can be constructed, such that $\mathrm{Opt}_{\hat{\mathcal{M}}}(x)$ is odd if and only if $x \in L$. Similar observations hold for $\mathsf{FP}^{\mathsf{PP}}$ and $\mathsf{P}^{\mathsf{PP}}$, and the $\mathrm{KthValue}_{\hat{\mathcal{M}}}$ and $\mathrm{KthValue}_{\hat{\mathcal{T}}}$ functions [11, 12]. $\mathsf{FP}^{\mathsf{NP}}$- 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 now introduce some functional variants of the satisfiability problem which we will use in the sequel, and state their completeness results.

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

The LexSat problem is the special case of the Kth SAT problem with $k = 1$. Kth SAT and LexSat are complete for $\mathsf{FP}^{\mathsf{NP}}$ and $\mathsf{FP}^{\mathsf{PP}}$, respectively [11, 12].

KthNumSat
**Instance:** A Boolean formula $\phi(X_1, \ldots, X_m, \ldots, X_n), m \leq n, n \geq 1$; natural numbers $k, l$.
**Output:** The lexicographically $k$th 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 $\phi$; 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 [13].

## 3 Complexity of Kth MPE

We study the complexity of the Kth MPE problem as 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 from $\mathsf{FP}^{\mathsf{PP}}$.

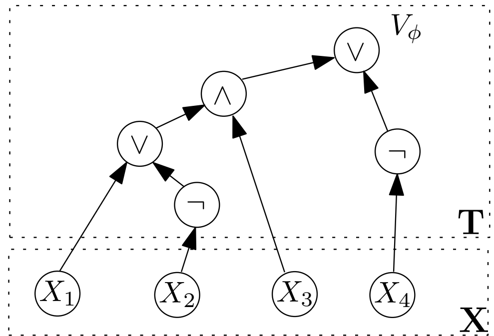


**Fig. 1.** The acyclic directed graph of the probabilistic network $\mathcal{B}_{\phi_{\text{ex}}}$ constructed from the Boolean formula $\phi_{\text{ex}} = ((X_1 \vee \neg X_2) \wedge X_3) \vee \neg X_4$

---

We begin by describing the construction of a probabilistic network $\mathcal{B}_\phi$ from the Boolean formula $\phi$ of an instance of the KTH SAT problem; upon doing so, we use the formula $\phi_{\text{ex}} = ((X_1 \vee \neg X_2) \wedge X_3) \vee \neg X_4$ for our running example. For each Boolean variable $X_i$ in $\phi$, we include a root node $X_i$ in the network $\mathcal{B}_\phi$, 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 lexicographically ordered before $\mathbf{x}'$. More specifically, we set $p_i = \frac{1}{2} - \frac{2^i - 1}{2^{n+1}}$. In our running example with four Boolean variables, the prior probabilities for the nodes $X_1, \ldots, X_4$ thus are set to $p_1 = \frac{15}{32}$, $p_2 = \frac{13}{32}$, $p_3 = \frac{9}{32}$, and $p_4 = \frac{1}{32}$. Note that we have that $p_i \cdot \overline{p_{i+1}} \cdot \ldots \cdot \overline{p_n} > \overline{p_i} \cdot p_{i+1} \cdot \ldots \cdot p_n$ for every $i$. Since the root nodes $X_i$ are mutually independent in the network under construction, therefore, the ordering property stated above is attained. Further note that the associated prior 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 $\phi$, 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 $\phi$ will be denoted by $V_\phi$. The operator nodes with each other constitute the truth-setting part $\mathbf{T}$ of the network. The probabilistic network $\mathcal{B}_{\phi_{\text{ex}}}$ that is constructed from the example formula $\phi_{\text{ex}}$ 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 $\phi$.

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

*Proof.* To prove membership, 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. 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 an 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 a binary representation of the assignment $\mathbf{m}$ itself. $\text{KthValue}_{\hat{\mathcal{T}}}(\mathcal{B}, \mathbf{e}, k)$ now returns an encoding of the $k$th most probable explanation for $\mathbf{e}$. We conclude that Kth MPE is in $\mathsf{FP}^{\mathsf{PP}}$.

To prove hardness, we reduce the Kth SAT problem to the Kth MPE problem. Let $(\phi, k)$ be an instance of Kth SAT. From $\phi$ we construct the 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 $\Pr(\mathbf{X} = \mathbf{x} \mid V_\phi = \mathit{true}) = \frac{\Pr(\mathbf{X}=\mathbf{x}, V_\phi=\mathit{true})}{\Pr(V_\phi=\mathit{true})} = \alpha \cdot \Pr(\mathbf{X} = \mathbf{x}, V_\phi = \mathit{true})$ for a normalisation constant $\alpha$, since the prior probability $\Pr(V_\phi = \mathit{true})$ can be regarded a constant. For any satisfying assignment $\mathbf{x}$ to the variables $\mathbf{X}$, we have that $\Pr(\mathbf{X} = \mathbf{x} \mid V_\phi = \mathit{true}) = \alpha \cdot \Pr(\mathbf{X} = \mathbf{x})$; for any non-satisfying assignment $\mathbf{x}$ on the other hand, we find that $\Pr(\mathbf{X} = \mathbf{x}, V_\phi = \mathit{true}) = 0$ and hence that $\Pr(\mathbf{X} = \mathbf{x} \mid V_\phi = \mathit{true}) = 0$. All satisfying joint value assignments thus are ordered before all non-satisfying ones. We now observe that the set $\mathbf{M}$ of explanation nodes contains both the variable-instantiation nodes $\mathbf{X}$ and the truth-setting nodes $\mathbf{T}$. Since the values of the nodes from $\mathbf{T}$ are fully determined by the values of their parents, we have that, given evidence $V_\phi = \mathit{true}$, the $k$th MPE corresponds to the lexicographically $k$th satisfying value assignment to the variables in $\phi$, 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 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, 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 [5]. 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 $\mathsf{FP}^{\mathsf{NP}}$ [14].

## 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 [6]. 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 first describe the construction of a probabilistic network $\mathcal{B}_\phi$ from an instance $(\phi(X_1, \ldots, X_m, \ldots, X_n), k, l)$, $m \leq n$, $n \geq 1$, of the KthNumSat problem. For our running example, we again use the Boolean formula $\phi_{\text{ex}} = ((X_1 \vee \neg X_2) \wedge X_3) \vee \neg X_4$, for which we now 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 $\phi_{\text{ex}}$, that is, $k = 2$ and $l = 3$; 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 $\phi$, 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 the MAP nodes for the Kth Partial MAP instance under construction.

For the logical operators from the formula $\phi$, we create additional nodes in the network as before, with $V_\phi$ for the node associated with the top-level operator. For any joint value assignment $\mathbf{x}$ to the instantiation 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 $\phi$.

We now further construct an enumeration part $\mathbf{N}$ for the network. To the variable-instantiation part $\mathbf{X}$, we add nodes $Y_1, \ldots, Y_m$, with values *true* and *false*, with $X_i$ the unique parent of $Y_i$ $(1 \leq i \leq m)$. We take, for $1 \leq i \leq m$:

$$\Pr(Y_i = \mathit{true} \mid X_i = \mathit{false}) = \frac{1}{2^{i+n-m+1}}.$$

and

$$\Pr(Y_i = \mathit{true} \mid X_i = \mathit{true}) = 0.$$

To this, we add a binary tree with $\vee$-nodes (i.e., nodes with a probability table such that the node attains value *true* if and only if at least one parent has value *true*. The leaves of this tree are exactly the nodes $Y_1, \ldots, Y_m$; we call the root of the tree $E_\phi$.

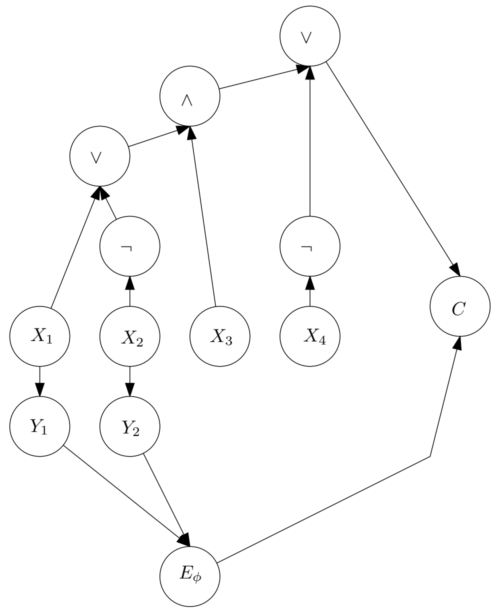


**Fig. 2.** The acyclic directed graph of the probabilistic network $\mathcal{B}_{\phi_{\text{ex}}}$ constructed from the KthNumSat instance with the Boolean formula $\phi_{\text{ex}} = ((X_1 \vee \neg X_2) \wedge X_3) \vee \neg X_4$ and the MAP nodes $X_1, X_2$. Note that $E_\phi$ is an $\vee$-node.

---

For example, when $m = 4$, we have three $\vee$-nodes, one with parents $Y_1$ and $Y_2$, one with parents $Y_3$ and $Y_4$, and one with parents the two other new $\vee$-nodes; the latter is $E_\phi$. In our running example, $E_\phi$ has $Y_1$ and $Y_2$ as parents. We have that $\Pr(Y_1 = \mathit{true} \,|\, X_1 = \mathit{false}) = \frac{1}{2^4}$, and $\Pr(Y_2 = \mathit{true} \,|\, X_2 = \mathit{false}) = \frac{1}{2^5}$.

Note that for each joint value assignment $\mathbf{m}$ to the nodes $X_1, \ldots, X_m$, we have that $\Pr(E_\phi = \mathit{true} \mid \mathbf{m}) < \frac{1}{2^{n-m}}$. More precisely, using the third law of probability theory, one can easily show that if $\mathbf{m}$ is the $j$th lexicographically largest joint value assignment to $X_1, \ldots, X_m$, then $\Pr(E_\phi = \mathit{true} \,|\, \mathbf{m}) = \frac{j-1}{2^{n+1}}$. In particular, this shows that if a joint value assignment $\mathbf{m}$ to the MAP nodes is lexicographically ordered before $\mathbf{m}'$, then $\Pr(E_\phi = \mathit{true} \,|\, \mathbf{m}') > \Pr(E_\phi = \mathit{true} \,|\, \mathbf{m})$.

To conclude the construction, we add to the network an additional node $C$ with $V_\phi$ and $E_\phi$ for its parents, with the following conditional probability table:

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

Since $\Pr(E_\phi = \mathit{true} \mid \mathbf{x}) < \frac{1}{2^{n-m}}$ for any joint value assignment $\mathbf{x}$ to the MAP nodes, this table ensures that the 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 $\phi$.

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

*Proof.* The membership proof 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 MAP variables $\mathbf{M}$ is within the interval $[a, b]$, the transducer outputs binary representations of $\mathbf{m}$ and $1 - \Pr(\mathbf{m}, \mathbf{e})$; otherwise, it outputs $\perp$. Clearly, $\text{KthValue}_{\hat{\mathcal{T}}}$ returns an encoding of the $k$th most probable value assignment to the MAP 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, we construct a probabilistic network $\mathcal{B}_\phi$ from a given instance $\phi(X_1, \ldots, X_m, \ldots, X_n)$, as described above. The conditional probabilities in the constructed network ensure that the probability of a value assignment $\mathbf{x}$ to the nodes $\{X_1, \ldots, X_m\}$ such that $l$ truth value assignments to the variables $\{X_{m+1}, \ldots, X_n\}$ satisfy $\phi$, is in the interval $[\frac{l}{2^{n-m+1}}, \frac{l+1}{2^{n-m+1}}]$. Moreover, if both $\mathbf{x}$ and $\mathbf{x}'$ are such that $l$ truth value assignments to the variables $\{X_{m+1}, \ldots, X_n\}$ satisfy $\phi$, then $\Pr(C = \mathit{true} \mid \mathbf{x}) > \Pr(C = \mathit{true} \mid \mathbf{x}')$ if the truth value that corresponds with $\mathbf{x}$ is lexicographically ordered before $\mathbf{x}'$. Thus, with evidence $C = \mathit{true}$ and ranges $[\frac{s}{2^{n-m+1}}, \frac{s+1}{2^{n-m+1}}]$, the $k$th Partial MAP corresponds to the lexicographical $k$th truth assignment to the variables $X_1, \ldots, X_m$ for which exactly $s$ truth assignments to $X_{m+1}, \ldots, X_n$ satisfy $\phi$. Clearly, the above reduction is a polynomial-time one-Turing reduction from KthNumSat to Kth Partial MAP. This proves $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-hardness of Bounded Kth Partial MAP. □

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

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

## 5 Conclusion

In this paper, we addressed the computational complexity of two problems that are relevant for the practical use of probabilistic networks. Intuitively, the problems state, for a given probabilistic network, what is the $k$th most likely explanation for a given set of observations, called the evidence. The first variant, which is $\mathsf{FP}^{\mathsf{PP}}$-complete, has as evidence all variables that are not given as explanation, while the second variant, being $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-complete has intermediate variables; these neither are used as evidence or as explanation. The contribution of our work is twofold: first, we pinpoint precisely the complexity of these relevant problems, although, from a practitioners point of view, knowing that they are NP-hard is sufficient. Secondly, our results form one of the very few results of the type where we show problems relevant from a practical context to be complete for complexity classes that are as special as $\mathsf{FP}^{\mathsf{PP}}$and $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$.

**Acknowledgement.** This research was done in the *Algorithmic Complexity of Probabilistic Networks* project supported by the Netherlands Organisation for Scientific Research (NWO).

## References

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

13. J.H.P. Kwisthout (2009). *The Computational Complexity of Probabilistic Networks*. PhD thesis, Universiteit Utrecht.
14. C.H. Papadimitriou (1994). *Computational Complexity*. Addison-Wesley, Reading.
15. D. Roth (1996). On the hardness of approximate reasoning. *Artificial Intelligence*, vol. 82, pp. 273 – 302.