Complexity Results for Enumerating MPE and Partial MAP

Johan Kwisthout
Department of Information and Computer Sciences, Utrecht University
P.O. Box 80.089, 3508TB Utrecht, The Netherlands
johank@cs.uu.nl

Abstract

While the computational complexity of finding the most likely joint value assignment given full (MPE) or partial (Partial MAP) evidence is known, less attention has been given to the related problem of finding the k-th most likely assignment, for arbitrary values of k. Yet this problem has very relevant practical usages, for example when we are interested in a list of alternative explanations in decreasing likeliness. In this paper a hardness proof of enumerating Most Probable Explanations (MPEs) and Maximum A-Priori Probabilities (Partial MAPs) is given. We prove that finding the k-th MPE is P^{PP}-complete, and prove that finding the k-th Partial MAP is $\mathsf{P}^{\mathsf{PP}^{\mathsf{PP}}}$-complete.

1 Introduction

An important problem that rises from the practical usage of probabilistic networks (Jensen, 2007; Pearl, 1988) is the problem of finding the most likely value assignment to a set of variables, given full or partial evidence. When the evidence is equal to the entire complement of that set in the network, the problem is known as the Most Probable Explanation or MPE-problem[1]. Finding, or even approximating, such a value assignment is NP-hard (Shimony, 1994; Bodlaender et al., 2002; Abdelbar and Hedetniemi, 1998). On the other hand, finding the most likely value assignment, given evidence for a *subset* of the complement set (the Partial MAP-problem), is even harder: Park and Darwich proved (2004) that this problem is $\mathsf{NP}^{\mathsf{PP}}$-complete and remains NP-complete on polytrees.

In practical applications, one often wants to find a number of different value assignments with a high likeliness, rather than only the most likely assignment (see e.g. Santos Jr. (1991) or Charniak and Shimony (1994)). For example, in medical applications one wants to suggest alternative (but also likely) explanations to a set of observations. One might like to prescribe medication that covers a number of plausible causes, rather than only the most probable cause. It may be useful to examine the second-best explanation to gain insight in *how good* the best explanation is, relative to other solutions, or, how sensitive it is to changes in the parameters of the network (Chan and Darwiche, 2006).

While algorithms exist that can sometimes find k-th best explanations fast, once the best explanation is known (Charniak and Shimony, 1994), it has been shown that calculating or even approximating the k-th best explanation is NP-hard (Abdelbar and Hedetniemi, 1998), whether the best explanation is known or not. Nevertheless, the exact complexity of this problem has not been established yet.

The complexity of finding k-th best assignments to the Partial MAP-problem has, to our best knowledge, not yet been investigated. However, in many applications it is unlikely that full evidence of the complement of the variables of interest in the network is available. For example, in the *Oesophagus Network*, a probabilistic network for patient-specific therapy selection for oesophageal cancer (van der Gaag et al., 2002), a number of variables (like the

[1] In the literature also denoted as Maximum Probability Assignment (MPA) or Maximum A-posteriori Probability (MAP).

presence of haematogenous metastases or the extent of lymph node metastases) are intermediate, non-observable variables. Likewise, the ALARM network (Beinlich et al., 1989) has sixteen observable and thirteen intermediate variables. Therefore, the problem of finding k-th best assignments, given *partial* evidence, may be even more relevant in practical applications than the corresponding problem where full evidence is available.

In this paper, we extend the problem of finding the most likely value assignment to the problem of enumerating joint value assignments, i.e., finding the k-th likely assignment for arbitrary values of k, with either full or partial evidence. We will prove that (decision variants of) these problems are complete for the complexity classes P^{PP} and $\mathsf{P}^{\mathsf{PP}^{\mathsf{PP}}}$, respectively, suggesting that these problems are much harder than the (already intractable) restricted cases where $k = 1$, and also much harder than the PP-complete Inference problem. Furthermore, while some problems are known to be P^{PP}-complete, finding the k-th Partial MAP is (to our best knowledge) the first problem with a practical application that is shown to be $\mathsf{P}^{\mathsf{PP}^{\mathsf{PP}}}$-complete, making this problem interesting from a more theoretical viewpoint as well.

This paper is organized as follows. First, in Section 2, we will briefly introduce probabilistic networks and introduce a number of concepts from computational complexity theory. We will discuss the complexity of enumerating value assignment with full, respectively partial, evidence in Sections 3 and 4. In Section 5 we conclude this paper.

2 Preliminaries

A probabilistic network $\mathbf{B} = (\mathbf{G}, \Gamma)$ is defined by a directed acyclic graph $\mathbf{G} = (\mathbf{V}, \mathbf{A})$, where $\mathbf{V} = \{V_1, \ldots, V_n\}$ models a set of stochastic variables and $\mathbf{A}$ models the (in)dependences between them, and a set of parameter probabilities Γ, 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. We will use bold upper case letters to denote sets of variables (i.e., subsets of $\mathbf{V}$) and bold lower case letters to denote particular value assignments to these sets. The set of observed variables (the *evidence* variables) will be denoted as $\mathbf{E}$, and the observations themselves as $\mathbf{e}$. We will use $\Pr(\mathbf{v} \,|\, \mathbf{e})$ as a shorthand for $\Pr(\mathbf{V} = \mathbf{v} \,|\, \mathbf{E} = \mathbf{e})$.

The MPE-problem is the problem of finding a joint value assignment $\mathbf{v}$ to $\mathbf{V} \setminus \mathbf{E}$ such that $\Pr(\mathbf{v} \,|\, \mathbf{e})$ is maximal. The Partial MAP-problem is the problem of finding a joint value assignment $\mathbf{v}$ to the so-called MAP-variables $\mathbf{V_{MAP}} \subsetneq \mathbf{V} \setminus \mathbf{E}$ such that $\Pr(\mathbf{v} \mid \mathbf{e})$ is maximal.

2.1 Complexity Theory

In the remainder, we assume that the reader is familiar with basic concepts of computational complexity theory, such as Turing Machines, the complexity classes P, NP, PP, $\#\mathsf{P}$, and completeness proofs for these classes. For a thorough introduction to these subjects we refer to textbooks like Garey and Johnson (1979) and Papadimitriou (1994). Furthermore, we use the concept of *oracle access*. A Turing Machine $\mathcal{M}$ has oracle access to languages in the class A, denoted as $\mathcal{M}^{\mathsf{A}}$, if it can query the oracle in one state transition, i.e., in $O(1)$. We can regard the oracle as a 'black box' that can answer membership queries in constant time. For example, $\mathsf{NP}^{\mathsf{PP}}$ is defined as the class of languages which are decidable in polynomial time on a non-deterministic Turing Machine with access to an oracle deciding problems in PP, like the well known Inference-problem, which is PP-complete (Littman et al., 1998).

We will frequently use the fact that $\#\mathsf{P}$ is polynomial-time Turing equivalent to PP (Simon, 1977). Informally, this implies that a class that uses $\#\mathsf{P}$ as an oracle, can also be defined as using PP and vice versa. For example, the class $\mathsf{NP}^{\mathsf{PP}}$ is equal to the class $\mathsf{NP}^{\#\mathsf{P}}$; however, the former notation is more common. We will use this property frequently in our hardness proofs.

The complexity class P^{PP} is defined as the class of languages, decidable by a deterministic Turing Machine with access to a PP oracle. While P^{PP} is less known than the related classes $\mathsf{NP}^{\mathsf{PP}}$ and $\mathsf{co-NP}^{\mathsf{PP}}$, complete decision problems have been discussed in Toda (1994). Intuitively, while NP is associated with the *existence* of a satisfying solution, PP with a *threshold* of satisfying solutions, and $\#\mathsf{P}$ with the *exact number* of satisfying solutions, P^{PP} is associated with the *middle* satisfying solution. For this class, the canonical complete problems Mid SAT and Kth SAT are the problems of determining whether in the lexicographically middle (k-th) satisfying assignment $x_1x_2\ldots x_n \in \{0,1\}^n$ to a Boolean formula ϕ, the least significant bit is odd (Toda, 1994).

The complexity results in this paper are based on *function*—rather than *decision*—problems. While a decision problem requires a *yes* or *no* answer (like 'Is there a satisfying truth assignment to the variables in a formula?'), a function problem requires a construct, like a satisfying truth assignment. Formally, traditional complexity classes like P and NP are defined on decision problems, using *acceptor* Turing Machines. The functional counterparts of these classes, like FP and FNP are defined using *transducer* Turing Machines; on an input x a transducer $\mathcal{M}$ *computes* y if $\mathcal{M}$ halts in an accepting state with y on its output tape. In our opinion, the problem of finding the k-th solution has a more 'natural' correspondence with function problems than decision problems and require less technical details in our hardness proofs.

To prove P^{PP} (or $\mathsf{FP}^{\mathsf{PP}}$) -hardness of a particular problem, one needs to reduce it from a known complete problem like Kth SAT. To prove *membership* of P^{PP} ($\mathsf{FP}^{\mathsf{PP}}$), one needs to show that it is accepted (computed) by a *metric Turing Machine.* Metric Turing Machines were defined by Krentel (1988).

Definition 1 (Metric Turing Machine). A metric Turing Machine (metric TM for short) is a polynomial-time bounded non-deterministic Turing Machine such that every computation path halts with a binary number on an output tape. Let $\hat{\mathcal{M}}$ denote a metric TM, then $\text{Out}_{\hat{\mathcal{M}}}(x)$ denotes the set of outputs of $\hat{\mathcal{M}}$ on an input x, and $\text{KthValue}_{\hat{\mathcal{M}}}(x,k)$ is defined to be the k-th smallest number in $\text{Out}_{\hat{\mathcal{M}}}(x)$.

Toda showed (1994), that a function f is in $\mathsf{FP}^{\mathsf{PP}}$ if and only if there exists a metric TM $\hat{\mathcal{M}}$ such that f is polynomial-time one-Turing reducible[2] to $\text{KthValue}_{\hat{\mathcal{M}}}$ ($f \leq^{\mathsf{FP}}_{1-T} \text{KthValue}_{\hat{\mathcal{M}}}$ for short). Correspondingly, a set L is in P^{PP} if and only if a metric TM $\hat{\mathcal{M}}$ can be constructed, such that $\text{KthValue}_{\hat{\mathcal{M}}}$ is odd for an input x if and only if $x \in L$. In the remainder, we will construct such metric TMs for the MPE- and Partial MAP-problems to prove membership in $\mathsf{FP}^{\mathsf{PP}}$ and $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$.

3 Enumerating MPE

In this section we will construct a $\mathsf{FP}^{\mathsf{PP}}$-completeness proof for the Kth MPE problem. More specifically, we show that Kth MPE can be computed by a metric TM in polynomial time (proving membership of $\mathsf{FP}^{\mathsf{PP}}$), and we prove hardness of the problem by a reduction from Kth SAT. We formally define the functional[3] version of Kth MPE problem as follows.

Kth MPE
Instance: Probabilistic network $\mathbf{B} = (\mathbf{G}, \Gamma)$, evidence variables $\mathbf{E}$ with instantiation $\mathbf{e}$, natural number k.
Question: What is the k-th most probable assignment $\mathbf{v_k}$ to the variables in $\mathbf{V} \setminus \mathbf{E}$ given evidence $\mathbf{e}$?

The functional version of Kth SAT, the problem that we will use in the reduction, is defined as follows.

Kth SAT
Instance: Boolean formula $\phi(x_1, \ldots, x_n)$, natural number k.
Question: What is the lexicographically k-th assignment $x_1 \ldots\ x_n \in \{0,1\}^n$ that satisfies ϕ?

We will use the formula $\phi_{ex} = ((x_1 \vee \neg x_2) \wedge x_3) \vee \neg x_4$ as a running example. We construct a

[2] A function f is *polynomial-time one-Turing reducible* to a function g if there exist polynomial-time computable functions T_1 and T_2 such that for every x, $f(x) = T_1(x, g(T_2(x)))$ (Toda, 1994, p.5).

[3] Note that we can transform this functional version into a decision variant by designating a variable $V_d \in \mathbf{V} \setminus \mathbf{E}$ with v_d as one of its values, and asking whether $V_d = v_d$ in $\mathbf{v_k}$.

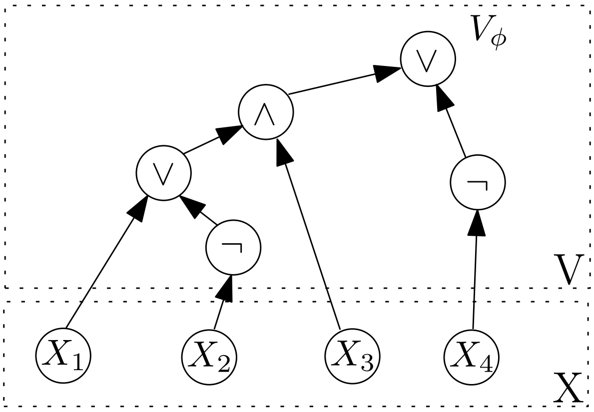

Figure 1: Example of k-th MPE construction for the formula $\phi_{ex} = ((x_1 \vee \neg x_2) \wedge x_3) \vee \neg x_4$

probabilistic network $\mathbf{B}_\phi$ from a given Boolean formula ϕ in the KTH SAT-instance with n variables x_i, as illustrated in Figure 1. For all variables x_i in the formula ϕ, we create a matching stochastic variable X_i in $\mathbf{V}$ for the network $\mathbf{B}_\phi$, with possible values *true* (T) and *false* (F). These variables are roots in the network $\mathbf{B}_\phi$ and are denoted as the *variable instantiation* part (X) of the network. The prior probabilities $p_1, \ldots, p_n$ for the variables $X_1, \ldots, X_n$ are chosen such that the prior probability of a particular value assignment $\mathbf{x}$ is higher than $\mathbf{x}'$, if and only if the corresponding truth assignment to $X_1, \ldots, X_n$ is lexicographically higher. More in particular, we choose prior probabilities $p_1, \ldots, p_i, \ldots, p_n$ such that $p_i = \frac{1}{2} - \frac{2^i - 1}{2^{n+1}}$. In our example with four variables, the probability distribution will be $p_1 = \frac{15}{32}$, $p_2 = \frac{13}{32}$, $p_3 = \frac{9}{32}$, and $p_4 = \frac{1}{32}$; the reader can verify that the probability of a value assignment $\mathbf{x}$ is higher than an assignment $\mathbf{x}'$, if and only if the corresponding truth assignment $x_1 \ldots\ x_n \in \{0,1\}^n$ is lexicographically smaller. Note that we can formulate these probabilities, using a number of bits which is polynomial in the input size.

For each logical operator in ϕ, we create an additional stochastic variable in the network, whose parents are the corresponding subformulas (or single sub-formula in case of a negation operator) and whose conditional probability table is equal to the truth table of that operator. For example, the variable corresponding to a $\wedge$-operator would have a conditional probability $\Pr(\wedge = T) = 1$ if and only if both its parents have the value *true*, and 0 otherwise. We denote the stochastical variable that is associated with the top-level operator in ϕ with V_ϕ. The thus constructed part of the network will be denoted as the *truth-setting* part (V) of the network. It is easy to see that, for a particular value assignment of variables X_i in the network, $\Pr(V_\phi = T) = 1$ if and only if the corresponding truth setting to the variables in ϕ satisfies ϕ.

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

Proof. To prove membership, we will show that a metric TM can be constructed for the KTH MPE-problem. Let $\hat{\mathcal{M}}$ be a metric non-deterministic TM that, on input $\mathbf{B}$, calculates $\Pr(\mathbf{V})$. Since $\Pr(\mathbf{V}) = \prod_{i=1}^{n} \Pr(v_i \mid \pi(V_i))$, $\hat{\mathcal{M}}$ calculates $\Pr(\mathbf{V} \mid \mathbf{e})$ by non-deterministically choosing instantiations v_i, consistent with evidence $\mathbf{e}$, at each step i, and multiplying the corresponding probabilities. The output is, for each computation path, a binary representation (e.g., in fixed precision notation) of $1 - \Pr(\mathbf{v} \mid \mathbf{e})$ with sufficient precision. Then, clearly $\text{KthValue}_{\hat{\mathcal{M}}}$ returns the k-th probable explanation of $\Pr(\mathbf{V} \mid \mathbf{e})$. This proves that KTH MPE is in $\mathsf{FP}^{\mathsf{PP}}$.

To prove hardness, we reduce KTH SAT to KTH MPE. Let ϕ be an instance of KTH SAT and let $\mathbf{B}_\phi$ be the network constructed from ϕ as described above. Observe that $\Pr(\mathbf{X} = \mathbf{x} \mid C = T) = 0$ if $\mathbf{x}$ represents a non-satisfying value assignment, and $\Pr(\mathbf{X} = \mathbf{x} \mid C = T)$ is equal to the prior probability of $\mathbf{X} = \mathbf{x}$ if $\mathbf{x}$ represents a satisfying value assignment. Furthermore note that the values of the variables that model logical operators are fully determined by the values of their parents. Then, given evidence $C = T$, the k-th MPE corresponds to the lexicographical k-th satisfying value assignment to the variables in ϕ. Thus, given an algorithm for calculating the k-th MPE, we can solve the KTH SAT problem as well. Clearly, the above reduction is a polynomial-time one-Turing reduction from KTH SAT to KTH MPE. This proves $\mathsf{FP}^{\mathsf{PP}}$-hardness of KTH MPE. □

Observe, that the problem remains $\mathsf{FP}^{\mathsf{PP}}$-complete when all nodes have indegree at most two, and all variables are binary.

4 Enumerating Partial MAP

While the MPE-problem is complete for the class NP (solvable by a nondeterministic TM), Partial MAP is complete for $\mathsf{NP}^{\mathsf{PP}}$, i.e., solvable by a nondeterministic TM with access to an oracle for problems in PP. In the previous section we have proven that the Kth MPE-problem is complete for $\mathsf{FP}^{\mathsf{PP}}$, thus solvable by a *metric* TM. Intuitively, this suggests that the Kth Partial MAP-problem is complete for $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$, the class of function problems solvable by a *metric* TM with access to a PP-complete oracle. To our best knowledge, no complete problems have been discussed for this complexity class. We introduce the Kth NumSat-problem, defined as follows.

Kth NumSAT
Instance: Boolean formula $\phi(x_1, \ldots, x_m, \ldots, x_n)$, natural numbers k, l.
Question: What is the lexicographically k-th assignment $x_1 \ldots\ x_m \in \{0,1\}^m$ such that exactly l assignments $x_{m+1} \ldots\ x_n \in \{0,1\}^{n-m}$ satisfy ϕ?

We will prove in the appendix that Kth NumSAT is $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-complete. To prove hardness of Kth Partial MAP, we will use a version of this problem with *bounds* on the probability of the MAP variables.

Kth Partial MAP
Instance: A probabilistic network $\mathbf{B} = (\mathbf{G}, \Gamma)$, evidence variables $\mathbf{E}$ with instantiation $\mathbf{e}$, observable variables $\mathbf{V_{MAP}} \subset \mathbf{V} \setminus \mathbf{E}$, natural number k, rational numbers $0 \leq q \leq r \leq 1$.
Question: What is, within the interval $[q, r]$, the k-th most probable assignment $\mathbf{v_k}$ to the variables in $\mathbf{V_{MAP}}$ given evidence $\mathbf{e}$?

Note that the Kth Partial MAP problem *without* boundary constraints is a special case where $q = 0$ and $r = 1$, and that we can use binary search techniques to find a solution to

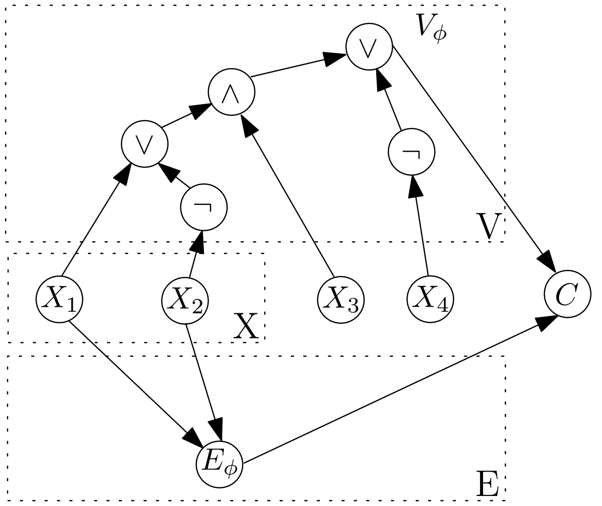

Figure 2: Example of k-th Partial MAP construction for the formula $\phi_{ex} = ((x_1 \vee \neg x_2) \wedge x_3) \vee \neg x_4$, with MAP variables x_1 and x_2

the *bounded* problem variant, using an algorithm for the *unbounded* problem variant, so we can transform a bounded problem variant into an unbounded problem variant in polynomial time, and vice versa. However, using the bounded problem formulation facilitates our hardness proof.

We will prove $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-completeness of Kth Partial MAP by a reduction from Kth NumSAT. We will again use the formula $\phi_{ex} = ((x_1 \vee \neg x_2) \wedge x_3) \vee \neg x_4$ as a running example (see Figure 2). We want to find the lexicographically k-th assignment to $\{x_1, x_2\}$ such that exactly l instantiations to $\{x_3, x_4\}$ satisfy ϕ_{ex}.

As in the previous section, we construct a probabilistic network $\mathbf{B}_\phi$ from a given Kth NumSAT instance $\phi(x_1, \ldots, x_m, \ldots, x_n)$. Again, we create a stochastical variable X_i for each variable x_i in ϕ, but now with uniform probability. We denote the variables $X_i, \ldots, X_m$ as the variable instantiation part (X). These variables are the MAP variables in our k-th Partial MAP construction. For each logical operator in ϕ, we create additional variables in the network as in the previous section, with V_ϕ as variable associated with the top level operator in ϕ. Observe that, for a particular value assignment $\mathbf{v_k}$ to the MAP variables

$\{X_1, \ldots, X_m\}$, $\Pr(V_\phi = T) = \frac{l}{n-m}$, where l is the number of value assignments to the variables $\{X_{m+1}, \ldots, X_n\}$ that satisfy ϕ.

Furthermore, we construct a *enumeration* part (E) of the network by constructing a $\log n$-deep binary tree with the MAP variables $X_1, \ldots, X_m$ as leafs and additional variables $E_{p,q}$, each with possible values *true* and *false*. Without loss of generality, we assume that the number of leafs is a power of two (we can use additional dummy variables). A variable $E_{p,1}$ has parents X_{2p-1} and X_{2p}; variables $E_{p,q} (q > 1)$ have parents $E_{2p-1,q-1}$ and $E_{2p,q-1}$. Let $\pi(E_{p,q}) = \{X_{2p-1}, X_{2p}\}$, respectively $\{E_{2p-1,q-1}, E_{2p,q-1}\}$ denote the parent configuration for $E_{p,1}$ and $E_{p,q} (q > 1)$. Then the conditional probability table for $E_{p,q}$ is defined as follows:

$$\begin{aligned}
\Pr(E_{p,q} = T \,|\, \pi(E_{p,q}) = \{T, T\}) &= 0 \\
\Pr(E_{p,q} = T \,|\, \pi(E_{p,q}) = \{T, F\}) &= \tfrac{1}{2^{p+n-m+1}} \\
\Pr(E_{p,q} = T \,|\, \pi(E_{p,q}) = \{F, T\}) &= \tfrac{2}{2^{p+n-m+1}} \\
\Pr(E_{p,q} = T \,|\, \pi(E_{p,q}) = \{F, F\}) &= \tfrac{3}{2^{p+n-m+1}}
\end{aligned}$$

The root of this tree will be denoted as E_ϕ. In the example network, there are only two MAP variables ($m = 2$) so $E_\phi = E_{1,1}$ with probabilities $\Pr(E_\phi = T) = 0, \frac{1}{16}, \frac{2}{16}$, and $\frac{3}{16}$ for the value assignments $\{T, T\}$, $\{T, F\}$, $\{F, T\}$ and $\{F, F\}$, respectively. Note that the above construct ensures that lexicographically smaller value assignments to the MAP variables, lead to a higher probability $\Pr(E_\phi = T)$, but that this probability is always less than $\frac{1}{2^{n-m}}$.

We add an additional variable C with parents V_ϕ and E_ϕ, with the following conditional probability table:

$$\Pr(C = T) = \begin{cases} 1 & \text{if } V_\phi = T \wedge E_\phi = T \\ \frac{1}{2} & \text{if } V_\phi = T \wedge E_\phi = F \\ \frac{1}{2} & \text{if } V_\phi = F \wedge E_\phi = T \\ 0 & \text{if } V_\phi = F \wedge E_\phi = F \end{cases}$$

We now have, that for a particular instantiation to the MAP variables, the probability $\Pr(C = T)$ is within the interval $[\frac{l}{2^{n-m}}, \frac{l+1}{2^{n-m}}]$, where l denotes the number of value assignments to the variables $X_{m+1}, \ldots, X_n$ that make ϕ true.

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

Proof. The $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$ membership proof is very similar to the $\mathsf{FP}^{\mathsf{PP}}$ membership proof of the Kth MPE-problem, but now we use an oracle for Exact Inference (which is #P-complete, see Roth (1996)) to compute the probability of the assignment $\mathbf{v_k}$. If it is within the interval $[q, r]$, we output 1 minus that probability; if not, we output 1. Note that we really need the oracle to perform this computation since we need to marginalize on $\mathbf{v_k}$. Clearly, $\text{KthValue}_{\hat{\mathcal{M}}}$ returns the k-th Partial MAP, and this proves that Kth Partial MAP is in $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$.

To prove hardness, we construct a probabilistic network $\mathbf{B}_\phi$ from a given instance $\phi(x_1, \ldots, x_m, \ldots, x_n)$, similar to the previous section. The conditional probabilities in the thus constructed network ensure that the probability of a value assignment $\mathbf{v_k}$ to the variables $\{X_1, \ldots, X_m\}$ such that l value assignments to the variables $\{X_{m+1}, \ldots, X_n\}$ satisfy ϕ, is in the interval $[\frac{l}{2^{n-m}}, \frac{l+1}{2^{n-m}}]$. Moreover, $\Pr(C = T \mid \mathbf{x_k}) > \Pr(C = T \mid \mathbf{x'_k})$ if the truth value that corresponds with $\mathbf{x_k}$ is lexicographically smaller than $\mathbf{x'_k}$. Thus, with evidence $C = T$ and ranges $[\frac{l}{2^{n-m}}, \frac{l+1}{2^{n-m}}]$, the k-th Partial MAP corresponds to the lexicographical k-th truth assignment to the variables $x_1 \ldots x_m$ for which exactly l truth assignments to $x_{m+1} \ldots x_n$ satisfy ϕ. Clearly, the above reduction is a polynomial-time one-Turing reduction from Kth NumSAT to Kth Partial Map. This proves $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-hardness of Kth Partial MAP. □

Observe again, that the problem remains $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-complete when the MAP-variables have no incoming arcs, when all nodes have indegree at most two, and all variables are binary.

5 Conclusion

In this paper, we have addressed the computational complexity of finding the k-th MPE or k-th Partial MAP. We have shown that the Kth MPE-problem is P^{PP}-complete, making it considerably harder than both MPE (which is NP-complete) and INFERENCE (which is PP-complete). The computational power (and thus the intractability of KTH MPE) of P^{PP} is illustrated by Toda's theorem (1991) that states that P^{PP} includes the entire Polynomial Hierarchy. Yet finding the k-th MPE is arguably easier than finding the most probable explanation given only partial evidence (the PARTIAL MAP-problem) which is $\mathsf{NP}^{\mathsf{PP}}$-complete. Moreover, when inference can be done in polynomial time (such as in polytrees) then we can find the k-th MPE in polynomial time (Sy, 1992; Srinivas and Nayak, 1996).

Finding the k-th Partial MAP, on the other hand, is considerably harder. We have shown that this problem is $\mathsf{P}^{\mathsf{PP}^{\mathsf{PP}}}$-complete in general. Park and Darwiche (2004) show that the PARTIAL MAP-problem remains NP-complete on polytrees, using a reduction from 3SAT[4]. Their proof can be easily modified to reduce KTH PARTIAL MAP on polytrees from the P^{PP}-complete problem KTH 3SAT (Toda, 1994), hence finding the k-th Partial MAP on polytrees remains P^{PP}-complete. Nevertheless, the approach of Park and Darwiche (2004) for approximating PARTIAL MAP may be extended to find the k-th Partial MAP as well.

For small or fixed k, these problems may be easier, depending on the exact problem formulation[5]. For example, it may be the case that KTH MPE is *fixed-parameter tractable*, i.e. an algorithm exists for KTH MPE which has a running time, exponentially only in k.

Acknowledgements

This research has been (partly) supported by the Netherlands Organisation for Scientific Research (NWO).
The author wishes to thank Hans Bodlaender and Gerard Tel for their insightful comments on earlier drafts of this paper, and Leen Torenvliet for discussions on the KTH NUMSAT problem.

[4]Technically, they reduce PARTIAL MAP from MAX SAT to preserve approximation results.

[5]The problem 'Are there *at least* k value assignments with a probability at least q' is trivially in NP for $k \leq \log n$, but when we want to know whether there are *exactly* k such assignments the problem may be considerable harder.

References

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

I. Beinlich, G. Suermondt, R. Chavez, and G. Cooper. 1989. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Proceedings of the Second European Conference on AI and Medicine*, pages 247–256.

H. L. Bodlaender, F. van den Eijkhof, and L. C. van der Gaag. 2002. On the complexity of the MPA problem in probabilistic networks. In *Proceedings of the Fifteenth European Conference on Artificial Intelligence*, pages 675–679.

H. Chan and A. Darwiche. 2006. On the robustness of most probable explanations. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pages 63–71.

E. Charniak and S. E. Shimony. 1994. Cost-based abduction and map explanation. *Artificial Intelligence*, 66(2):345–374.

S. A. Cook. 1971. The complexity of theorem proving procedures. In *Annual ACM Symposium on Theory of Computing*, pages 151–158.

M. R. Garey and D. S. Johnson. 1979. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W. H. Freeman and Co., San Francisco.

F. V. Jensen. 2007. *Bayesian Networks and Decision Graphs.* Berlin: Springer Verlag, second edition.

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

M. L. Littman, J. Goldsmith, and M. Mundhenk. 1998. The computational complexity of probabilistic planning. *Journal of Artificial Intelligence Research*, 9:1–36.

C. H. Papadimitriou. 1994. *Computational Complexity.* Addison-Wesley.

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

J. Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, Palo Alto.

D. Roth. 1996. On the hardness of approximate reasoning. *Artificial Intelligence*, 82(1-2):273–302.

E. Santos Jr. 1991. On the generation of alternative explanations with implications for belief revision. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 339–347.

S. E. Shimony. 1994. Finding MAPs for belief networks is NP-hard. *Artificial Intelligence*, 68(2):399–410.

J. Simon. 1977. On the difference between one and many. In *Proceedings of the Fourth Colloquium on Automata, Languages, and Programming*, volume 52 of *LNCS*, pages 480–491. Springer-Verlag.

S. Srinivas and P. Nayak. 1996. Efficient enumeration of instantiations in Bayesian networks. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence*, pages 500–508.

B.K. Sy. 1992. Reasoning MPE to multiply connected belief networks using message-passing. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 570–576.

S. Toda. 1991. PP is as hard as the polynomial-time hierarchy. *SIAM Journal of Computing*, 20(5):865–877.

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

J. Torán. 1991. Complexity classes defined by counting quantifiers. *Journal of the ACM*, 38(3):752–773.

L. C. van der Gaag, S. Renooij, C. L. M. Witteman, B. M. P. Aleman, and B. G. Taal. 2002. Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25:123–148.

Appendix

In Section 4 we reduced KTH NUMSAT to KTH PARTIAL MAP. Here we show that KTH NUMSAT is $\mathsf{FP}^{\mathsf{PP}^{\#\mathsf{P}}}$-complete, and thus also $\mathsf{FP}^{\mathsf{PP}^{\mathsf{PP}}}$-complete.

KTH NUMSAT
Instance: Boolean formula $\phi(x_1, \ldots, x_m, \ldots, x_n)$, natural numbers k, l.
Question: What is the lexicographically k-th assignment $x_1 \ldots x_m \in \{0,1\}^m$ such that exactly l assignments $x_{m+1} \ldots x_n \in \{0,1\}^{n-m}$ satisfy ϕ?

The hardness proof of KTH NUMSAT is based on the $\mathsf{FP}^{\mathsf{PP}}$-hardness proof of KTH SAT by Toda (1994), and uses a result by Torán (1991) that states that the Counting Hierarchy (and thus $\mathsf{P}^{\mathsf{PP}^{\#\mathsf{P}}}$ in particular) is closed under polynomial time many-one reductions (and consequently, the functional counterpart $\mathsf{FP}^{\mathsf{PP}^{\#\mathsf{P}}}$ is closed under polynomial time one-Turing reductions). Thus, any computation in $\mathsf{FP}^{\mathsf{PP}^{\#\mathsf{P}}}$ can be modeled by a metric TM that calculates a bit string q based on its input x, then queries its $\#\mathsf{P}$ oracle and writes down a number based on q and the result of the oracle, thus only querying the oracle once.

Theorem 3. KTH NUMSAT *is* $\mathsf{FP}^{\mathsf{PP}^{\#\mathsf{P}}}$*-complete.*

Proof. Since Toda's proof (Toda, 1994) relativizes, a function f is in $\mathsf{FP}^{\mathsf{PP}^{\#\mathsf{P}}}$ if there exists a metric TM $\hat{\mathcal{M}}$ with access to an oracle for $\#\mathsf{P}$-complete problems such that $f \leq^{\mathsf{FP}}_{1-T}$ $\text{KthValue}_{\hat{\mathcal{M}}}$. It is easy to see that a metric TM, that nondeterministically computes a satisfying assignment to $x_1 \ldots x_m$ (using an oracle for counting the number of satisfying assignments to $x_{m+1} \ldots x_n$), and writing the binary representation of this assignment on its output tape, suffices.

To prove hardness, let $\hat{\mathcal{M}}$ be a metric TM with a $\#\mathsf{P}$ oracle. Given an input x to $\hat{\mathcal{M}}$, we can construct (using Cook's theorem (1971)) a tuple of two Boolean formulas $(\phi_x(q), \psi_x(r))$ such that ϕ_x is true if and only if q specifies a computation path of $\hat{\mathcal{M}}$ that is presented to the $\#\mathsf{P}$ oracle, which returns the number l of satisfying instantiations to $\psi_x(r)$, such that $F(q, l)$ is the output of $\hat{\mathcal{M}}$. Since the computation path that computes q is uniquely determined, q is the k-th satisfying assignment to ϕ_x for which l instantiations to r satisfy $\psi_x(r)$, if and only if $F(q, l)$ is the k-th output of $\hat{\mathcal{M}}$. Thus, we can construct a $\leq^{\mathsf{FP}}_{1-T}$-reduction from every function accepted by a metric TM with access to a $\#\mathsf{P}$ oracle to KTH NUMSAT. $\square$