Conditional Lower Bounds on the Complexity of Probabilistic Inference

Johan Kwisthout

Hans L. Bodlaender

Technical Report UU-CS-2009-018
August 2009

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

ISSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

Conditional Lower Bounds on the Complexity of Probabilistic Inference

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

August 2009

Abstract

The INFERENCE problem in probabilistic networks (given a stochastic variable V, what is the posterior probability that $V = v$ given evidence $\mathbf{e}$?) has been proven to be intractable; in fact, has a PP-complete decision variant [17]. The currently most efficient algorithms for this problem are all exponential in the treewidth of the moralised graph of the network. We prove, using a recent result of Marx [18], that these algorithms are in some sense optimal: we prove a lower bound of $f(\mathbf{G})^{\omega(\frac{\mathrm{tw}(\mathbf{G})}{\log \mathrm{tw}(\mathbf{G})})}$ for any algorithm solving arbitrary instances of INFERENCE with graph $\mathbf{G}$, unless the ETH fails. To obtain this lower bound we introduce *treewidth-preserving reductions* which may be of independent interest.

1 Introduction

Probabilistic networks [20, 13], also called *Bayesian* or *belief* networks, are one of the formalisms that can be used to represent and reason about uncertain information. A probabilistic network, denoted as $\mathcal{B} = (\mathbf{G}, \Gamma)$, represents a joint probability distribution on a set of stochastic variables, and is described by a directed acyclic graph $\mathbf{G}$ and a set of conditional probabilities Γ. The nodes of the graph represent the stochastic variables, the arcs (or lack of them) represent (in)dependencies in the joint probability distribution. The structure of the graph and the conditional probabilities in the network can be either *elicited* from domain experts, or *learned* from data. Probabilistic networks are often used in decision support systems, mainly—but not exclusively—in medical diagnosis systems, see, e.g., the networks described in [23],[24],[9], or [11]. In such systems, the network typically consists of *classification* (or *output*), *observable* (or *evidence*), and *intermediate* nodes.

Arguably the most important computational problem related to probabilistic networks, is determining the posterior probability distribution $\Pr(V_i \,|\, \mathbf{e})$ of some variable V_i, given evidence $\mathbf{e}$ for other variables in the network. This problem, known as the INFERENCE problem, occurs if we want to compute the probability distribution on a particular classification variable, given knowledge of the values of some observable variables. But the problem also arises if we want to compute the likeliness of a set of observable variables, given the values of the classification nodes in the network. The posterior probability distribution of any node (or set of nodes) can be calculated using well known properties in probability theory, however,

this calculation can take time, exponential in the size of the network. Typical algorithms for probabilistic inference are the *loop cutset conditioning* [20], *variable elimination* [22], and *clique tree propagation* [15] approaches (also called *join tree* propagation or *clustering*). When the network structure is restricted to be a polytree (or *singly connected* graph), inference can be done in polynomial time, for example using the message passing algorithm [20]. However, for multiply connected graphs, no polynomial time algorithm exists.

In general, no polynomial time algorithms are known for the inference problem in probabilistic networks, and indeed the INFERENCE problem has a PP-complete decision variant [17]. Nevertheless, for probabilistic networks whose moralised graphs[1] have *bounded treewidth*, the algorithms discussed above are exponential *only* in the treewidth of the moralised graph. In this paper, we show that a low treewidth of the moralised graph is actually a *necessary* condition for the network[2] to make inference efficient in an algorithm accepting arbitrary inference instances, under the assumption that the so-called *Exponential Time Hypothesis* (ETH) holds. In other words, we show that no algorithm can exist that solves arbitrary inference instances with unbounded treewidth in sub-exponential time, unless the ETH fails. Nevertheless, there might be some specific structure (that we are unaware of), for example in the *arc orientation*, that may be exploited by an algorithm to solve *particular* instances in polynomial time.

Whether or not low treewidth is a necessary condition for algorithms to run in polynomial time has also been investigated recently for binary constraint satisfaction and graph homomorphism problems [10] and for inference in undirected graphical models [6]. Using the assumption that FPT $\neq$ W[1], respectively that NP $\not\subseteq$ P/poly, they showed that these problems have no algorithm solving instances with unbounded treewidth in polynomial time. Marx [18] proved a sub-exponential lower bound (up to a logarithmic factor in the exponent) for the constraint satisfaction and graph homomorphism problems with unbounded treewidth, assuming that the ETH holds. A related result by Chen et al. [7] proved a sub-exponential lower bound for K-CLIQUE, even without logarithmic factor, assuming that not all problems in SNP have subexponential algorithms[3] These known results are summarised in Table 1.

We will take Marx' result [18] as a starting point. In this paper, we introduce *treewidth preserving reductions*, and we show that a polynomial-time treewidth preserving reduction from CONSTRAINT SATISFACTION to INFERENCE exists. This proves, that if an algorithm for INFERENCE exists that can solve arbitrary instances with high treewidth in sub-exponential time, then this algorithm can also solve such instances of CONSTRAINT SATISFACTION in sub-exponential time, which in turn would contradict the ETH according to Marx' result [18]. Treewidth preserving reductions, as a means to prove conditional lower bounds, may be of independent interest.

We conclude from our result, that Marx' conditional lower bound [18] of $f(\mathbf{G})^{\omega(\frac{\mathrm{tw}(\mathbf{G})}{\log \mathrm{tw}(\mathbf{G})})}$ for the constraint satisfaction problem also holds for inference on probabilistic networks, showing that the algorithms discussed above are optimal, up to a logarithmic factor in the exponent, for arbitrary networks. After introducing some preliminaries and notation in Section 2, we will sketch our approach in Section 3, and we present our main results in Section 4. The

[1] A moralised graph is the undirected graph obtained from a directed graph by connecting nodes that have a common child, and then dropping the arc directions.

[2] Note that we are interested in properties of the graph, rather than the probability distribution. There exists a trivial probability distribution (the uniform distribution where the probability distribution of each variable is independent of the values of its parents) in every network in which inference is trivial.

[3] Observe that a graph with a k-clique necessarily also has a treewidth of at least k, but that the opposite does not hold.

Study	Topic	Assumptions	Reduction	Result
Grohe [10]	CSP, graph homomorphism	FPT $\neq$ W[1], unbounded variable cardinality	k-Clique	no polyn. algorithms for instances with unbounded treewidth
Marx [18]	CSP, graph homomorphism	ETH, unbounded variable cardinality	n-variable 3Sat	$f(\mathbf{G})^{\omega(\frac{\mathrm{tw}(\mathbf{G})}{\log \mathrm{tw}(\mathbf{G})})}$ lower bound for instances with unbounded treewidth
Chandrasekaran [6]	inference in graphical models	NP $\neq$ P/poly, Graph Minor Hypothesis	non-uniform Max2Sat	no polyn. algorithms for instances with unbounded treewidth
Chen et al. [7]	K-Clique	Not all problems in SNP have subexp. algorithms	weighted antitone 2CNFSat	$f(\mathbf{G})^{\omega(k)}$ lower bound for k-clique
current study	inference in probabilistic networks	ETH, unbounded variable cardinality	treewidth-preserving reduction from CSP	$f(\mathbf{G})^{\omega(\frac{tw(\mathbf{G})}{\log \mathrm{tw}(\mathbf{G})})}$ lower bound for algorithm solving arbitrary instances with unbounded treewidth

Table 1: Comparison between various lower bound results

paper will be concluded in Section 5.

2 Preliminaries

A probabilistic network models a set of stochastic variables, the (in-)dependencies among these variables, and a joint probability distribution over these variables. The variables and dependences in the network are modelled by a directed acyclic graph $\mathbf{G} = (\mathbf{V}, \mathbf{A})$. A node $X \in \mathbf{V}$ is called a parent of $Y \in \mathbf{V}$, and Y is called a child of X, if $(X, Y) \in \mathbf{A}$. The set of all parents of Y is denoted as $\pi(Y)$, the set of all children of X is denoted as $\sigma(X)$. A probabilistic network is formally defined as follows.

Definition 2.1 (probabilistic network). *A probabilistic network, denoted by $\mathcal{B}$, is a tuple $(\mathbf{G}, \Gamma)$, where $\mathbf{G} = (\mathbf{V}, \mathbf{A})$ is a directed acyclic graph, and Γ denotes the set of conditional probability distributions on $\mathbf{V}$, coded into conditional probability tables or CPTs. The network models a joint probability distribution $\Pr(\mathbf{V}) = \prod_{i=1}^{n} \Pr(x_i \,|\, \pi(X_i))$ over its variables.*

We will use upper case letters to denote individual variables in the network, upper case bold letters to denote sets of variables, lower case letters to denote value assignments to variables, and lower case bold letters to denote joint value assignments to sets of variables. If not specified otherwise, we will use indexed lower case variables $x_1, x_2, \ldots, x_n$ to denote the values of a variable X, and assume that the set of values of a variable is ordered such that $x_i < x_j$ whenever $i < j$. We will use the notation $X = x$, respectively $\mathbf{X} = \mathbf{x}$ to denote an unspecified (joint) value assignment to X and $\mathbf{X}$, respectively. If no ambiguity can occur, we will write $\Pr(x)$ as a shorthand for $\Pr(X = x)$.

The Inference problem in probabilistic networks, discussed above, is formally defined as follows.

Inference

Instance: Let $\mathcal{B} = (\mathbf{G}, \Gamma)$ be a probabilistic network, and let Pr be its joint probability distribution. Let $X \in \mathbf{V}$, with $x \in \Omega(X)$, furthermore let $0 \leq q \leq 1$.
Question: Is $\Pr(X = x) \geq q$?

Throughout this paper, we assume that the reader is familiar with basic notions of computational complexity and hardness proofs. We refer to classical textbooks like [8] and [19] for a thorough introduction to these subjects.

A *tree-decomposition* [21] of an undirected graph is defined as follows.

Definition 2.2 (tree decomposition). *A tree decomposition of a graph* $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ *is a pair* $\langle \mathcal{X}, T \rangle$*, where* $T = (I, F)$ *is a tree, and* $\mathcal{X} = \{\mathbf{X_i} \mid i \in I\}$ *is a family of subsets (or bags) of* $\mathbf{V}$*, one for each node of* T*, such that*

- $\bigcup_{i \in I} \mathbf{X_i} = \mathbf{V}$,
- *for every edge* $(V, W) \in \mathbf{E}$ *there exists an* $i \in I$ *with* $V \in \mathbf{X_i}$ *and* $W \in \mathbf{X_i}$*, and*
- *for every* $i, j, k \in I$*: if* j *is on the path from* i *to* k *in* T*, then* $\mathbf{X_i} \cap \mathbf{X_k} \subseteq \mathbf{X_j}$.

The width of a tree decomposition $((I, F), \{\mathbf{X_i} \mid i \in I\})$ *is* $\max_{i \in I} \mid \mathbf{X_i} \mid -1$*. The treewidth of* $\mathbf{G}$*, denoted by* $\mathrm{tw}(\mathbf{G})$ *is the minimum width over all tree decompositions of* $\mathbf{G}$.

For every fixed W, there is an algorithm using $\mathcal{O}(n)$ time, that, given a graph $\mathbf{G}$ with n vertices, decides if the treewidth of $\mathbf{G}$ is at most W and, if so, constructs a tree decomposition of $\mathbf{G}$ with width W [2]. While this linear algorithm is not very practical due to its very large constants, efficient heuristics and exact algorithms for small values of W are known; see e.g. [4].

A so-called *nice* tree decomposition is a tree decomposition with a particularly simple structure. In particular, the tree is rooted and every node in a nice tree decomposition has at most two children. A node in a nice tree decomposition is either a LEAF NODE, INTRODUCE NODE, FORGET NODE, or JOIN NODE.

- A LEAF NODE i is a leaf of T with $\mid \mathbf{X_i} \mid = 1$.
- An INSERT NODE i has one child j with $\mathbf{X_i} = \mathbf{X_j} \cup \{Y\}$ for some vertex $Y \in \mathbf{V}$, and $\mid \mathbf{X_i} \mid = \mid \mathbf{X_j} \mid +1$.
- A FORGET NODE i has one child j with $\mathbf{X_i} = \mathbf{X_j} \setminus \{Y\}$ for some vertex $Y \in \mathbf{V}$, and $\mid \mathbf{X_i} \mid = \mid \mathbf{X_j} \mid -1$.
- A JOIN NODE i has two children j, k where $\mathbf{X_i} = \mathbf{X_j} = \mathbf{X_k}$.

Every tree decomposition T with treewidth W and m nodes can be converted to a nice tree-decomposition of the same width and $\mathcal{O}(m)$ nodes in time $\mathcal{O}(f(W) \cdot m)$ for a polynomial function f [3, 14].

The complexity of a probabilistic network is normally measured by the minimal treewidth of a triangulation of the moral graph of the network. Let $\mathcal{B} = (\mathbf{G}, \Gamma)$ be a probabilistic network, where $\mathbf{G} = (\mathbf{V}, \mathbf{A})$ is a directed acyclic graph with set of vertices $\mathbf{V} = \{V_1, \ldots, V_n\}$ and set of arcs $\mathbf{A}$. The *moralisation* $\mathbf{G_M}$ of $\mathbf{G}$ is the undirected graph obtained from $\mathbf{G}$ by adding arcs connecting all parents of a variable, and then dropping all arc directions. A *triangulation* of $\mathbf{G_M}$ is any graph $\mathbf{G_T}$, such that $\mathbf{E}(\mathbf{G_M}) \subseteq \mathbf{E}(\mathbf{G_T})$ and $\mathbf{G_T}$ is *chordal*, i.e.

every cycle in $\mathbf{G_T}$ of more than three nodes has an edge connecting two non-adjacent nodes in the cycle. The treewidth of this chordal graph, denoted as $\text{tw}(\mathbf{G_T})$, is the size of the largest clique in $\mathbf{G_T}$ minus one.

Many computational problems can be formulated as *constraint satisfaction problems* (CSPs). A CSP $\mathcal{I}$ is defined as a 3-tuple $\langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$, where $\mathbf{V}$ denotes a set of variables, $\mathbf{D}$ denotes a domain of variables, and $\mathbf{C}$ denotes a set of constraints, which are defined as tuples $\langle t, \mathbf{R} \rangle$. In these tuples, t denotes a tuple of variables and $\mathbf{R}$ denotes a set of tuples of values defining relations between variables. A solution to $\mathcal{I}$ is a function f from $\mathbf{V}$ to $\mathbf{D}$ such that for each constraint $\langle t, \mathbf{R} \rangle$, with $t = \langle v_1, \ldots, v_m \rangle$, $\langle f(v_1), \ldots, f(v_m) \rangle \in \mathbf{R}$. The *primal graph* $\mathbf{G}$ of $\mathcal{I}$ is an undirected graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $(V_1, V_2) \in \mathbf{E}$ if and only if there is a constraint $\langle t, \mathbf{R} \rangle \in \mathbf{C}$ with $V_1, V_2 \in t$.

3 Approach

In Marx' paper [18], the following result was proven.

Theorem 3.1 (CSP Lower Bound). *If there is a recursively enumerable class $\mathcal{G}$ of graphs with unbounded treewidth $\text{tw}(\mathbf{G})$ and a computable function f such that CSP($\mathcal{G}$) can be decided by an algorithm running in time $f(\mathbf{G})\|\mathcal{I}\cdot\|^{o(\frac{\text{tw}(\mathbf{G})}{\log \text{tw}(\mathbf{G})})}$ for instances $\mathcal{I}$ with primal graph $\mathbf{G} \in \mathcal{G}$, then the* ETH *fails.*

This result also holds for CSPs with binary relations; in the remainder, we assume that all CSP instances are binary.

The *Exponential Time Hypothesis* (ETH), introduced by Impagliazzo et al. [12], states that there exists a constant $c > 1$ such that deciding any 3Sat instance with n variables takes at least $\Omega(c^n)$ time. Note that the ETH is a stronger assumption than the assumption that P $\neq$ NP. A sub-exponential, but not polynomial-time, algorithm for 3Sat (comparable with the General Number Field Sieve algorithm for Integer Factorisation, see [16]) contradicts the ETH, but not P $\neq$ NP.

In the remainder of this paper, we will build on Marx' result. We will reduce Constraint Satisfaction to Inference, using a polynomial-time, treewidth preserving reduction, i.e., given an instance $\mathcal{I}$ of Constraint Satisfaction, we construct in polynomial time an instance $(\mathcal{B}, V, v, q)$ of Inference with the same treewidth (up to a constant) such that a solution to $(\mathcal{B}, V, v, q)$ yields also a solution to $\mathcal{I}$. This means intuitively that, if an algorithm A exists that solves arbitrary instances of Inference with high treewidth in sub-exponential time, then we can construct an algorithm B solving instances of Constraint Satisfaction with high treewidth in sub-exponential time, thus contradicting the ETH. We formalise our result in the following main theorem, which we will prove in Section 4.

Theorem 3.2 (Inference Lower Bound). *If there is a computable function f such that an arbitrary instance $(\mathcal{B}, V, v, q)$ with moralised graph $\mathbf{G_M}$ of the* Inference *problem on probabilistic networks can be decided by an algorithm running in time $f(\mathbf{G_M}) \cdot \|\mathcal{B}\|^{o(\frac{\text{tw}(\mathbf{G_M})}{\log \text{tw}(\mathbf{G_M})})}$, then the* ETH *fails.*

4 Complexity of Inference

It is often desired that a reduction between decision problems also preserves some other property. For example, parsimonious reductions [8] preserve the number of solutions, and approximation-preserving reductions [1] preserve approximation results. We introduce a reduction on graphical structures that preserves a structural property of a problem instance, namely the *treewidth* of the instance. Since high treewidth is often an indicator of intractability of a problem, it is desirable that a reduction preserves this property.

Definition 4.1 (Polynomial-time treewidth-preserving reductions). *Let A and B be problems such that treewidth is defined on instances of A and B. A is polynomial-time treewidth-preserving reducible to B (denoted $A \leq^p_{\text{tw}} B$), if there exists a polynomial time computable function f and a constant L such that for all instances $x : x \in A \iff f(x) \in B$ and $\text{tw}(f(x)) \leq L \cdot \text{tw}(x)$. We will call such a reduction (f, l) a* tw*-reduction.*

We show that Constraint Satisfaction is tw-reducible to Inference, using a polynomial time treewidth-preserving reduction. We will first formally define the Constraint Satisfaction problem, which was introduced in Section 2.

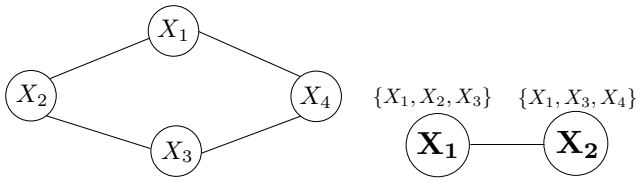
Constraint Satisfaction
Instance: $\langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$, where $\mathbf{V}$ denotes a set of variables, $\mathbf{D}$ a domain of values, and $\mathbf{C}$ a set of constraints; every constraint is a tuple $\langle\ t, \mathbf{R} \rangle$, where t is a tuple of variables from $\mathbf{V}$ and $\mathbf{R}$ is a binary relation over $\mathbf{D}$.
Question: Is there an function f from the set of variables $\mathbf{V}$ to the domain $\mathbf{D}$ that satisfies all constraints in $\mathbf{C}$?

Given a Constraint Satisfaction instance $\mathcal{I} = \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$, there is a straightforward way to construct a probabilistic network $\mathcal{B}_\mathcal{I} = (\mathbf{G_I}, \Gamma)$ that simulates $\mathcal{I}$. However, care must be taken to ensure that the construction preserves treewidth. We will first show how to construct a network $\mathcal{B}_\mathcal{I}$ that captures the information from the Constraint Satisfaction instance $\mathcal{I}$, and then show how the tree decomposition of $\mathcal{B}_\mathcal{I}$ can be used to construct a network $\mathcal{B}'_\mathcal{I}$ with the same treewidth as the primal graph of $\mathcal{I}$, with a designated variable A_m with values TRUE and FALSE. We will then show that in this network $\mathcal{B}'_\mathcal{I}$, $\Pr(A_m = \text{TRUE}) > 0$ if and only if $\mathcal{I}$ has a solution.

We will illustrate this construction for the example problem $\mathcal{I}_{\text{ex}} = \langle \mathbf{V}, \mathbf{D}, \mathbf{C} \rangle$, with $\mathbf{V} = \{X_1, X_2, X_3, X_4\}, \mathbf{D} = \{a, b, c\}$, and $\mathbf{C} = \{\ \langle (X_1, X_2), \{(a, a), (b, a)\} \rangle, \langle (X_1, X_3), \{(a, b), (a, c), (b, b), (b, c)\} \rangle, \langle (X_1, X_4), \{(a, a), (a, b), (b, a), (c, a)\} \rangle, \langle (X_2, X_3), \{(a, b), (b, c)\} \rangle, \langle (X_3, X_4), \{(b, a), (b, b)\} \rangle\ \}$. Note that $\mathcal{I}_{\text{ex}}$ is satisfiable; an example solution f_{ex} sets X_1 to b, X_2 to a, X_3 to b, and X_4 to a. The primal graph of $\mathcal{I}_{\text{ex}}$ is given in Figure 2(a) and the resulting tree-decomposition in Figure 2(b).

For every variable $X_i \in \mathbf{V}$ in $\mathcal{I}$, we add a root node X_i to $\mathcal{B}_\mathcal{I}$ with the domain D as values, with a uniform distribution. For every relation $R_j \in \mathbf{R}$ in $\mathcal{I}$, we add a node R_j in $\mathcal{B}_\mathcal{I}$, with the variables that occur in the relation as parents, and values TRUE and FALSE. We set $\Pr(R_j = \text{TRUE} | \mathbf{x}) = 1$ for any combination of values $\mathbf{x}$ for its parents such that $\mathbf{x}$ is a tuple in R_j, and we set $\Pr(R_j = \text{TRUE} | \mathbf{x}) = 0$ if $\mathbf{x}$ is *not* a tuple in R_j. See Figure 3 for the network $\mathcal{B}_{\mathcal{I}_{\text{ex}}}$, constructed from the example $\mathcal{I}_{\text{ex}}$.

With respect to the treewidth of the thus constructed graph $\mathcal{B}_\mathcal{I}$ we observe the following. Standard arguments from the theory of treewidth show that the treewidth of the moralised



(a) Primal graph (b) Tree-decomposition

Figure 2: The primal graph (a) and resulting tree-decomposition (b) of CSP instance $\mathcal{I}_{\text{ex}}$

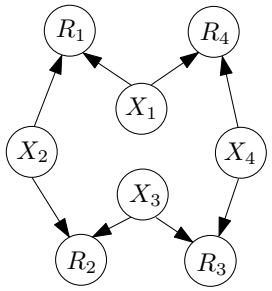


Figure 3: Graph $\mathbf{G}_{\text{ex}}$ constructed from CSP instance $\mathcal{I}_{\text{ex}}$

graph of $\mathcal{B}_\mathcal{I}$ equals $\min(2, \text{tw}(\mathcal{I}))$: the nodes of the form R_j are simplicial in the moralised graph, i.e., they are adjacent to a complete set of nodes, and for any simplicial node V with degree d in a graph $\mathbf{G}$, the treewidth of $\mathbf{G}$ equals the maximum of d and the treewidth of $\mathbf{G} \setminus V$ (see e.g. [5]).

To combine the information from the nodes representing the various relations, we add extra variables that mimic the function of an 'and'-operator. This has to been done with care to avoid an exponential blow-up of the treewidth of the moralised graph $\mathbf{G_M}$ of $\mathcal{B}_\mathcal{I}$. Such a blow-up might appear if we just add one designated node A_m with all $R_j \in \mathbf{R}$ as its parents, or if we construct a log-deep binary tree to connect the relations in $\mathbf{R}$ with A_m. We use the structure of the tree-decomposition of $\mathbf{G_M}$ to construct $\mathcal{B}'_\mathcal{I}$ from $\mathcal{B}_\mathcal{I}$.

Let $\mathbf{G_M}$ be the moralised graph of $\mathcal{B}_\mathcal{I}$, and let $\mathbf{T_G}$ be a tree-decomposition of $\mathbf{G_M}$ in which every node has at most two children. This is, e.g., the case if $\mathbf{T_G}$ is a *nice* tree-decomposition. As we have seen in Section 2, every graph has a nice tree-decomposition; moreover a nice tree-decomposition can be computed in time $\mathcal{O}(f(\text{tw}(\mathbf{G_M})) \cdot \|\mathbf{G_M}\|)$ for a particular computable function f [3, 14]. While we assume in the proof of Theorem 4.2 that the tree-decomposition is nice, we use a non-nice tree-decomposition in our example. The reason for this is straightforward: since *niceness* imposes more constraints on a tree-decomposition than needed in our construction, a nice tree-decomposition of $\mathbf{G_M}$ would simply be too large to fit as an example.

We obtain $\mathcal{B}'_\mathcal{I}$ by adding additional variables $A_1, \ldots, A_m$ and arcs (R_j, A_k) to $\mathcal{B}_\mathcal{I}$ as follows. For every node k in $\mathbf{T_G}$, we add a node A_k, and for every edge (k, l) (where $k < l$) in $\mathbf{T_G}$, we add an arc (A_l, A_k). Furthermore, we add arcs (R_j, A_k), for every node R_j that is contained in bag $\mathbf{X_k}$. Every node $A_i, 1 \leq i \leq m$ has a CPT that corresponds to a logical 'and' of its parents, i.e., $\Pr(A_i = \text{TRUE} \,|\, \pi(A_i)) = 1$ if and only if all parents of A_i have the value TRUE,

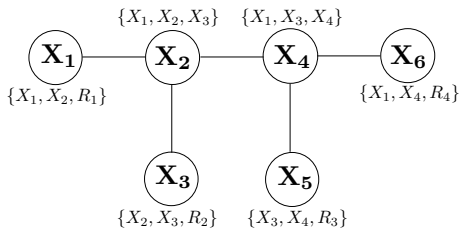


Figure 4: Tree-decomposition of the moralisation of $\mathbf{G}_{\text{ex}}$

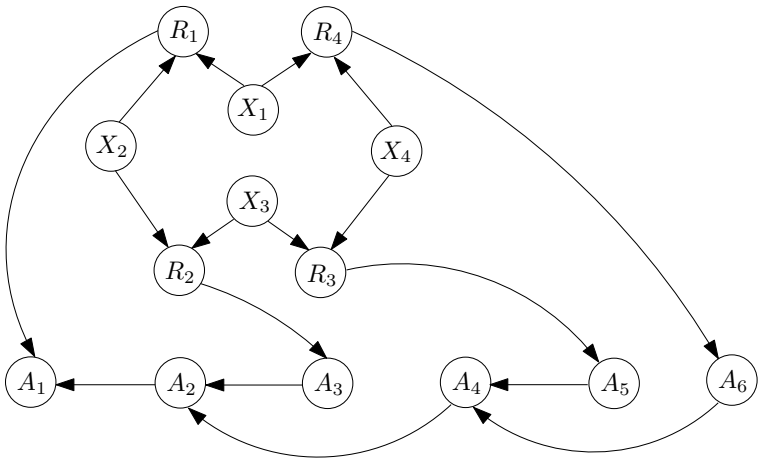


Figure 5: Adding arcs A_i to $\mathbf{G}_{\text{ex}}$ to construct $\mathbf{G}'_{\text{ex}}$

and 0 otherwise. If A_i has no parents, then we set $\Pr(A_i = \text{TRUE}) = 1$. Note that eventually all nodes R_j are 'chained' together in A_m, and $\Pr(A_m = \text{TRUE}) > 0$ if $\Pr(\forall_j R_j = \text{TRUE}) > 0$. We then make a tree-decomposition $\mathbf{T}'_{\mathbf{G}}$ of the moralised graph $\mathbf{G}'_{\mathbf{M}}$ of $\mathcal{B}'_{\mathcal{I}}$, by enhancing any bag $\mathbf{X_k}$ in $\mathbf{T_G}$ with the variable A_k and all variables A_l that are contained in the bags of the children of k.

A tree-decomposition (with at most two children per node) of the moralisation of our example network $\mathcal{B}_{\mathcal{I}_{\text{ex}}}$ is given in Figure 4. We assume that X_1 is the root of the tree. This tree-decomposition has six nodes so we add nodes $A_1, \ldots, A_6$ to $\mathbf{G}_{\text{ex}}$ (Figure 5). For the first bag, we add A_1 and A_2 and add the arcs (A_2, A_1) (since $\mathbf{X_1}$ is adjacent to $\mathbf{X_2}$ in the tree-decomposition) and (R_1, A_1) (since R_1 is contained in bag $\mathbf{X_1}$. For consecutive bags $\mathbf{X_i}$, we similarly add nodes $A_3, \ldots, A_6$ and arcs $(A_3, A_2), (A_4, A_2), (A_5, A_4)$, and (A_6, A_4) and from the relation nodes R_j in bags $\mathbf{X_i}$ to A_i. In the thus constructed fragment, every node A_i either has a CPT that corresponds to the truth table of a logical 'and' operator on its parents, or $\Pr(A_i = \text{TRUE}) = 1$ for nodes without incoming arcs. The resulting tree-decomposition of $\mathcal{B}_{\mathcal{I}_{\text{ex}}}$ is given in Figure 6.

We claim that, for any instance $\mathcal{I}$ of Constraint Satisfaction, in the thus constructed probabilistic network $\mathcal{B}_{\mathcal{I}}$, $\Pr(A_m = \text{TRUE}) > 0$ if and only if $\mathcal{I}$ is satisfiable.

Theorem 4.2. Constraint Satisfaction *tw-reduces to* Inference.

Proof. To show that this construction indeed gives a treewidth-preserving polynomial time

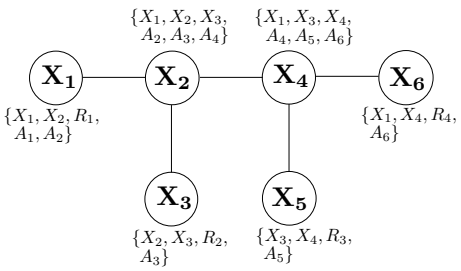


Figure 6: Resulting tree-decomposition of the moralisation of $\mathbf{G}'_{\text{ex}}$

---

reduction from Constraint Satisfaction to Inference, we need to prove that there exists a function f such that for all $x \in$ Constraint Satisfaction, $f(x) \in$ Inference, that f is computable in polynomial time and that f preserves treewidth up to a linear factor, i.e., $\text{tw}(f(x)) = l(\text{tw}(x))$.

Let $\mathcal{I}$ be any instance of Constraint Satisfaction, and let $\mathcal{B}'_{\mathcal{I}}$ be the probabilistic network constructed from I as shown above. If $\Pr(A_m = \text{TRUE}) > 0$, then the 'and' construction modelled with the A_i nodes enforces that $\Pr(\forall_j R_j = \text{TRUE}) > 0$. For any node R_j, $\Pr(R_j = \text{TRUE} \mid \mathbf{v}) = 1$ for a particular joint value assignment $\mathbf{v}$ if and only if that value assignment satisfies the relation R_j. Thus, if $\Pr(\forall_j R_j = \text{TRUE}) > 0$ then there must be a joint value assignment to all variables that satisfies all relations; hence, $\mathcal{I}$ is satisfiable. On the other hand, if there is a satisfying assignment to $\mathcal{I}$ then $\Pr(\forall_j R_j = \text{TRUE}) > 0$ and hence $\Pr(A_m = \text{TRUE}) > 0$.

Clearly, the above reduction can be constructed in polynomial time; what remains is to show that the reduction preserves treewidth up to a linear factor. The construction using the V_i and R_j nodes has a treewidth of $\min(2, \text{tw}(\mathcal{I}))$ and thus increase the treewidth by at most 1. We will show that the A_k nodes and the arcs (R_j, A_k) and (R_k, A_l) increase the treewidth by at most three. To facilitate our proof, we assume that $\mathbf{T_G}$ is *nice*, thus every node in $\mathbf{T_G}$ is either a leaf node, insert node, forget node, or join node; we show that our claim holds for all of these nodes.

- leaf nodes: Suppose i is a leaf node. Then i has no children, and only the node A_i is added to the bag $\mathbf{X_i}$, and the treewidth is increased by at most 1.
- insert/forget nodes: Suppose i is an insert or forget node. Then i has one child j, and the nodes A_i and A_j are added to the $\mathbf{X_i}$, and the treewidth is increased by at most 2.
- join nodes: Suppose i is a join node. Then i has two children j and k, and A_i, A_j, and A_k are added to $\mathbf{X_i}$, and the treewidth is increased by at most 3.

We conclude that any operation on one of the nodes in the tree-decomposition can increase the size of the corresponding bag by at most three, and thus that the treewidth is increased by at most three. Hence, the above reduction preserves treewidth up to a constant term, and thus Constraint Satisfaction tw-reduces to Inference. □

Our main result in Theorem 3.2 now follows from Theorem 4.2 and Marx' result [18] as formulated in Theorem 3.1.

of Theorem 3.2. Assume that there exists an algorithm A that solves arbitrary instances $\mathcal{B}$ of the INFERENCE problem with unbounded treewidth in time $f(\mathbf{G_M}) \cdot \|\mathcal{B}\|^{o(\frac{\mathrm{tw}(\mathbf{G_M})}{\log \mathrm{tw}(\mathbf{G_M})})}$, where f is a computable function and $\mathbf{G_M}$ denotes the moralised graph of $\mathcal{B}$. Let $\mathcal{I}$ be an instance of CONSTRAINT SATISFACTION with sufficiently large treewidth. Following Theorem 4.2, we can reduce $\mathcal{I}$ to $\mathcal{B}'_{\mathcal{I}}$ (with moralised graph $\mathbf{G_I}$) in polynomial time, where $\mathrm{tw}(\mathbf{G_I}) = \mathrm{tw}(\mathcal{I})+3$. Since we assumed that A solves $\mathcal{B}'_{\mathcal{I}}$ in time $f(\mathbf{G_M}) \cdot \|\mathcal{B}\|^{o(\frac{\mathrm{tw}(\mathbf{G_M})}{\log \mathrm{tw}(\mathbf{G_M})})}$, there exists a function g such that $\mathcal{I}$ can be solved in time $g(\mathbf{G}) \cdot \|\mathcal{I}\|^{o(\frac{\mathrm{tw}(\mathbf{G})}{\log \mathrm{tw}(\mathbf{G})})}$. From Theorem 3.1 follows, that this contradicts the ETH. □

5 Conclusion

In this paper, we have proven that there can not exist an algorithm solving arbitrary instances of INFERENCE with high treewidth in polynomial time, unless the ETH fails. In fact, we show that any algorithm solving arbitrary instances of INFERENCE must have a running time of $f(\mathbf{G_M}) \cdot \|\mathcal{B}\|^{\omega(\frac{\mathrm{tw}(\mathbf{G_M})}{\log \mathrm{tw}(\mathbf{G_M})})}$, i.e., exponential in the treewidth of the moralised graph, up to a logarithmic factor in the exponent. This result is weaker than Marx' result [18] for CONSTRAINT SATISFACTION and GRAPH HOMOMORPHISM, which shows a lower bound for algorithms solving these problems on *any* recursively enumerable class of graphs. Our result still allows algorithms to use specific properties of a network, like a particular arc direction, or planarity of the moralised graph, to arrive at sub-exponential running times, whereas Marx' result holds even for restricted classes of graphs. Nevertheless, our result is derived in a different way, namely using a treewidth preserving reduction from CONSTRAINT SATISFACTION. This technique may be of independent interest. For example, it can be used to prove (conditional) lower bounds on the running time of other problems (in probabilistic networks or otherwise) whose instances have treewidth as a structural property.

Acknowledgements

The authors wish to thank Linda van der Gaag, Jan van Leeuwen, and Gerard Tel for fruitful discussions on this subject and useful comments on earlier drafts of this paper.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti Spaccamela, and M. Protasi. *Complexity and Approximation.* Berlin: Springer, 1998.

[2] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.

[3] H. L. Bodlaender. Treewidth: Algorithmic techniques and results. In *Twenty-second International Symposium on Mathematical Foundations of Computer Science*, volume LNCS 1295, pages 19–36. Springer-Verlag, 1997.

[4] H. L. Bodlaender. Treewidth: characterizations, applications, and computations. In *Proceedings of the 32nd International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 1–14, 2006.

[5] Hans L. Bodlaender, Arie M. C. A. Koster, and Frank van den Eijkhof. Pre-processing rules for triangulation of probabilistic networks. *Computational Intelligence*, 21(3):286–305, 2005.

[6] V. Chandrasekaran, N. Srebro, and P. Harsha. Complexity of inference in graphical models. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI'08)*, pages 70–78. AUAI Press, 2008.

[7] J. Chen, X. Huang, I. A. Kanj, and G. Xia. Linear FPT reductions and computational lower bounds. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing*, pages 212–221, 2004.

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

[9] P. L. Geenen, A. R. W. Elbers, L. C. van der Gaag, and W. L. A. van der Loeffen. Development of a probabilistic network for clinical detection of classical swine fever. In *Proceedings of the Eleventh Symposium of the International Society for Veterinary Epidemiology and Economics*, pages 667–669, 2006.

[10] Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1–24, 2007.

[11] H. J. Henriksen, P. Rasmussen, G. Brandt, D. von Bülow, and F. V. Jensen. Public participation modelling using Bayesian networks in management of groundwater contamination. *Environmental Modelling and Software*, 22(8):1101–1113, 2007.

[12] R. Impagliazzo and R. Paturi. On the complexity of k-sat. *Journal of Computer and System Sciences*, 62(2):367 – 375, 2001.

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

[14] T. Kloks. *Treewidth*. LNCS 842. Springer-Verlag, Berlin, 1994.

[15] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, 50(2):157–224, 1988.

[16] A. K. Lenstra, H. W. Lenstra Jr., M. S. Manasse, and J. M. Pollard. The number field sieve. In *Proceedings of the 22nd Annual ACM Conference on Theory of Computing*, pages 564–572, 1990.

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

[18] D. Marx. Can you beat treewidth? In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 169–179, 2007.

[19] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

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

[21] N. Robertson and P.D. Seymour. Graph minors II: Algorithmic aspects of tree-width. *Journal of Algorithms*, 7:309–322, 1986.

[22] R. D. Schachter. Evaluating influence diagrams. *Operations Research*, 34(6):871–882, 1986.

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

[24] C. Zhang, S. Sun, and G. Yu. A Bayesian networks approach to time series forecasting of short-term traffic flows. In *Seventh International IEEE Conference on Intelligent Transportation Systems*, pages 216–221, 2004.